1

INTRODUCTION TO MULTIMEDIA

Unit Structure

1.1 Objectives
1.2 Introduction
1.3 What is Multimedia?
1.4 Defining the Scope of Multimedia
1.5 Hypertext and Collaborative research
1.6 Multimedia and personalized computing
1.7 Multimedia on the map
1.8 Emerging applications
1.9 The challenges
1.10 Summary
1.11 Unit End Exercises

1.1 OBJECTIVES

After studying this Unit, you will be able to:

1. Define Multimedia
2. Define the scope and applications of multimedia
3. Understand Challenges faced while developing Multimedia applications

1.2 INTRODUCTION

Multimedia refers to content that uses a combination of different content forms. This contrasts with media that use only rudimentary computer displays such as text-only or traditional forms of printed or hand-produced material. Multimedia includes a combination of text, audio, still images, animation, video, or interactivity content forms.

Multimedia is usually recorded and played, displayed, or accessed by information content processing devices, such as computerized and electronic devices, but can also be part of a live performance. Multimedia devices are electronic media devices used to store and experience multimedia content. Multimedia is distinguished from mixed media in fine art; by including audio, for example, it has a broader scope. The term "rich

media" is synonymous for interactive multimedia. Hypermedia can be considered one particular multimedia application.

1.3 WHAT IS MULTIMEDIA?

The word “multimedia” comes from the Latin words *multus* which means “numerous” and *media* which means “middle” or centre. Multimedia therefore means “multiple intermediaries” or “multiple means”.

Multimedia is a combination of following elements:

- Text [E.G. books, letters, newspapers]
- Images and graphics [E.G. photographs, charts, maps, logos, sketches]
- Sound [E.G. radio, gramophone records and audio cassettes]
- Video and animation [E.G. TV, video cassettes and motion pictures]

1.4 DEFINING THE SCOPE OF MULTIMEDIA

Today, much of the media content we consume is available in a variety of formats, intended to serve multiple purposes and audiences. For example, a book usually starts out as a print-only product. However, if the market demand is large enough, it may also be published in a spoken-word format and delivered via compact disc or MP3. With the right equipment, you can avoid paper altogether by downloading the e-book, a digital version of the text designed for reading on a computer screen or a handheld device such as the Kindle or iPad.

The website for a bestseller may offer bonus material or value-added content to online users through a gamut of multimedia channels—featuring audio excerpts, video interviews, background stories, pictures, and more. With such a vast sea of information and social networking potential, you can easily imagine many of the other possibilities that exist.

The opportunities for shaping content to meet the diverse needs and habits of different user groups are numerous, and they are changing rapidly, as the culture of multimedia continues to grow and permeate nearly every aspect of our personal and professional lives.

1.5 HYPERTEXT AND COLLABORATIVE RESEARCH

The Hypertext Editing System or HES was an early research project separately conceptualized and undertaken at two different universities in the 60’s. In the year 1967, at Brown University, Theodor Holm Nelson and Andries van Dam collaborated to create HES which served as a predecessor for animation and GUI. At Stanford University in

the year 1968, Doug Engelbart designed and developed the On-Line System (NLS) consisting of text editing, text branching, links and text search services.

1.6 MULTIMEDIA AND PERSONALIZED COMPUTING

Multimedia found its way into personalized computing making it simpler.

- Spatial Data Management System
- Movie maps and Surrogate Travel
- The Electronic Book
- Formation of the MIT Media Lab
- Digital representation

1.6.1 SPATIAL DATA MANAGEMENT SYSTEM:

In the year 1976, the Architecture Machine Group made a proposal under the title "Augmentation of Human Resources in Command and Control through Multiple Media Man-Machine Interaction" to the U.S. Defense Advanced Research Projects Agency (DARPA). The Spatial Data Management System (SDMS) makes use of spatial information to access data in a virtual space. The SDMS media room consisted of a large wall-sized display screen, an instrumented Eames chair, touch-sensitive monitors on the side and an octaphonic sound system that provide spatial audio cues and wrap-around sound. The chair had joysticks and touch sensitive screens that guide the user through "DataLand" on the huge screen. The side-monitors gave a top-view of the "DataLand". Later SDMS II was developed that helped get a detailed view of information with voice activated commands built into the system.

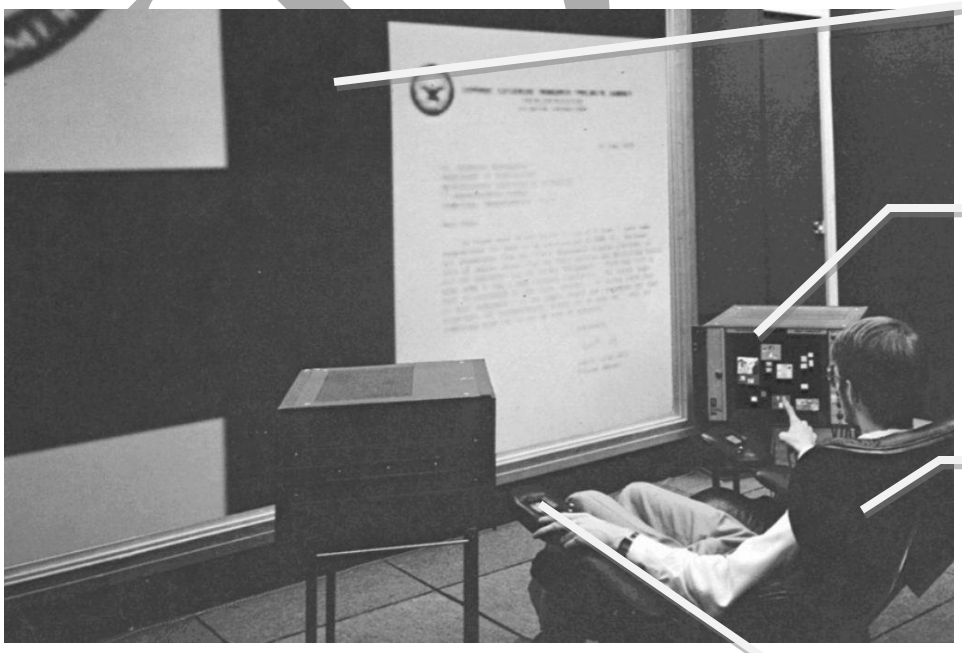


Figure: Top-level view of DataLand in SDMS

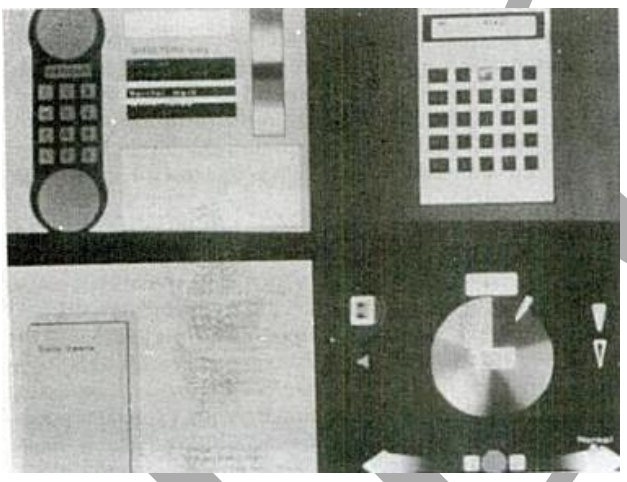


Figure: SDMS monitor interface showing various tools available

1.6.2 MOVIE MAPS AND SURROGATE TRAVEL:

The Aspen Project was initiated by Andrew Lippman and Robert Mohl in 1987.It allowed the user to take a virtual tour of the city similar to surrogate travel. Cameras were mounted on top of the car and it shot views of the Aspen city in Colorado. The film was stored on laserdiscs after assembling them.

These images could be then accessed via joystick that allowed the user to adjust the speed and direction of travel and also access additional details about the buildings they came across in the course of travel. The research student Robert Mohl developed an added feature that enabled the user to get a navigational aerial view of the city. This allowed the user to go back and forth as well as zoom in and out the city with ease

using routes and landmarks. This kind of surrogate travel had innumerable applications resulting from navigational ease to Military operations.

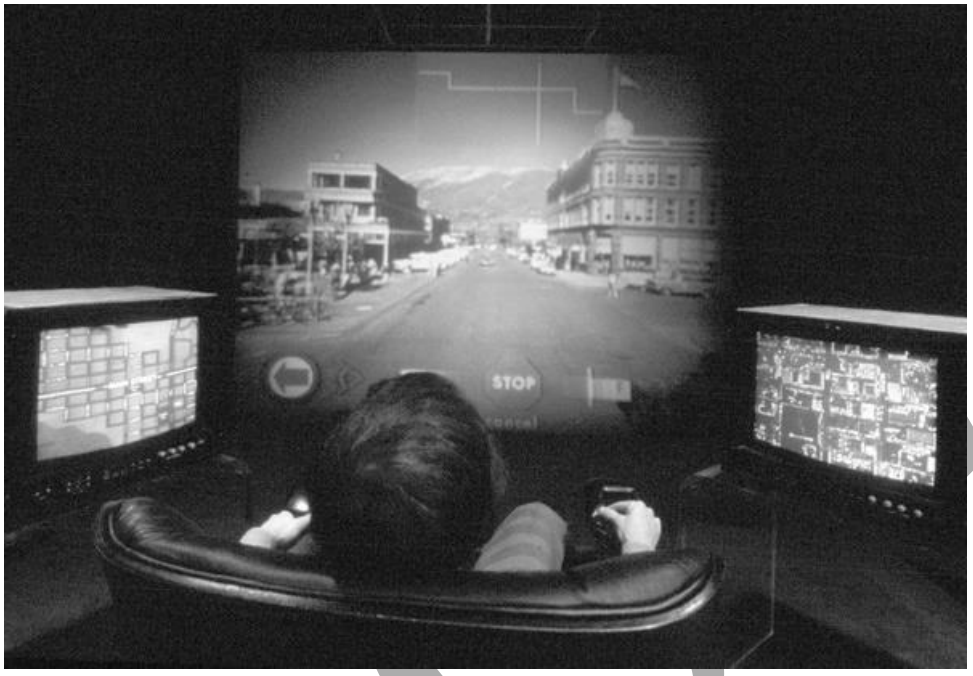


Figure: The Aspen Moviemap experienced in the "Media Room" at the Architecture Machine Group, MIT, c1980. The "traveler," seated in an instrumented armchair, controls speed and direction of travel. Touch screens displaying map and aerial views allow access to additional multimedia material. (Photo: Bob Mohl)

1.6.3 THE ELECTRONIC BOOK:

The Electronic book was actually a prototype created by Dave Becker for his research thesis. It had the following advantages over an ordinary book: It had authoring tools that enables linking objects together like text, images or videos. It also supported scripting language.

Personalization was another feature that allowed user to add annotation and make further additions to the existing content. It also had searching and indexing. The E-book had integrated audio, video, sound, images and text. The video and text alongside the video are bound in a synchronized manner i.e. changes in any one of them are reflected immediately in the other on its own. For example: The text alongside the video changes as the video progresses.

1.6.4 THE ESTABLISHMENT OF THE MIT MEDIA LAB:

In the year 1985, Nicholas Negroponte co-founded MIT Media Lab with Jerome Wiesner who was the president of MIT.

Group	Investigators	Interests
Electronic Publishing	Walter Bender	On-Line, personalized multimedia newspapers/magazines
Film and Video	Ricky Leacock	Convergence of film production and presentation with computer technology
Visual Language Workshop	Muriel Cooper, Ron McNeil	New User interface and authoring techniques
Electronic Music	Barry Vercoe	Computer composition, synthesis and performance of music
Spatial Imaging and Photography	Steve Benton	Technology for holograms and holographic movies
Learning and Epistemology	Seymour Papert	Deconstructible computer-based learning environments
Movies of the Future	Andy Lippman	Digital video as media; hierarchical coding of video
Advanced Television Research	William Schreiber	High-definition and digital television coding techniques
Speech Processing	Chris Schmandt	Uses of audio and speech recognition in the user interface; computer-based telephony
Human-Machine Interface Group	Dick Bolt	Development and use of new input devices such as gloves and eye-trackers

1.7 MULTIMEDIA ON THE MAP

Here we will discuss about two other esteemed research groups that emerged in the 1980's:

- Olivetti Research Lab
- Apple Computer Multimedia Lab

- Olivetti Research Lab

Olivetti Research Lab was situated in Cambridge, England. Its research work was dedicated towards creating a multimedia system called Pandora and a tracking system based on sensors called Active Badge. The Olivetti research group at California focused its development on an Audio Server Software called VOX.

- Apple Computer Multimedia Lab

Apple Computer Multimedia Lab was an electronic multimedia research group founded by Krishna Hooper Woolsey. This lab was involved in groundbreaking research on education based projects.

1.8 EMERGING APPLICATIONS

1.8.1 ENTERTAINMENT: GROWING INTERACTIVITY AND GROUP PARTICIPATION:

Multimedia plays a vital role in the flourishing of the Entertainment Industry. This industry has a huge impact on the economy of any nation. A multimedia system enables user interaction with games and movies as well as maximum involvement of people in the process of software development as well as deployment.

Video-On-Demand

Video-On-Demand or VOD gives the viewer the power to choose and watch any video irrespective of the broadcast time with facilities such as pausing, fast-forwarding or rewinding the video. This system may use a set-top box to stream video content to the user's home from the television providers.

Interactive Cinemas

The Movie I'm Your Man was a movie designed on Interactive Cinema. David Bejan's interactive cinema company called Controlled Entropy Entertainment released this movie in 1992.It provided the users with joysticks allowing them to vote and decide the fate of the three characters.

The audience could register their votes after every 90 seconds when asked for their inputs with three options to lead the story further. Each armrest had three buttons. The majority of votes decided the direction of the movie plot.

Collaborative Computer-Supported Games

The Next President Game launched by Prodigy Services Company during the 1992 Presidential elections garnered the collaboration of hundreds of participants who decided the outcome of the elections in the on-line game by voting for their preferred fictitious candidate.

Figure: The Next President, an on-line game in which hundreds of participants compete and collaborate to determine the fate of their favourite fictitious presidential aspirant. (Reproduced with permission of Prodigy Services Company, © 1992 Prodigy Services Company; game designed and developed by Crossover Technologies)

1.8.2 HOME SHOPPING:

Home-Shopping is a booming industry today. Customers can purchase products as per their convenience within the confines of their home. Home-Shopping has branched out into two categories viz. Interactive television and Video-telephony.

- Interactive television:

  The Television is not an interactive media. Interactive television is a model that blends the Television and the computer. This blend will result in the best of both worlds i.e. maximum Interactivity as well as a huge audience that can access various channels empowering them to shop from home.

- Video-telephony:

  Video – telephony is a model that combines the television and the Telephone. In order to buy a product, the customers would dial the respective retailers and speak to a salesperson or be redirected to a recorded audio.

Home-Shopping was also applied in real-estate firms. Kiosks called Home-Vision™ developed by a Denmark Company enabled buyers to view various perspectives of the rooms of their property of interest. It also provided selection parameters such as price and geographic location of the property. The interested

buyers could mention their respective criteria and get a collage of images matching their selected preferences. Selection of a particular house from among the displayed images, takes the buyer on a virtual tour. There is also an accompanied sound track with the images that can be enabled or disabled on the buyer's will.

The Home Vision System could track the prospective buyers as they viewed the data about Houses on Sale. These statistics thus generated were used by real-estate agents. The data about prospective buyers could be now used for marketing purpose as well as by sellers to better understand the distinct needs of their target audience.

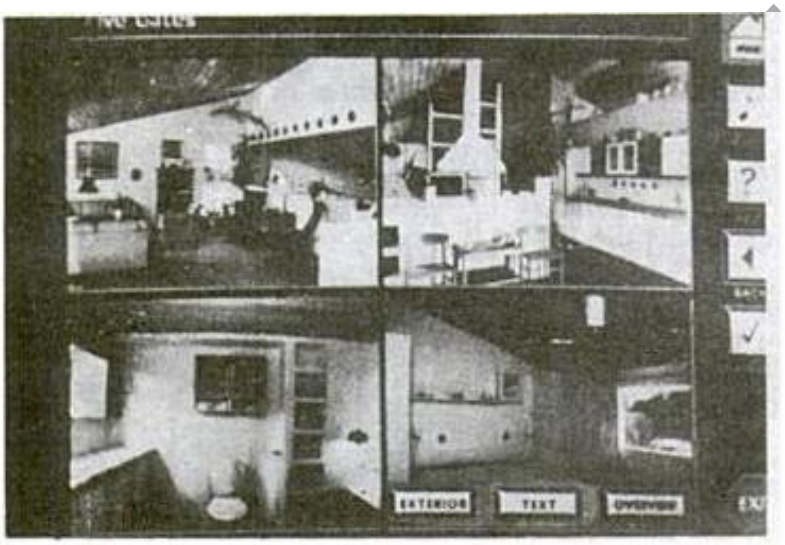


Figure: User Interface of Home-Vision™ , a multimedia Kiosk application that uses an IBM PS/2® running OS/2® and was developed by the Danish company Multimedia Management

1.8.3 MULTIMEDIA COMMUNICATIONS FOR HEALTHCARE:

NYNEX was a regional Bell operating company that was involved in a Media Broadband Services (MBS) study where it provided four Boston hospitals with broadband interconnections. NYNEX worked together with the hospitals to provide video conferencing and image storage and retrieval at high bandwidth-rates.

NYNEX derived the below conclusions from its study:

- MBS enabled electronic transfer of images thus saving the time and effort required to transport the hardcopy images.

- Specialists have multimedia workstations installed in their homes that enable them to examine Computer Tomography (CT) and Magnetic Resonance Imaging (MRI) images and consult physicians.

- Hospitals now had an upgraded on-line Medical database complete with Images along with the existing textual data. This enabled physicians from different hospitals to easily access and share patient information.

Although NYNEX incurred the cost of MBS, but that served as an opportunity to reap rewards in the following domain:

- Cost Cutting in Healthcare services:
  - On-line storage and retrieval of hardcopy images and medical records will reduce the cost of transportation between storage and hospitals since millions of paper documents & medical records could be stored on disks.
- Multiple possibilities for income:
  - MBS will improve communication and collaboration between small hospitals and well established ones. The former can easily buy the advanced quality services from the major hospitals thus improving revenue growth.
- Quality Patient Care:
  - On-line Video Conferencing and sharing of data promises best patient care since specialists can be reached easily for consultation.

High bandwidth communication thus betters Healthcare Quality by conveniently storing, processing and sharing of images like X-ray, CT, MRI etc.

Figure: Multimedia Communications for a radiology application. (Reproduced with permission of NYNEX Science and Technology)

1.8.4 GEOGRAPHIC INFORMATION SYSTEMS:

GIS is a system designed to store and process spatially organized data. Users can write interactive queries related to areas or locations. Below are examples of uses of GIS:

- The National Capital Planning Commission (NCPC) in Washington DC provides comprehensive planning of park system by making available all the spatial

information online.

- Union Pacific Railroad used GIS for infrastructure and maintenance plans as well as safety and security so as to reduce accidents.

- Great Britain Historical GIS is a Database created and maintained by Portsmouth University. It stores geo-spatial maps, statistics as well as travel literature.

- Crime mapping is used by analysts in law enforcement agencies to map, visualize, and analyze crime incident patterns. It is a key component of crime analysis and the CompStat policing strategy. Mapping crime, using Geographic Information Systems (GIS), allows crime analysts to identify crime hot spots, along with other trends and patterns.

- GIS is used in Remote Sensing Application that process remote sensing data. Remote sensing is the acquisition of information about an object or phenomenon without making physical contact with the object and thus in contrast to on site observation.

  Georgia Power Company developed multimedia GIS served as a Database for locating typographic information. Search could be performed on various criteria such as building, demographics, education, transportation etc.
  The system consisted of distributed computers that controlled six video projectors for the wall and Table display. The outputs of queries fired on the database were sent to the CPU by the Interface Manager. The CPU displayed the required spatial data onto the respective screens.

Figure: The Georgia Power Company's multimedia GIS for the Georgia Resource Center's business locator presentation system. (Reproduced with permission of Georgia Power Company)

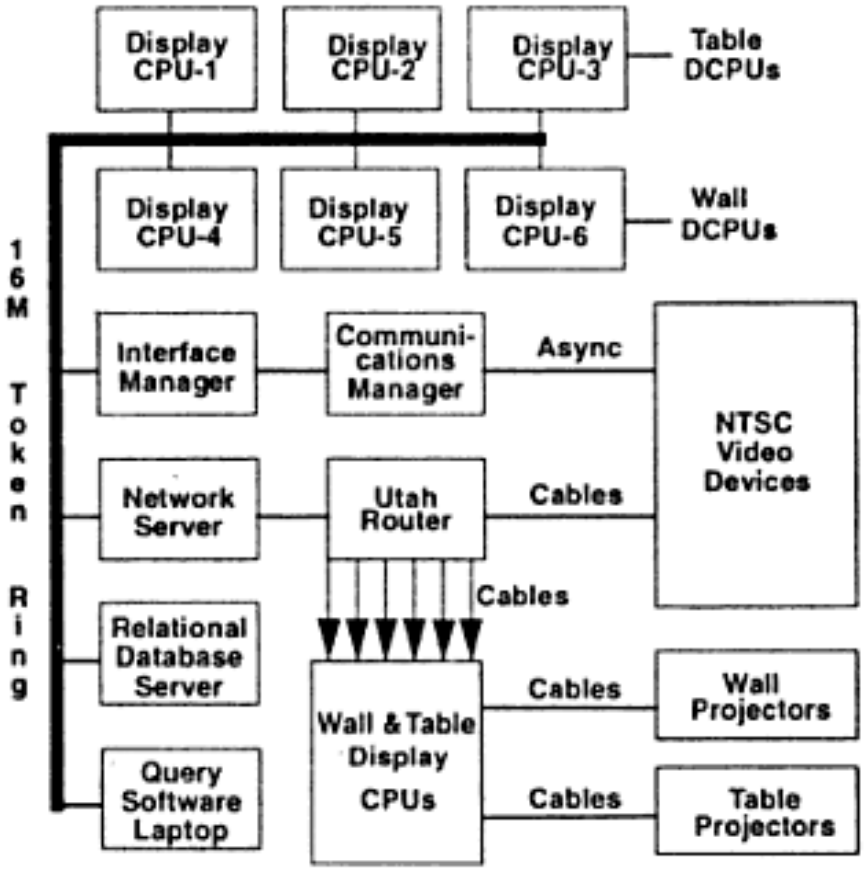


Figure: Architecture of the multimedia presentation system developed by the Georgia Power Company. (Reproduced with permission of Georgia Power Company)

1.8.5 EDUCATION:

Multimedia has played a very vital role in Education for ages. Multimedia makes learning an interesting and engaging experience. Apple Media Lab's contribution towards developing Educative Multimedia projects has been worthwhile. The below table describes some noteworthy Projects created by Apple Labs.

Project	Description
Visual Almanac	A videodisk collage of audiovisual materials for educators organised in 12 collections. Over 7000 media objects, 5000 from external sources.
Classroom Multimedia Kiosk	A kiosk with video presentation, a barcode scanner and a printer to make printed cards for a particular topic. The cards are bar-coded and associated with a video sequence decided by the student. It can be later scanned to play that sequence.
101 Activities	The computer guides the kids through creative and interesting

	activities by providing images and videos. It provides the list of activities, the complexity level and the time of completion on the basis of gathered items with the child.
GTV: A Geographical Perspective on American History	Combines still images and music in an MTV-style presentation popular with middle school age group
Interactive NOVA: Animal Pathfinders	Supplements the NOVA movie about animal migration with additional documentation and three activities that involve the students in the subject matter.
Life Story	Supplements BBC's movie about discovery of DNA with accessory materials including documentary interviews, text transcriptions, simulations, references and navigational tools
Mystery of the Disappearing Ducks	Using raw footage from a TV documentary on wetlands provides an interactive means to explore the ecological controversy. Includes a mystery game about disappearing ducks developed by high school students and professional designers

1.8.6 MULTIMEDIA COMMUNICATIONS: AN ENABLING TECHNOLOGY FOR CONCURRENT ENGINEERING AND MANUFACTURING

Concurrent engineering, also known as simultaneous engineering, is a method of designing and developing products, in which the different stages run simultaneously, rather than consecutively. It is based on the parallelization of task. It decreases product development time and also the time to market, leading to improved productivity and reduced costs.

Computer-aided manufacturing (CAM) and Computer-aided design (CAD) softwares help design, manufacture and maintain any product under development. The University of Defence Analysis conceived concurrent engineering activities. Multimedia technology accelerates production using Concurrent engineering. The group members can collaborate with each other despite geographic boundaries in a distributed environment via group discussions and video conferencing and also share of data on-line.

A Demonstrative project on distributed online workstation was developed by CPE and funded by NYNEX. The CPE Factory-of-the-future was designed to demonstrate various concepts:

- Design and manufacturing of the product could be performed simultaneously
- Coordination among Robots could be established to reduce errors.
- Design served as an instructional blueprint for manufacturing
- Design process could be carried out with precision.

Engineering with the aid of Multimedia benefited in two ways:

- Visually designed products in detail lead to high quality manufactured products.
- Communication and Collaboration among team -members via the computer workstation helps them share data as well as discuss and resolve issues on-line.

1.8.7 THE IMPACT OF UBIQUITOUS MULTIMEDIA SERVICES:

Multimedia is extensively used in a variety of fields. The Impact of Multimedia is quite significant. Communication and collaboration via video, audio, games, tools and data among geographically divided groups results in an enriched experience.

1.9 THE CHALLENGES

Below listed are some challenges that might be faced while developing Multimedia systems:

- Maintaining synchronization between various types of media like audio, video, text and the sequencing of information according to time in correct order
- Storage and management of Data
- Networking operation issues
- Adjusting to New Multimedia communication paradigms and techniques

1.10 SUMMARY

- The Hypertext Editing System or HES was an early research project consisting of text editing, text branching, links and text search services.
- Multimedia found its way into personalized computing making it simpler through Spatial Data Management System, Movie maps and Surrogate Travel, Electronic Book etc.
- Multimedia plays a vital role in the flourishing of the Entertainment Industry.
- The Next President, an on-line game in which hundreds of participants compete and collaborate to determine the fate of their favourite fictitious presidential aspirant.

- Home-Shopping has branched out into two categories viz. Interactive television and Video-telephony.
- GIS is a system designed to store and process spatially organized data.
- Multimedia makes learning an interesting and engaging experience.

Reference:

1. Multimedia Systems, John F. Koegel Buford, Pearson Education.
2. Wikipedia: http://en.wikipedia.org/wiki/
3. www.theverge.com
4. Concurrent-Engineering : http://www.concurrent-engineering.co.uk/what-is-concurrent-engineering/

❖❖❖❖

2

THE CONVERGENCE OF COMPUTERS, COMMUNICATIONS, AND ENTERTAINMENT PRODUCTS

Unit Structure

2.1 The Technology Trends
- 2.1.1 Electronics
- 2.1.2 Communications
- 2.1.3 Presentation Technology
- 2.1.4 Input Technology

2.2 Multimedia Appliances: Hybrid Devices
- 2.2.1 Computer-Based Video Phones
- 2.2.2 Computer-Based Consumer Entertainment Products
- 2.2.3 Some More Hybrid Examples

2.3 Designers Perspective
- 2.3.1 Industrial Design Considerations
- 2.3.2 Human Factors and the Design of New Products

2.4 Industry Perspective of the Future
- 2.4.1 Telecommunications
- 2.4.2 Entertainment
- 2.4.3 Information Services
- 2.4.4 Computer Industry

2.5 Key Challenges Ahead: Technical, Regulatory, Social

2.6 Summary

2.7 Unit End Exercises

2.1 THE TECHNOLOGY TRENDS

2.1.1 ELECTRONICS: INCREASING CIRCUIT DENSITY

Intel was founded by Robert Noyce and Gordon Moore. In his 1965 paper, Gordon Moore described that over the history of computing hardware, the number of transistors in a dense integrated circuit doubles approximately every two years. This prediction about the growth rate of semiconductors is known as Moore's Law. His prediction has proven to be accurate. In1990 at the Microprocessor Forum, Andrew Rappaport claimed that silicon is free and stated that technology is growing at such a fast rate that we are not limited by technology but by more ideas.

2.1.2 COMMUNICATIONS: INCREASING BANDWIDTH AND SWITCHING SPEED

Datakit is a virtual circuit switch developed at Bell Labs. This could achieve 9600 baud or full duplex data on the same twisted pair phone wire independent of data. The ***A***symmetric ***D***igital ***S***ubscriber ***L***ine (ADSL) yielded higher bit rates but required expensive modems. Use of fiber optics proved to be a breakthrough as it was a replacement of the existing wiring and more data could be sent over it. Asynchronous Transfer Mode (ATM) is a high speed networking standard that yields increased bandwidth and is a foundation for Broadband ISDN (B-ISDN) and future Cable Television (CATV) Networks.

2.1.3 PRESENTATION TECHNOLOGY

The CRT uses a vacuum tube with red, green and blue electron guns emitting electron beams on a phosphor screen to create images onto the screen. CRTs have a spherical shaped screen and are quite bulky. The newer CRT displays are generally flat.

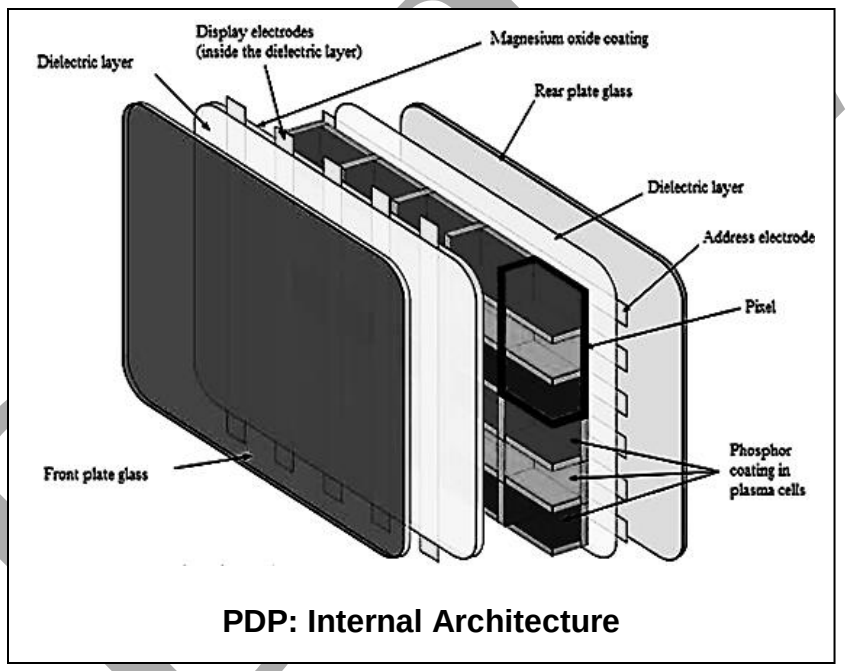
The newer display types also use projection systems to display an image. Projection systems are available as front-projection and rear-projection types. In a front-projection system, a video projector is used to illuminate a screen in much the same manner as film is displayed.

The screen size can be as large as one hundred inches measured diagonally. Rear-projection systems generally have an internal projector and a mirror to reflect the image on to the viewing screen and are viewed in much the same manner as a direct-view CRT.

The viewing screen is generally forty to eighty inches in size. In the older type of projection systems, both front and rear systems use three CRTs—one red, one blue and one green—that are aligned to create a single image on the viewing screen.

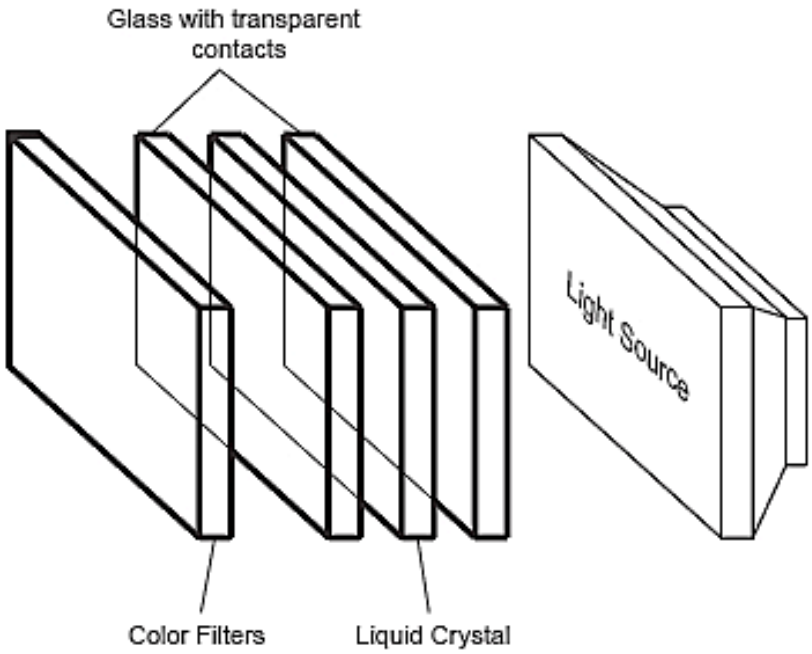
The newer types of displays include Plasma Display Panel (PDP), Liquid Crystal Display (LCD), Digital Light Processing (DLP), Liquid Crystal on Silicon (LCoS), Light Emitting Diode (LED), Surface Conduction electron-emitter Display (SED).

Plasma is also called ionized gas. Plasma is an energetic gas-phase state of matter, often referred to as “the fourth state of matter”, in which some or all of the electrons in the outer atomic orbital have become separated from the atom. The result is a collection of ions and electrons which are no longer bound to each other. Plasma Display Panel(PDP) is an emissive (i.e. discharge of electromagnetic radiation or particles) Flat Panel Display. It is so called because by applying voltage between electrodes, the neon-based gas within the panel behaves as they were small independent cells i.e plasmas. It produces a quite steady and totally flicker free image. Even if images are viewed from wider angles, that does not degrade its quality as in the case of LCDs that need straight ahead viewing angles. The screen size may vary from 30 inches to larger. The thickness of the panel reaches about 10 cm.



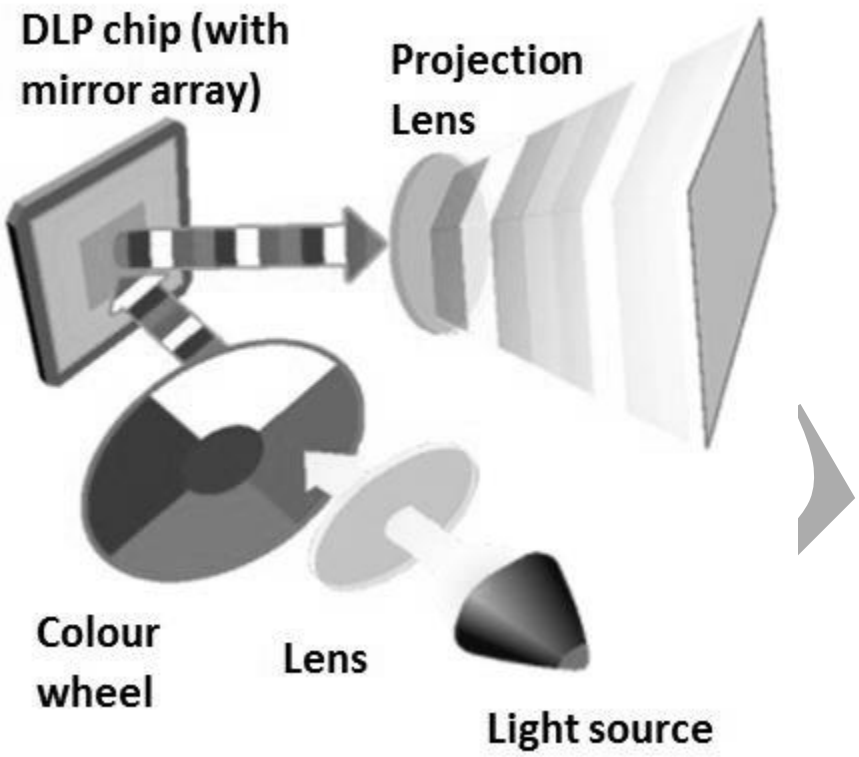
PDP: Internal Architecture

Liquid Crystal behaves like a liquid but has a crystalline arrangement of molecules. There are two glass plates, one having rows of conductors and the other, column of conductors etched into it. Each glass plate has a light Polarizing film at right angles to the other plate with liquid-crystal material in between them to generate pixels. In order to get different colors, different polarizing films can be used. Large sized LCDs having more than 24 inches can be made. Both plasma and LCDs provide better pictures that will take up less space in our rooms. CRTs have served us well for the last 60 years, but their days are numbered.

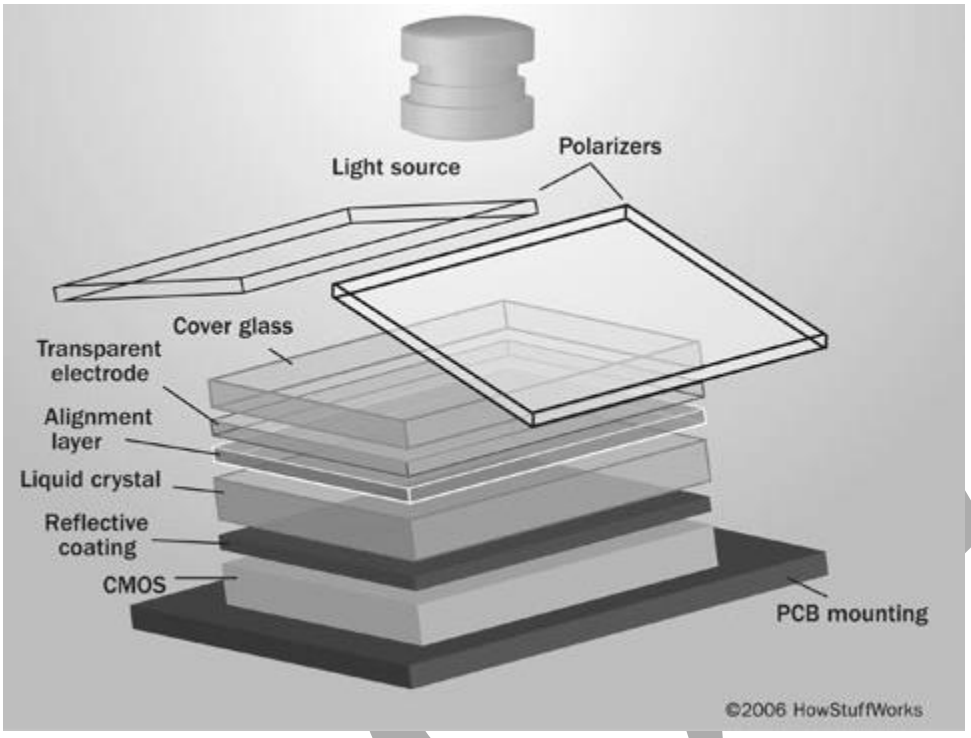


LCD screens are made up of many layers.

Digital Light Processing (DLP) uses micro-mirrors that are arranged in a matrix on a semiconductor chip. The mirrors refer to pixels on the screen and the number of mirrors is proportional to the resolution of the screen.



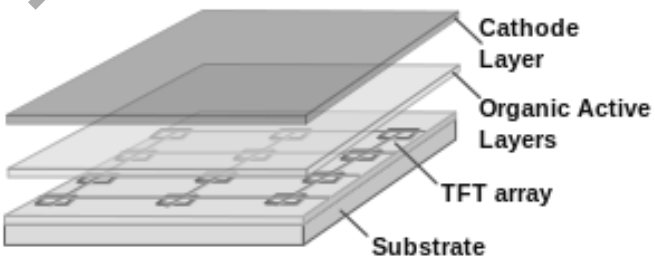
Liquid Crystal on Silicon (LCoS) is a beautiful merger of LCD and DLP. LCoS makes use of both Liquid crystals as well as mirrors. The Liquid crystals monitors the amount of light that is applied to the reflective mirrors. Three LCoS chips red, green and blue adjust the amount of light to appropriate color channels. The intensity of voltage determines the amount of light allowed to pass through the crystals. LCoS TV sets deliver Super-eXtended Graphics Array (SXGA) or higher resolution i.e. 1400 x 1050 pixels which make them expensive than LCDs or DLPs.



Light Emitting Diode (LED) Displays have an array of LEDs arranged as pixels on a flat screen.There are two types of LEDs: Conventional and Surface-Mounted device(SMD).Red, green and blue diodes are grouped together to form a colourful pixel on the screen.

AMOLED (active-matrix organic light-emitting diode) is a display technology for use in mobile devices and television. OLED describes a specific type of thin-film-display technology in which organic compounds form the electroluminescent material, and active matrix refers to the technology behind the addressing of pixels.

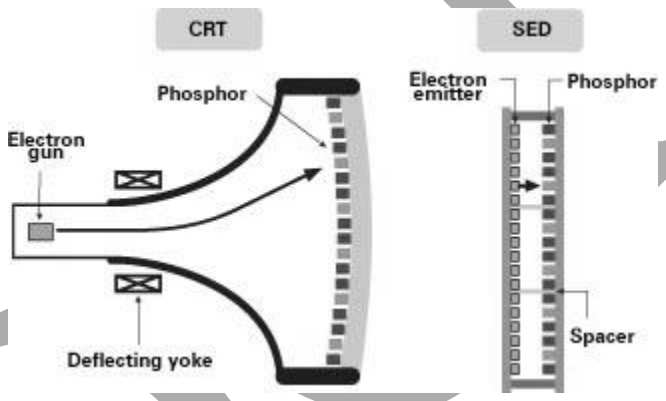
An AMOLED display consists of an active matrix of OLED pixels that generate light (luminescence) upon electrical activation that have been deposited or integrated onto a thin-film-transistor (TFT) array, which functions as a series of switches to control the current flowing to each individual pixel.



Illustrationof active-matrix OLED display

Surface Conduction electron-emitter Display (SED) has an array of electron emitters with a vacuum separated phosphor screen.

A SED consists of a matrix of tiny cathode ray tubes, each "tube" forming a single sub-pixel on the screen, grouped in threes to form red-green-blue (RGB) pixels. SEDs combine the advantages of CRTs, namely their high contrast ratios, wide viewing angles and very fast response times, with the packaging advantages of LCD and other flat panel displays. They also use much less power than an LCD television of the same size.



Comparison of CRT and SED

2.1.4 INPUT TECHNOLOGY: PORTABILITY AND NEW INTERACTION MODES

Pointing and clicking as a means to input data has been around for a long time. But now-a-days, Touch screens are ruling the roost.

Touchscreens are not only commonly used in devices like game consoles and smartphones but also in medical arenas , automated teller machines (ATMs), and kiosks such as Automatic Ticket Vending Machines (ATVM) as well as museum displays to allow quick interaction between the device and the user.

We can use pen computing to write data using a stylus pen that gets digitized. Instead of any paperwork to show proof of delivery, an electronic signature capture allows the delivery personnel to obtain a signature from a customer, by signing the screen on the smartphone or mobile device with a finger or a stylus pen.

2.2 MULTIMEDIA APPLIANCES: HYBRID DEVICES

Hybrid Devices are combination of two or more devices having distinct characteristics that morph together and blur the lines of differentiation between devices.

2.2.1 COMPUTER-BASED VIDEO PHONES:

The best example of a device that has undergone hybridization is the telephone. It is integrated into fax machines, answering machines, Video phones etc. AT & T created a videophone that depends on telephone lines and encoding and decoding of audio and video and complex modem technology.

Examples of Hybrid Approach

Conventional Devices	Portable Versions	Hybrid Function
Telephone	Cellular phones	Video phones
Computers	Laptops, palmtops	Computer-based telephony
Televisions	TV Walkmans	Interactive TV

A Video phone is a telephone with a video display so it is possible to transfer audio as well as video simultaneously in real-time. Videophone service provided the first form of videotelephony, later to be followed by videoconferencing, webcams, and finally high-definition telepresence. Increased bandwidth and improved technology has made it possible to perform video call for multiple people.

A webcam is a video camera connected to the computer that can stream audio and video in real time and can be used with software clients like Cisco: WebEx, Skype, ooVoo etc for video calls and videoconferencing.

A videoconference system allows many participants at various locations to conduct a conference using a Multipoint Control Unit (MCU) to bridge the connections.

A telepresence system is a high-end videoconferencing system and service usually employed by enterprise-level corporate offices. Telepresence conference rooms use state-of-the art room designs, video cameras, displays, sound-systems and processors, coupled with high-to-very-high capacity bandwidth transmissions.

2.2.2 COMPUTER-BASED CONSUMER ENTERTAINMENT PRODUCTS

Entertainment products are a package of innumerable utilities combined together perfectly to please the user's senses. Such devices support customization as well as have the ability to communicate with other devices.

Gaming Systems like Xbox One offers a variety of applications built into it. It has live television that works with your existing set-top box, split-screen multitasking of

apps. The kinect sensor allows the user to take motion tracking to the next level. It also has a voice recognition feature. It can also view and play content from USB devices using Media Player applications.

Home Theatre PC or HTPC or media center computer is a convergence device that combines some or all the capabilities of a personal computer with a software application that supports video, photo, audio playback, and sometimes video recording functionality. In recent years, other types of consumer electronics, including gaming systems and dedicated media devices have crossed over to manage video and music content. The term "media center" also refers to specialized application software designed to run on standard personal computers.

The media itself may be stored, received by terrestrial, satellite or cable broadcasting or streamed from the internet. Stored media is kept either on a local hard drive or on network attached storage. Some software is capable of doing other tasks, such as finding news (RSS) from the Internet.

Beyond functioning as a standard PC, normally HTPCs have some additional characteristics:

- Television connectivity

Standard PC units are usually connected to a CRT or LCD display, while HTPCs are designed to be connected to a television. All HTPCs should feature a TV-out option, using either an HDMI, DVI, DisplayPort, Component video, VGA (for some LCD televisions), S-Video, or Composite video output.

- Remote control

Integrating a HTPC into a typical living room requires a way of controlling it from a distance.

Many TV tuner/capture cards include remote controls for use with the applications included with the card. Software such as Boxee, GB-PVR, SageTV, MediaPortal and Beyond TV support the use of Windows MCE and other remote controls. Another option is an in-air mouse pointer like the Wii Remote, GlideTV Navigator, or Loop Pointer which gives cursor control from a distance. It is also possible to utilize common wireless keyboards and other peripherals to achieve the same effect (though the range may not be as long as a typical remote control).

Some HTPCs, such as the Plex / Mac Mini combination, support programmable remote controls designed for a wide range of typical home theater

devices. More recent innovations include remote control applications for Android and Apple iOS smartphones and tablets.

- External and networked storage devices

Because of the nature of the HTPC, higher than average capacities are required for HTPC units to allow storage of pictures, music, television shows, videos, and other multimedia. Designed almost as a 'permanent storage' device, space can quickly run out on these devices. Because of restrictions on internal space for hard disk drives and a desire for low noise levels, many HTPC units utilize a NAS (Network Attached Storage) device, or another type of network connected file server.

- TV tuner cards

A TV tuner card is a computer component that allows television signals to be received by a computer. Most TV tuners also function as video capture cards, allowing them to record television programs onto a hard disk. Several manufacturers build combined TV tuner plus capture cards for PCs. Many such cards offer hardware MPEG encoding to reduce the computing requirements. Some cards are designed for analog TV signals such as standard definition cable or off the air television, while others are designed for high definition digital TV.

- Network TV Tuner

A network TV Tuner or TV gateway is a TV server that converts TV signal from Satellite, Cable or Antenna to IP. With multiple TV tuners the TV Gateway can stream multiple TV channels to devices across the network. Several TV Gateway manufacturers build the device to stream the entire DVB stream, relying on the host player device to process the feed and to capture/record, while other devices such as VBox Home TV Gateway provide a variety of option from full PVR and live TV features, to streaming of specific DVB layers to support less powerful devices and to save network bandwidth.

- Quiet/minimal noise

A common user complaint with using standard PCs as HTPC units is background noise, especially in quieter film scenes. Most personal computers are designed for maximum performance, while the functions of a HTPC system may not be processor-intensive. Thus, passive cooling systems, low-noise fans, vibration-absorbing elastic mounts for fans and hard drives, and other noise-minimizing devices are used in place of conventional cooling systems.

HTPC options exist for each of the major operating systems: Microsoft Windows, Mac OS X and Linux. The software is sometimes called "Media Center Software".

2.2.3 SOME MORE HYBRID EXAMPLES

Newton Speak and Spell was the first Learning application to feature a voice synthesizer for children.

Apple watches are the best example today of a Hybrid Device. It has a plethora of other functions other than being an accurate timepiece. It enables users to send a tap, sketches or heartbeat to people in touch. It also gives personalized suggestions about fitness by monitoring your daily activities. It supports navigation by syncing Wifi and GPS from the iPhone.

Videoconferencing that is a huge collaboration solution is being provided by Cisco, Skype and Apple (FaceTime) which is another example of Hybridization.

Several hardware companies have build hybrid devices with the possibility to work with both the Windows 8 and Android operating systems thus giving the user a Hybrid OS operation.

In mid-2014, Asus released a hybrid touchscreen Windows tablet/laptop with a detachable Android smartphone; when docked to the back of the tablet/laptop display, the Android phone is displayed within the Windows 8 screen, which is switchable to Android tablet and Android laptop.

2.3 A DESIGNER'S PERSPECTIVE

2.3.1 INDUSTRIAL DESIGN CONSIDERATIONS:

The very first thing that needs to be taken into consideration is how user friendly the appliance is. The role of an industrial designer is to create and execute design solutions for problems of form, usability, physical ergonomics, marketing, brand development, and sales.

2.3.2 HUMAN FACTORS AND THE DESIGN OF NEW PRODUCTS:

Human factor is a vital contributor to new product design. The approach to starting new development is based on what already exists before. There are many questions about the future of multimedia devices. But priority is to be given to make technology easier to use beyond various boundaries.

2.4 INDUSTRY PERSPECTIVES FOR THE NEXT DECADE

2.4.1 TELECOMMUNICATIONS:

The Telecommunication industry is one of the rapidly changing sectors, the main reason being increased bandwidth rates. There will be a boost in Cloud Services. Cloud is being used for storage, processing and Software as a Service (SaaS).Big Data technology will be harnessed to analyze and classify the huge amount of unstructured data.

2.4.2 ENTERTAINMENT:

Computer Industries have taken entertainment to the next level. In this digital age, Entertainment and multimedia go hand in hand.TV will go obsolete as smartphones and mobile devices make it convenient to watch media content wherever they want. The technology Digital Multimedia Broadcasting (DMB) also known as Mobile TV is gaining popularity day by day. Movies, TV and creative content now have a strong dependency on technology which keeps growing every day. Hollywood will rely immensely on computer bigwigs like Apple, Microsoft and other major Software Companies to entertain the digital generation.

2.4.3 INFORMATION SERVICES:

Home Shopping-Newspapers have become second nature to us in this digital age. Consumer demands for quick and summarized information need to be satisfied with appropriate classification of Big Data. Huge amount of varied unstructured content comprising of text, audio as well as video needs to be filtered to serve the concise needs of various segments.

2.4.4 COMPUTER INDUSTRY:

As Technology is becoming portable and affordable, computers have become a part and parcel of our lives. Companies like Apple and IBM come together to create customer centric applications. Computer Giants are ruling the marketplace due to huge demand of smart gadgets versus mere electronic companies.

Networking will be seamless and transparent with seamless connectivity to networks irrespective to devices or location.Perfect image recognition,voice recognition and motion tracking will revolutionize how we interact with computers.

3-D Printing is changing the world by enabling DIYers to create objects on demand.It is empowering them to design their creations which would have been expensive without its use.Nanotechnology will be a boon for patients around the world by introducing digestible cameras that take high-definition photographs of the intestines.PillCam will become widely used throughout the world for perfect colonoscopies for thorough diagnosis in complex medical cases.

Robotics in the field of medicine will be a lifesaver from performing accurate surgeries to telemedicine and prostheses.

Smart cities, self-driven electric cars, smarter and faster rechargeable battery technology and seamless networks indicate a promising future for the Computer Industry.

2.5 KEY CHALLENGES AHEAD: TECHNICAL, REGULATORY, SOCIAL

The biggest technical challenge was to fit so many transistors on a single integrated chip. If this challenge wouldn't have been achieved, performing any task efficiently would have been impossible. The next technical obstacle is to make technology available at affordable prices to everyone around the world. It absolutely makes no sense only if a selected few are able to gain access to it. Another major obstacle is to make the product compatible with every platform throughout the world.

Another hurdle is getting past government rules and regulations. Government rules have a major impact on the growth and progress of technology. Different government regulations throughout the word should be collectively supportive of technological progress.

If we assume that all technical challenges are defeated and governments of different countries show full co-operation to promote development of hi-tech Devices, there could be social barriers stopping people from making that newly developed technology a part of their lives. People may feel that their privacy is being invaded and there is fear among the people that their personal details on the web is not secure and might even be sold.

2.6 SUMMARY

- In his 1965 paper, Gordon Moore described that over the history of computing hardware, the number of transistors in a dense integrated circuit doubles approximately every two years.

- Use of fiber optics proved to be a breakthrough as it was a replacement of the existing wiring and more data could be sent over it.

- The newer types of displays include Plasma Display Panel (PDP), Liquid Crystal Display (LCD), Digital Light Processing (DLP), Liquid Crystal on Silicon (LCoS), Light Emitting Diode (LED), Surface Conduction electron-emitter Display (SED).

- Hybrid Devices are combination of two or more devices having distinct characteristics that morph together and blur the lines of differentiation between devices.

- The role of an industrial designer is to create and execute design solutions for problems of form, usability, physical ergonomics, marketing, brand development, and sales.

2.7 UNIT END EXERCISES

Reference:

I. Multimedia Systems, John F. Koegel Buford, Pearson Education.

II. Wikipedia

III. www.htxt.co.za

IV. electronics.howstuffworks.com

V. http://www.wired.com

❖❖❖❖

3

Architectures and Issues in Distributed Multimedia Systems

Unit Structure

3.1 Objectives
3.2 Introduction
3.3 Intramedia synchronization, Intermedia synchronization
3.4 Distributed Systems and Multimedia Systems
3.5 Synchronization, orchestration, and QOS Architecture
3.6 The role of Standards
3.7 A Framework for Multimedia Systems
3.8 Summary
3.8 Unit End Exercises
3.9 Additional Reference

3.1 OBJECTIVES

In this unit, you will learn:

1. Synchronization of multimedia in different forms
2. A Framework for multimedia systems

3.2 INTRODUCTION

The need for multimedia data types in existing computing and communications systems is leading to a rethinking of the design of these systems to accommodate new requirements for performance and application abstractions. Further, the scope of distributed systems is no longer focused on the LAN topology; instead, global scale networks are envisioned which connect nodes with telephony as well as computing functionality. The structure of these multimedia computing and communication systems is complicated by the range of system facilities that are impacted. A partial list includes: distributed object management facilities, hypermedia document architectures, multimedia interchange

formats, scripting languages, media formats, application toolkits, operating system services and network protocols and architectures.

This chapter presents a framework for organizing and inter-relating these activities. The framework is a high-level model used to discuss current status, direction and open issues as a precursor to the remaining chapters.

3.3INTRAMEDIA SYNCHRONIZATION, INTERMEDIA SYNCHRONIZATION

The time-sampled nature of digital video and audio, referred to as *isochronous* data, requires that delay and jitter be tightly bounded from the point of generation or retrieval to the point of presentation. This requirement is referred to as *intramedia synchronization*.

If several continuous media streams are presented in parallel, potentially from different points of generation or retrieval, constraints on their relative timing relationships are referred to as *intermedia synchronization*.

Both types of synchronization require coordinated design of the resource managers so that end-to-end synchronization can be met.

3.4DISTRIBUTED SYSTEMS AND MULTIMEDIA SYSTEMS

Multimedia computing and communication systems provide mechanisms for end-to-end delivery and generation of multimedia data that meet QOS requirements of applications. *Distributed multimedia systems add capabilities such as global name spaces, client/server*

computing, global clocks, and distributed object management. Such facilities enable the sharing of resource over a larger population of users. With the technology for multimedia computing, distributed services will be feasible over a wide area using broadband networks.

3.5 SYNCHRONIZATION, ORCHESTRATION, AND QOS ARCHITECTURE

A fundamental requirement for multimedia systems is to provide intramedia and intermedia synchronization. The intermediate subsystems involved in delivering a stream may introduce delay, jitter, and errors. Along any path these values are cumulative, and it is the cumulative delay, jitter, and error rate that must be managed to achieve the end-to-end QOS requirements.

The management of collections of resource managers to achieve end-to-end synchronization is referred to as ***orchestration***. QOS parameters are considered to be a basic tool in orchestration.

The definition of QOS parameters which permit system-wide orchestration is referred to as a ***QOS architecture***.

3.5.1 Synchronization

Synchronization is the coordinated ordering of events in time, and various mechanism and formalisms for synchronization have been developed, ranging from low-level hardware-based techniques to abstractions for concurrent programming languages.

Systems using continuous media data do not require fundamentally new synchronization primitives, but do require consideration of two aspects of multimedia applications

1. synchronization events have real-time deadlines, and
2. failure to synchronize can be handled using techniques such as frame repetition or skipping such that the application can still continue to execute.

For a single media element which has a presentation deadline t_p, if the maximum end-to-end delay due to retrieval, generation, processing, transmission, etc., is D_{max}, then the scheduling of the presentation steps must begin by time t_p-D_{max}. If the media object is a stream of elements, not necessarily isochronous, with deadlines $\{t_{pl}, t_{p2}, t_{p3},...\}$, then the scheduling problem becomes meeting the sequence of deadlines $\{t_{p1}$-D_{max}, t_{p2}-D_{max}, t_{p3}-D_{max},...} for each object being presented. Any admissibility test which is to satisfy the synchronization requirement must consider the delay requirements of the application, i.e., $D_{req}<D_{max}$. If the average delay experienced per media element, $D_{avg'}$ is less than D_{max}, then additional capacity exists to schedule other media objects, though with increased probability of failure.

If elements arrive prior to the presentation deadline D_{max}, due to variations in system latencies, buffering is required to hold the element in reserve until time t_{pi}. Due to the deadline specification, data errors in retrieval or transmission may not be correctable via re-retrieval or retransmission. Acceptable error rates are application and media dependent. In order to meet the requirements of schedulability of a continuous media stream, each subsystem must provide a maximum delay with some probability p. Further, in order to limit buffering

requirements, the variation in delay, referred to as jitter, must also be bounded.

3.5.2 Orchestration or Meta-Scheduling

Each resource manager includes a scheduling function which orders the current requests for servicing so as to meet the required performance bounds. For example, a continuous media file system schedules storage system access operations, and the network layer schedules traffic to the transport layer. An application requires the coordinated operation of these scheduling functions if end-to-end performance bounds are to be met. An approach to coordinating resource scheduling of the various systems is to add a layer between the application and the resource managers for orchestration or meta-scheduling.

3.5.3 QOS Architecture

Quality of Service (QOS) is used in the OSI reference model to allow service users to communicate with network service regarding data transmission requirements. In OSI, QOS is specified using a number of parameters which can be grouped into three sets: single transmission, multiple transmission, and connection mode.

QOS parameters include transit delay, residual error rate and throughput. For connection-oriented service, QOS parameters required by a network service user are specified in the request for connection.

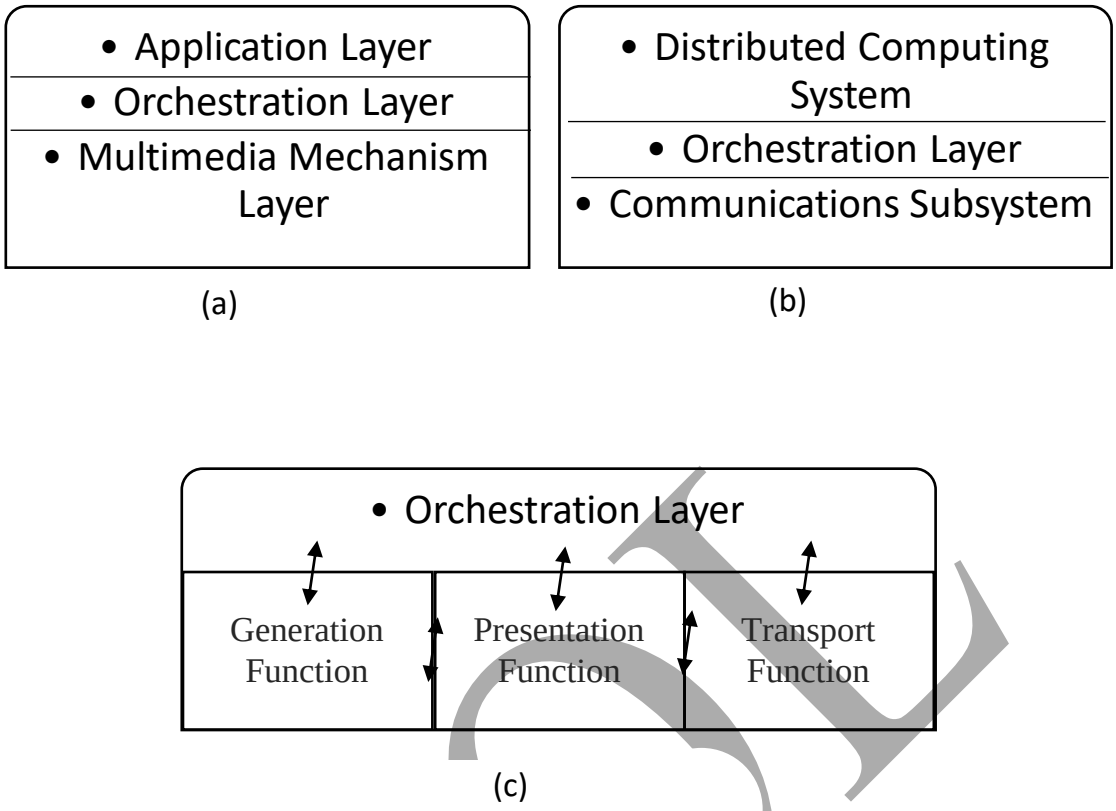


Figure 3.1: Real-time multimedia system architecture:

(a) The orchestration layer as a middle layer between the application and multimedia system services

(b) The orchestration layer as a service for providing synchronization for a distributed computing system

(c) The orchestration function, with vertical arrows indicating control paths and horizontal arrows indicating data paths.

3.6THE ROLE OF STANDARDS

In order to achieve universal access and distribution in a distributed multimedia system, interoperability between components of the system is very necessary. Interoperability has a direct dependency on communication standards, data format and services. The varied and diversified standardization requirements are illustrated in the below figure.

The below figure represents components of future telecommunication infrastructure. The table below shows the standards classified on the basis of use and displays the broad spectrum of standardization.

Standards facilitate in fulfilling interoperability needs. Development of Compatible Standards is a need for overall system-wide usage.

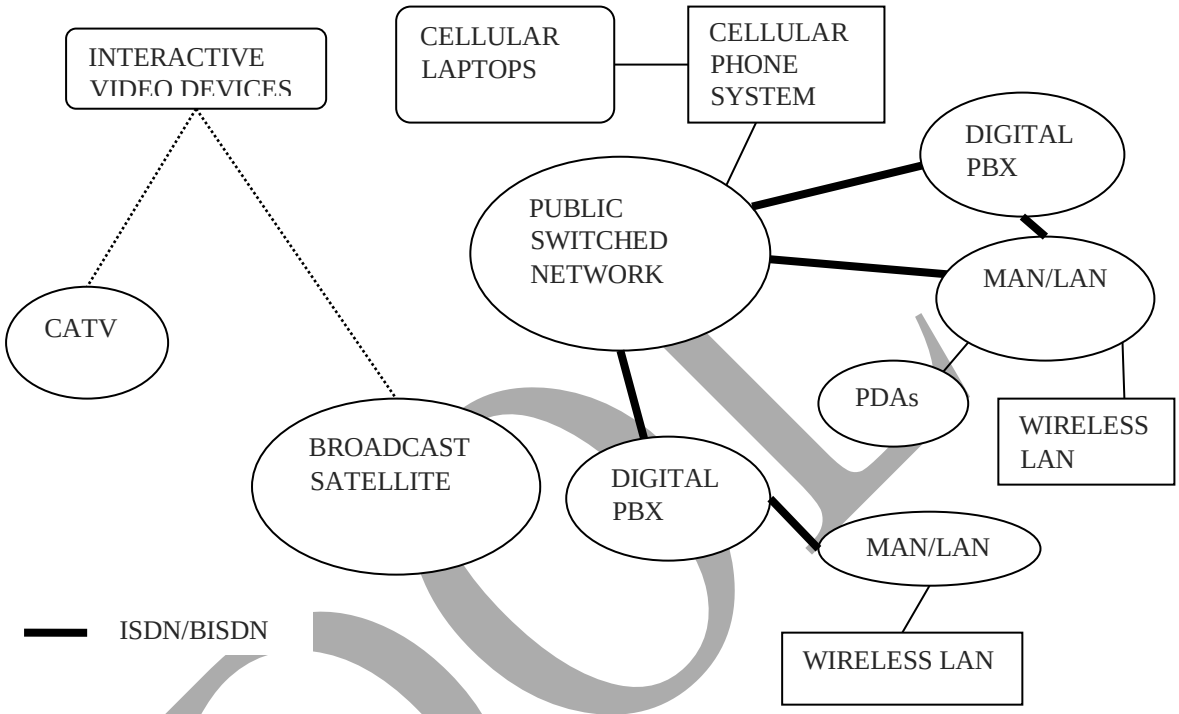


Figure3.2 :Intersection of telephony with computer communications and television networks will involve both ISDN and BISDN

User	Function	ISO/ITU	Trade Group	Vendor
AUTHOR	Scripting Language	SMSL	IMA RFT	Kalieda Labs ScriptX GainMomentum GEL
	HyperMedia Document Architecture	SGML,HyTime ODA/HyperODA		
DEVELOPER	Distributed Object Arch. UI Toolkits	ODP/ANSA PIKS PREMO	OMG CORBA X Consortium XIE COSE	Microsoft OLE Apple QuickTime Microsoft MME
SYSTEM VENDOR	Multimedia System Services		IMA RFT COSE UNIX Intl	
	Multimedia Mail Interchange Format	MHEG	IETF MIME IMA RFT OMFI	Apple QuickTime Movie File Format Microsoft AVI
NETWORK PROVIDER	Multiservice Network	ATM FDDI-II	IEEE 802.6	
	Protocol Stack	OSI	IETF RTP	
PUBLISHER	Storage Formats	9660	Rock Ridge	Kodak PhotoCD Philips CD-I
	Media Formats	MPEG,MPEG-2,-4 JPEG H.261	MMA MIDI	Intel DVI

Table: Categorization of Multimedia Standards

3.7 A FRAMEWORK FOR MULTIMEDIA SYSTEMS

The framework presented here provides an overall picture of the development of distributed multimedia systems from which a system architecture can be developed. The framework highlights the dominant feature of multimedia systems: the integration of multimedia computing and communications, including traditional telecommunications and telephony functions.

Low-cost multimedia technology is evolving to provide richer information processing and communications systems. These systems, though tightly interrelated, have distinct physical facilities, logical models, and functionality. Multimedia information systems extend the processing, storage, and retrieval capabilities of existing information systems by introducing new media data types, including image, audio, and video. These new data types offer perceptually richer and more accessible representations for many kinds of information. Multimedia communication systems extend existing Point-to-Point connectivity by permitting synchronized multipoint group communications. Additionally, the communication media include time-dependent visual forms as well as computer application conferencing.

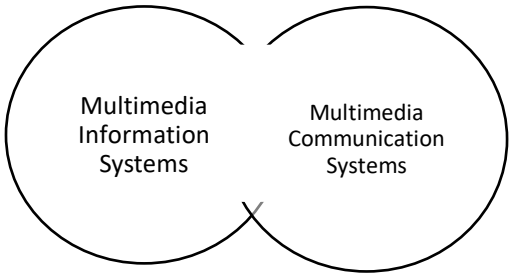


Figure 3.3 Multimedia technology is facilitating the convergence of multimedia information processing systems and multimedia communications systems.

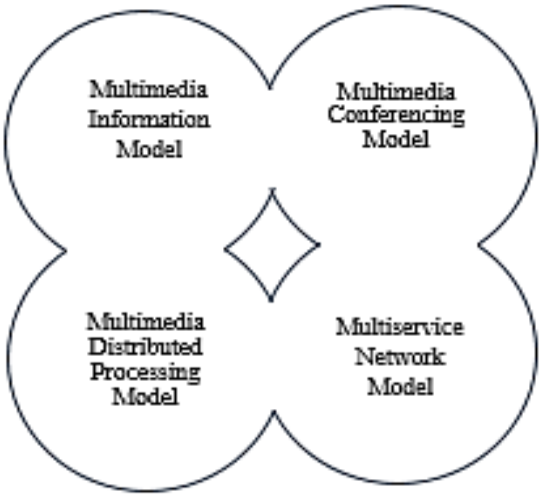


Figure 3.4 The framework consists of four interrelated models. The information and distributed processing models constitute the Multimedia Information System (MMIS). The conferencing and multiservice network models form the Multimedia Communications System(MCS).

3.7.1 Multimedia Distributed Processing Model

A layered view of the multimedia distributed processing model is shown in Figure 3.5. Models similar to this have been published by the Interactive Multimedia Association in its Architecture Reference Model and UNIX International's Open Distributed Multimedia Computing model. Each layer provides services to the layers above. Significant additions to the facilities of traditional computing environments include (from the top):

Scripting languages: Special-purpose programming languages for controlling interactive multimedia documents, presentations, and applications.

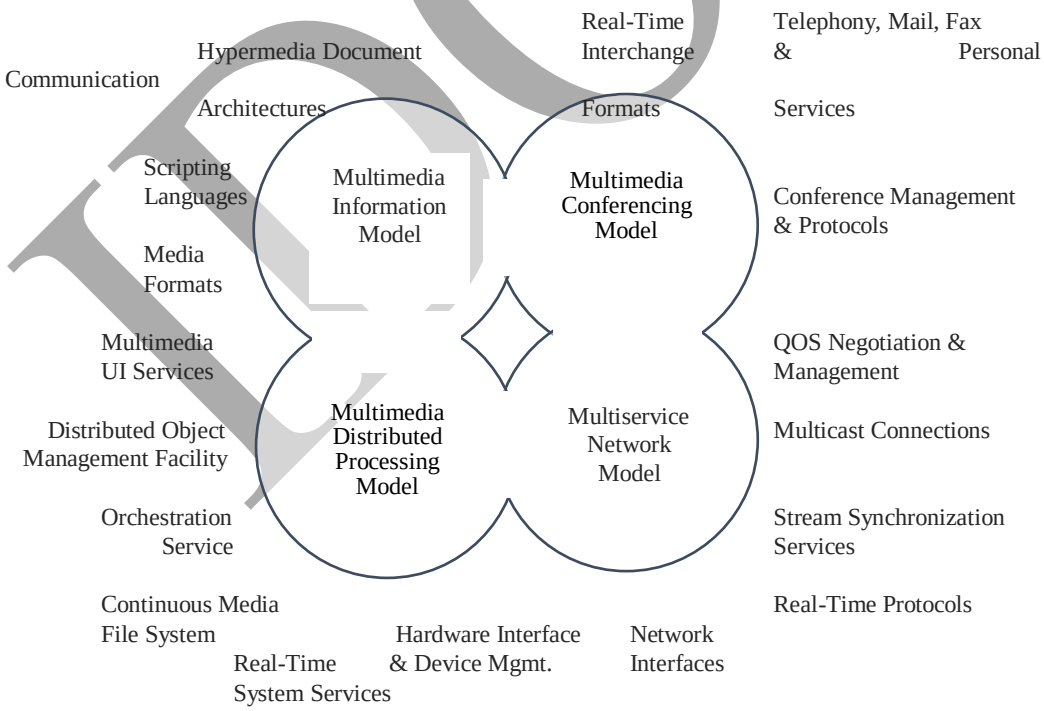


Figure 3.5: Each of the four models of the distributed multimedia systems framework specifies various components. Example components, which might be services, formats, and/or APIs are shown in the periphery of the corresponding models.

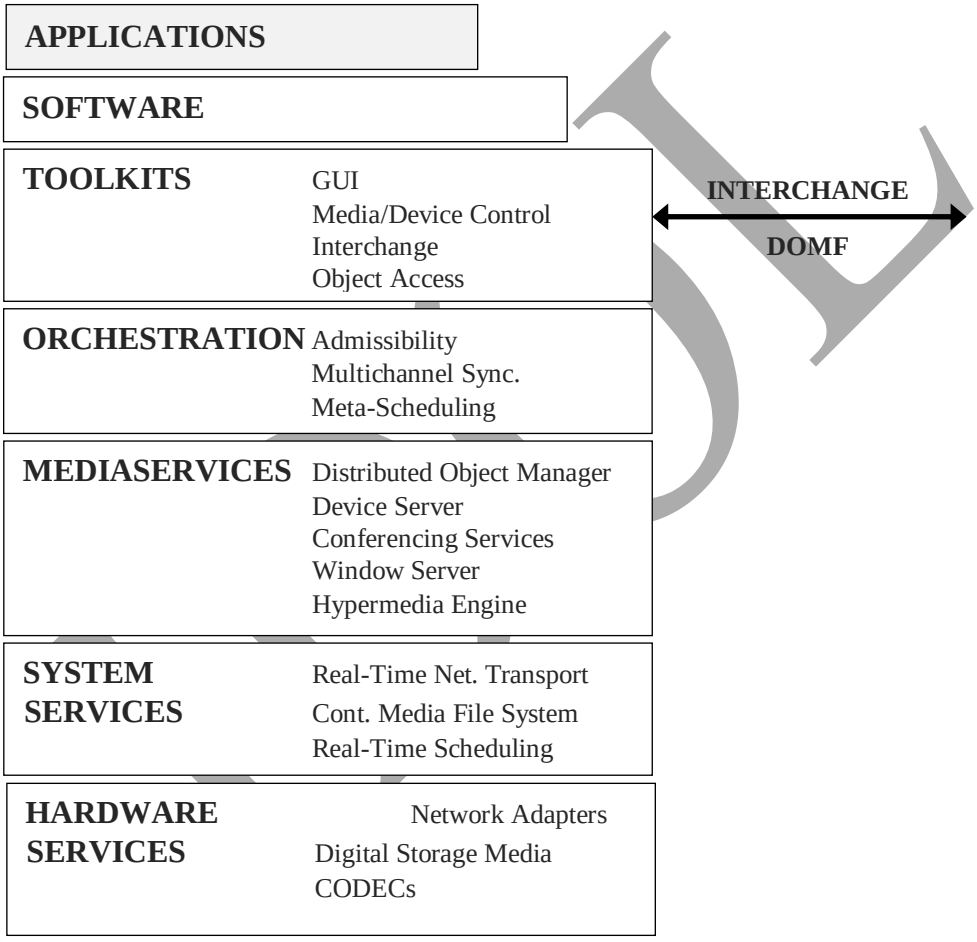


Figure 3.6 Multimedia distributed processing model: a layered view of a distributed environment

Media device control: A combination of toolkit functions, programming abstractions, and services which provide application programs access to multimedia peripheral equipment.

Interchange: Multimedia data formats and services for interchange of multimedia content.

Conferencing services: Facilities for managing multiparty communications using high-level call model abstractions.

Hypermedia engine: A hypermedia object server that stores multimedia documents for editing and retrieval.

Real-time scheduler: Operating system process or thread scheduling so as to meet real-time deadlines.

3.7.2 MULTIMEDIA INFORMATION MODEL

The second component of the framework relates to the abstractions and data models for organizing multimedia documents, presentations and other information. While there is no universal hypermedia document model, the ability to include multimedia content and describe associations between document components (hyperlinking) is considered basic. Support for temporal relationships between components, interactivity, and components with active behaviour is also considered important.

While several hypermedia document models exits, further work is needed in characterizing the logical structure of multimedia documents (Table below). This issue is related to the authoring process for creating multimedia information, a process which has many forms and tools. The related issues of hypermedia document models and multimedia authoring are discussed in later chapters.

Text Document	Multimedia Document
Book	?
Chapter	?
Section	?

Paragraph	?

Table: Logical models for text documents are well understood; standards such as SGML and ODA provide languages for specifying such models. By comparison, multimedia documents have a relatively short history and small population by which to define logical models.

3.7.3 MULTISERVICE NETWORK MODEL

Networks for distributed multimedia systems must support a wide range of traffic requirements, including traffic with real-time requirements. Such networks are described as *multiservice*. The requirements for the network architecture include QOS guarantees that are sufficient for real-time transport, multiway connections, and high performance. Existing network architectures such as OSI and TCP/IP do not satisfy these requirements. Efforts in designing multiservice networks are being carried out for both the public switched network and the Internet.

Public Switched Network

During the past decade, the telecommunications industry has developed roadmap for the evolution of digital switched communication service based on ISDN (Integrated Services Digital Network). ISDN standardizes connection interfaces, transmission protocols and services.More recently, initial recommendations for Broadband ISDN (BISON) have been adopted. Unlike ISDN, which is a digital circuit switching network, BISDN uses cell relay or asynchronous transfer mode (ATM). ATM is suitable for very high speed switching of fibre optic transmission networks. BISDN will be upwardly compatible with ISDN, leading to a global high-speed network suitable for high-speed

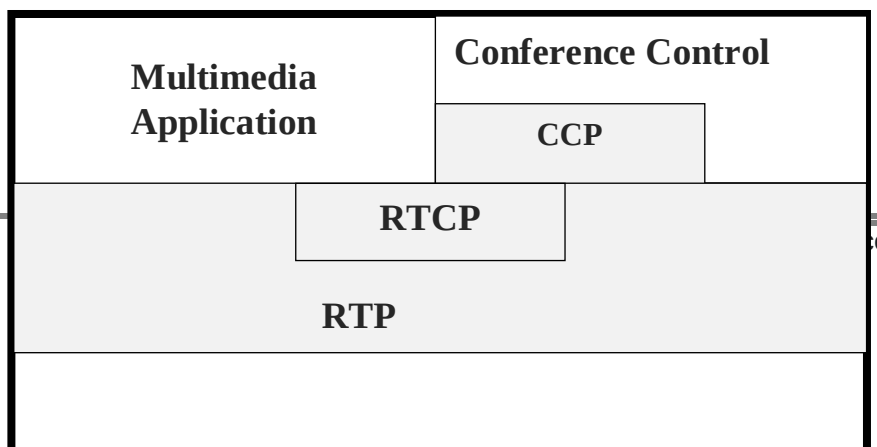
multimedia traffic With Common service definitions throughout the switching components.

Internet

Early experimental work on supporting real-time traffic for multimedia conferencing over the Internet included the development of a revised experimental Internet Stream Protocol commonly called ST Il. ST IIcreates simplex tree reserved-bandwidth connections from an origin to the specified destinations. It is at the same protocol level as IP. ST Il has been used for experiments in video conferencing over the Internet.

More recently, several Internet working groups have been formed to address the requirements of multimedia conferencing and real-time traffic. The Audio-Video Transport Working Group is developing a transport protocol for real-time applications, RTP. RTP uses the end-to-end transport services such as TCP, UDP, or ST Il. RTP uses timestamps to provide playout synchronization between a source and a set Of destinations. Multiple media and conferences can be multiplexed over one connection and de-multiplexed at a destination. RTP is intended for use by conference control protocols such as CCP.

The Internet work is still in the experimental stages. Issues which will be of interest in the near future include scalability of current research systems, accounting mechanisms based on usage, and general use of the Internet for telephony.



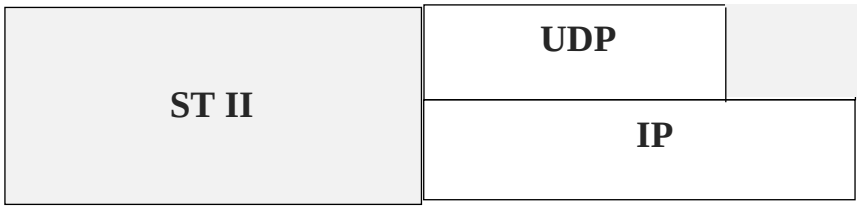


Figure 3.7 Experimental Internet protocol stack for real-time transport and multimedia multiparty conferencing

3.7.4 MULTIMEDIA CONFERENCING MODEL

Trends in communications and computing are leading to convergence of computer-based telephony and communications functions. Existing network architectures such as the OSI reference model and TCP/IP were not designed with the intent of supporting real-time multiparty conferencing. Today's computer-based communications applications, such as mail systems and shared windowing systems, are typically monolithic entities that are not designed to be integrated with other applications.

New software architectures for computer-based applications, such as those described in Chapter IS, will enable applications generally to access and control communications services. In contrast to traditional telecommunications where communications services are accessed separately, it is expected that in the future the majority of computer applications will support communications access, and the dedicated communication application will be the exception rather than the rule.

3.8SUMMARY

- A QOS architecture can offer a unified view of resource management to the application and the various resource managers.

- The two major functions enabled by multimedia systems are unambiguous: richer information processing and delivery and real-time multimedia-based multiway communication.

- Kurose has surveyed QOS and identified 4 approaches: Tightly Controlled Approach, Approximate Approach, Bounding Approach and Observation-based Approach.

- Standards facilitate in fulfilling interoperability needs. Development of Compatible Standards is a need for overall system-wide usage.

- Multimedia System's framework represents integration of Multimedia computing and communication inclusive of telecommunication and telephony.

- The Framework can be divided into 4 parts: Multimedia Distributed Processing Model, Multimedia Information Model, Multiservice Network Model and Multimedia Conferencing Model.

3.9UNIT END EXCERCISES

1. What is QOS?

2. What are the components related to Multimedia Distributed Processing Model?

3. Write a note on Multimedia Information Model.

4. What are the roles of Public switched network and Internet in Multiservice Network Model?

5. Write a difference between intramedia synchronization and intermedia synchronization.

3.10ADDITIONAL REFERENCE

1. MULTIMEDIA SYSTEMS, John F. Koegel Buford, University of Massachusetts Lowell, Pearson, Fourteenth Impression

4

Digital Audio Representation and Processing

Unit Structure

Objectives
Introduction
Uses of Audio in Computer Applications
Psychoacoustics
Digital Representation of Sound
Transmission of Digital Sound
Digital Audio Signal Processing
Digital Music-Making
Speech Recognition and Generation
Digital Audio and the Computer
Summary
Unit End Exercises
Additional Reference

OBJECTIVES

In this unit, you will understand:

1. Psychoacoustics and its types
2. Importance of digitization in sound

INTRODUCTION

This is the first of several chapters to provide important background information on basic media. This chapter is devoted to audio, with specific discussion of music and speech, and reviews fundamentals necessary for developing tools, applications, and interfaces involving

audio. The survey includes physical aspects of sound, psychoacoustics, the nature of musical sound, stereophonic and quadrophonic sound, and audio processing building blocks. Standard formats for digital audio are described, as well as the MIDI protocol for music synthesizers. Algorithms and architectures for data compression, music synthesizers, speech recognition, and speech generation are reviewed.

USES OF AUDIO IN COMPUTER APPLICATIONS

The initial interfaces to digital data processing equipment were visually oriented. The operator turned switches or fed punched cards; audio interfaces came later. Early anecdotal references can be found to late-night researchers playing songs using the bells and whistles on teletypes and printers of a mainframe. Others used radio frequency static generated by digital equipment to drive radios placed on the top of the computer cabinet.

The first serious work involving the computer and sound carne at labs, where John Pierce and Max Mathews Ill pioneered digitized speech work that evolved into computer-generated sound. Computers have been adopted in the music community for composition, printing, and data processing (e.g., insipid catalogs), but those applications are not discussed here. In the music industry, digital technology has all but supplanted analog technology for synthesis and is making major inroads in sound storage and processing. Meanwhile, the loudspeaker and microphone remain analog.

Outside the music and audio industries, audio has been increasingly associated with digital technology; a few examples are given here.

Rosenberg discusses the computer-supported cooperative work environment, including teleconferencing. Voice mail is now a common adjunct to electronic mail. The popularization of multimedia brings with it the embedding of audio information in hyperdocuments, supported by storage media such as CD-ROM. With optical character recognition now commonplace on personal computers for encoding incoming faxes, one can envision faxes read aloud by the computer. Telephone voice-response systems allow those in the field to retrieve or modify information in a database at the home office, Speaker recognition and identification provides a new level of security for access to sites, equipment, and data. Digit recognition systems are now widely used by telephone companies to help with automating information requests or collect call billing.

Sonification is a relatively new and relevant field that deserves some explanation even in this brief introduction. Sonification involves mapping the parameters of sound to one or more variables of a set of data. In one implementation, an independent variable is treated as musical time. As the independent variable increases in value, the data points indexed by the independent variables are performed, so to speak. In another implementation, the user Sweeps the mouse over a graphic representation of the data points. That graphic representation may itself incorporate information about more than one variable. But as the mouse is swept, a more or less musical note is sounded for each data point encountered. The parameters of the note, including attack time, decay time, spectrum, amplitude, and frequency, amount of reverberation, apparent elevation, and left-right mix can be varied. In some cases, each parameter in the data space is mapped onto a different attribute of sound. In other cases, several attributes of sound are mapped onto the same parameter of the data space on purpose. The interplay of

redundant parameters helps make it possible for human perceptual and cognitive mechanisms to find structure in the data.

One of the disadvantages of audio compared with other media is the ephemeral nature of sound. A painting by Cézanne or a pie graph showing profit and loss for each corporate division can be placed on the wall for enjoyment and/or study. The eye can wander over the image at will, easily revisit those parts already viewed, and in general choose freely between a micro and macro view. The visual mode also allows the viewer to integrate largest amounts of data into a whole. With music as an art form and with sound as a communications medium, the listener must pay attention and remember what has already passed. For example, with audio as a component of the user interface, the alternative to relying on memory is to keep resounding background information. But if too many things are constantly in the audio background, cacophony results. For most people, fatigue sets in as well; a constant background sound is no longer consciously perceived.

Audio is an important communication channel and, in some cases (e.g., involving human emotions), the most appropriate and efficient. We all know the richness of the subtle cues transmitted by inflection in the voice. Audio provides three-dimensional cues (e.g., in the back of the head) that are not available to other human sensory systems.

This chapter concentrates on certain important aspects of audio as related to digital technology. The selection of topics is motivated by relevance to implementation of audio in digital systems, especially workstations, multimedia systems, and computer-based conferencing systems. On the other hand, topics such as memory of auditory experiences or speaker identification by humans are largely ignored.

PSYCHOACOUSTICS

We first turn to the interaction between physical stimuli and the human nervous system. For this discussion, audio is defined as a disturbance in air pressure that reaches the human eardrum. In terms of frequency, amplitude, time, and other parameters, there are limits to the kinds of air pressure disturbance that will evoke an auditory percept in humans. The development of the human ear's limitations and capabilities was undoubtedly motivated by evolutionary necessity. Survival is granted to those who can distinguish the rustle of an attacker in a forest, for example, from a babbling brook nearby. As we design multimedia systems, the limitations of the auditory channel need to be respected.

Frequency Range of Human Hearing

As the frequency of periodic disturbances in the air increases, the human ear starts hearing sound when the disturbances are in the region of about 20 cycles per second, or 20 Hz. When we are first born, the upper range of audibility lies around 20,000 Hz, or 20 kHz, and usually declines with increasing age. It is important to distinguish between the frequency of a tone, which is a physical measure, and the percept of pitch, which the tone evokes. There is a close but not always exact relationship between frequency and pitch.

Dynamic Range of Human Hearing

The lower limit of the dynamic range of human hearing is at the threshold of audibility, and the upper limit is the threshold of pain (or damage). The audibility threshold for a I-kHz sinusoidal wave is generally set at 0.000283 dyne per square centimeter. (A sinusoidal waveform, a fundamental building block of audio waveforms, is shown in Figure 4.1)

As with pitch, it is important to distinguish between amplitude as a physical measure and loudness as a percept.

The amplitude of the sinusoidal waveform can be increased from the threshold of audibility by a factor of approximately 1,000,000 until the threshold of pain is reached. Working with such a large range of numbers is not convenient. To characterize the range from the threshold of hearing to the threshold of pain, it is more convenient to define the unit decibel, which results in a range of numbers that can be easily managed. Also, our general impression is that the loudness of a sound increases logarithmically with the power of the sound, rather than linearly.

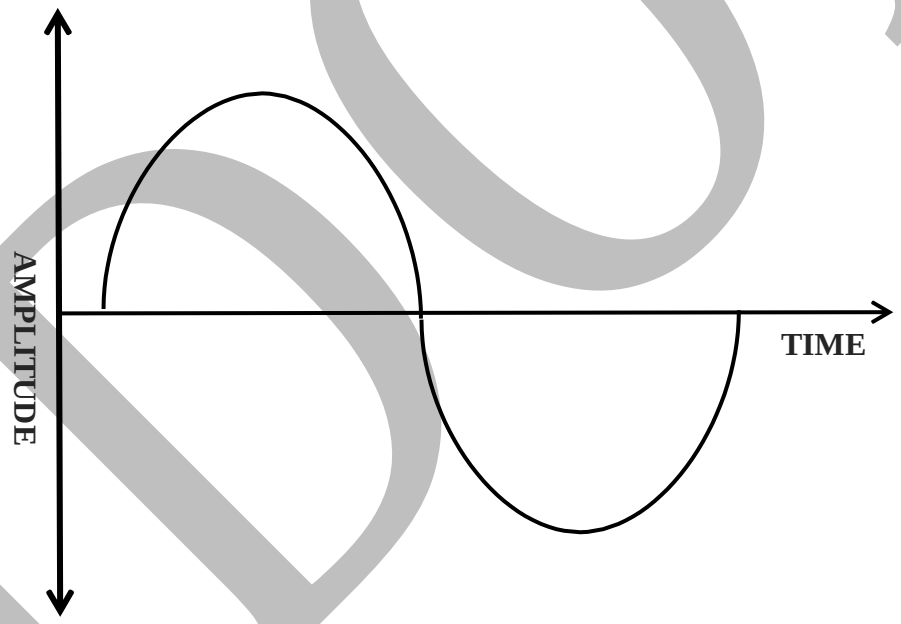


Figure 4.1 Sinusoidal waveform

For two waveforms with peak amplitudes A and B, the decibel measure of the difference in their amplitudes is given by

$$dB = 20 \log_{10}(A/B)$$

If the value of 0.000283 dyne per square centimeter (for a I-kHz signal) given above is used as a reference value for 0 dB, then the threshold of pain is reached at a sound pressure level (SPL) somewhere between 100 and 120 dB for most individuals.

There are complex interactions between the frequency and amplitude of a signal. For example, the perceived pitch of a tone can be modified by changing its amplitude. Consider further a plot of frequency as the x-axis and sound pressure level in dB as the y-axis, as shown in Figure 4.2. The threshold of audibility, the lowest curved line in the figure, is not a horizontal line. Instead, the threshold of audibility is higher at low and high frequencies than in the middle.

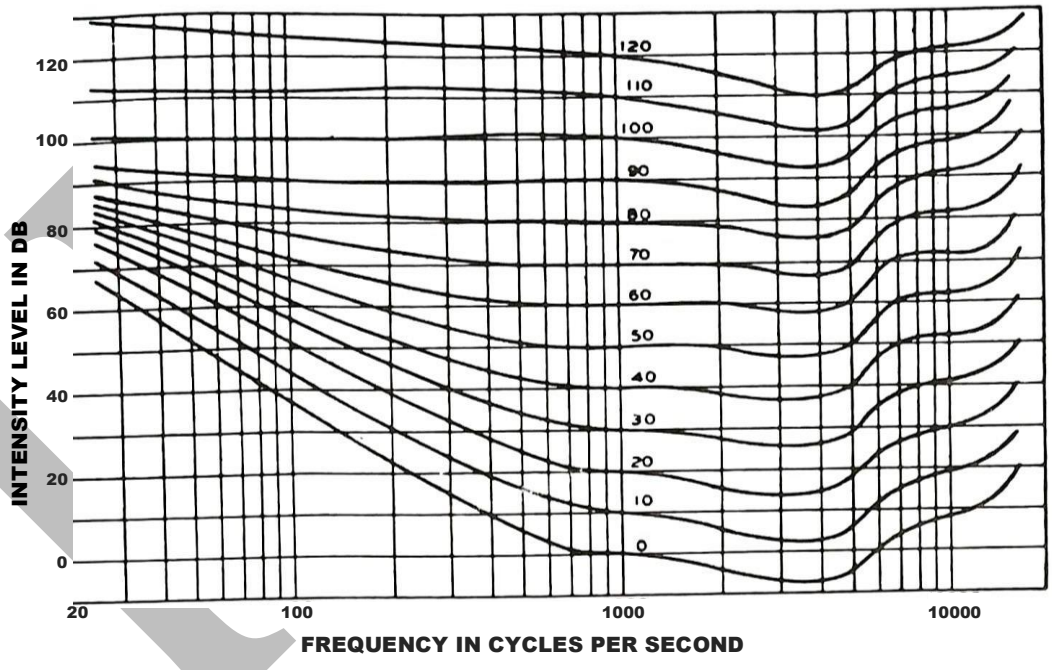


Fig 4.2 Perceived loudness. Each line shows a contour of equal loudness

Furthermore, the perceived loudness of a tone is not constant if the frequency of the tone changes. Put another way, a tone at some

frequency may need to have a larger or smaller amplitude than a tone at some different frequency, if both tones are to be perceived as having the same loudness. Figure 4.2 shows contours of equal loudness, known as Fletcher-Munson curves. On the thick curve that passes through the 1000 Hz/40 dB point, a tone at, say, 10,000 Hz would need an amplitude of more than 50 dB to be perceived at the same loudness.

Spectral Characteristics in Human Hearing

Figure 4.3 shows a section of the waveform of a musical note. Clearly this waveform does not match the sinusoidal wave shape of Figure 4.1. Using the mathematical techniques of Fourier analysis, one can break such a waveform down into its spectral components. The spectral components for a more or less constant wave shape are often graphed as in Figure 4.4.

For waveforms from wind, brass, and string instruments and from vowels in speech, the frequencies at which spectral components occur are given by more or less whole-number multiples of the lowest, or fundamental, frequency. In Figure 4.4, the vertical bars are thus more or less evenly spaced. The spectral components for percussion instruments often occur at non-integral multiples of the fundamental.

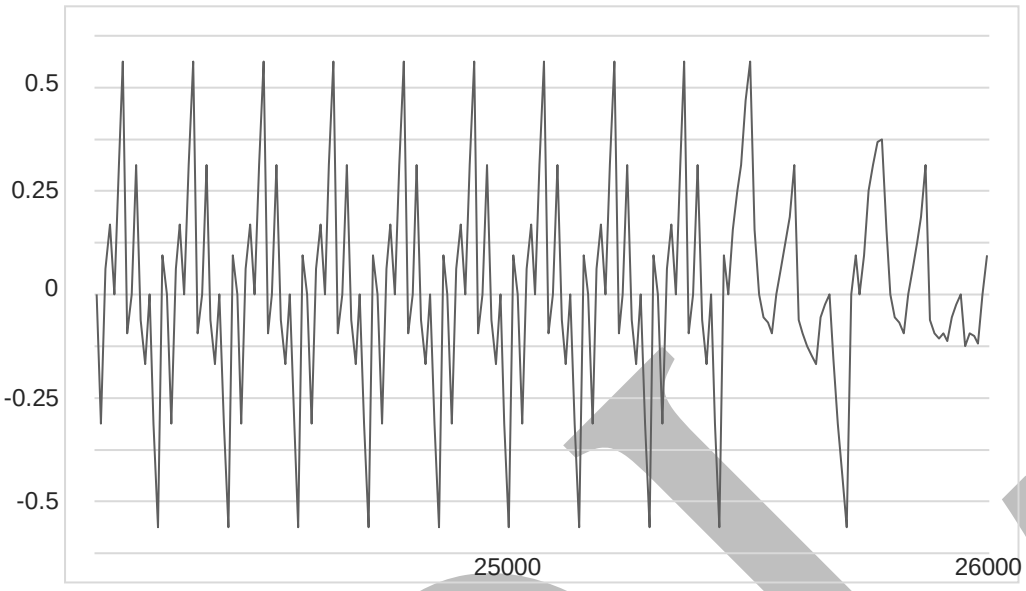


Fig 4.3 Time-varying waveform from the end of a clarinet note – x-axis: time; y-axis: amplitude

The ear is sensitive to peaks and valleys in the overall shape of the spectral components. A *formant* is a frequency region in which the amplitudes of spectral components are significantly raised or lowered; such regions can be seen in Figure 4.4. In human speech, the vowels are distinguished by having a few marked formant regions. In musical instruments, formant effects are often given by the shape of the resonating body (such as the tube of a woodwind instrument). As the fundamental frequency changes, fixed formants affect spectral components at different distances from the fundamental. The ear is quite sensitive to unnatural changes in the relationship between fundamental frequency and formant structure. Consider the chipmunk effect, when audio recordings are played more quickly than normal. Both the pitch and the formants are transposed, whereas the ear usually expects the pitch to be transposed independently of the formants.

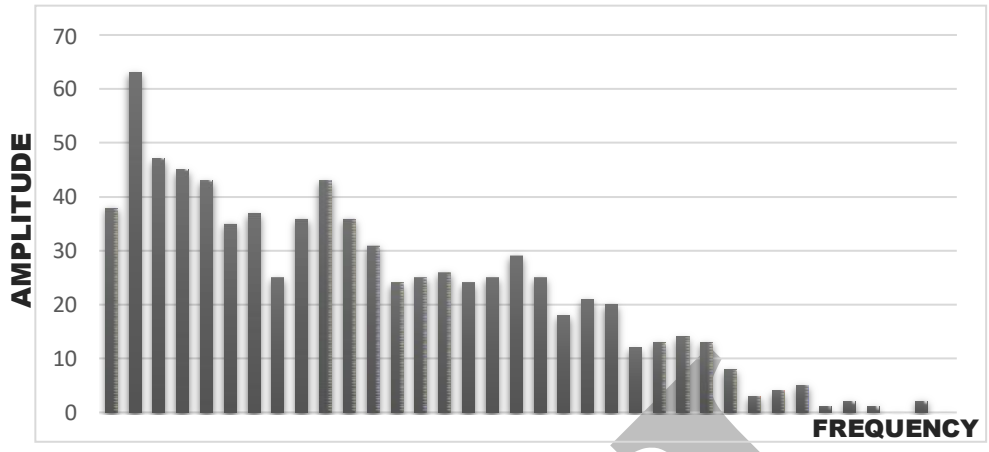


Fig 4.4 Spectrum of a portion of the steady state from a musical tone

The spectrum and other parts of an auditory event give rise to the percept labelled timbre. Timbre perception is complicated and not yet well understood. One way of treating timbre is to say that timbre allows us to identify the source of a sound. The timbre of a trumpet is said to be different from that of a trombone. Another aspect of timbre lies in the subjective qualities associated with a sound. One speaks of a piercing piccolo or a boomy bass drum. Unfortunately, in music, there is as yet no simple way to move from the physical waveform, or even the spectrum of a musical instrument tone, to a generalized framework for treating either source identification or the quality of a sound. (Some possibilities in speech are discussed below)

This has implications for multimedia systems. Suppose one wants to use an attention-grabbing sound from a sound database as part of a presentation. What makes a sound attention-grabbing? Or suppose one wants to search a sound track for relaxing sounds. In a written article, one can search for words and phrases; or if an author sprinkles labels in the text, it is easy to do a keyword search. There is no equivalent in

dealing with an audio signal. At best, textual descriptions can be recorded in parallel to recorded audio segments. In building a sound effects library, the sound editor in a film or video studio can supply subjective text labels. Fortunately, the labels can themselves be entered into a database to simplify retrieval of sounds. But new sounds entered into the system must still be classified subjectively, by hand.

Time-Varying Aspects of Natural Sound

In the past century, sound has come to be generated by non-traditional means, many of them electric and electronic. Traditionally, musical tones were divided into three regions in time, called attack, steady-state, and decay (Figure 4.5). In the steady-state region, it is supposed that the spectrum remains fixed. The simplest synthesis model calls for generating a waveform with fixed components, such as those of Figure 4.4, and applying an amplitude envelope, such as in Figure 4.5.

Almost everyone can tell the resulting sound from the richer sound of an orchestral instrument. Research has shown that there are significant changes in the spectral components of a musical tone in time. Figure 4.6 shows the time-varying amplitude spectrum of the attack of a musical tone. By studying such plots, we have learned that the spectral components vary in time with respect to each other, both in frequency and in amplitude. The frequencies are almost never in exact integer ratios, but instead vary in the region of 0.999 to 1.001

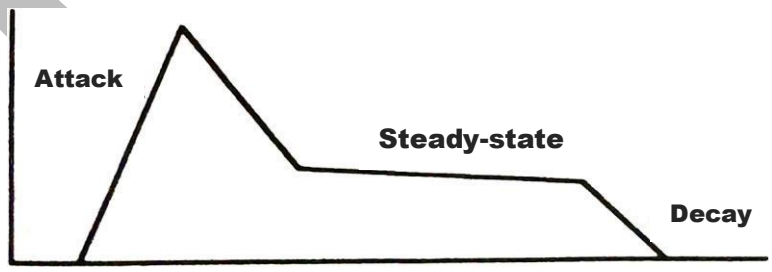


Figure 4.5 Simplified amplitude evolution of a musical note

Closely related to this is the realization that the attack portion of natural sounds is crucial for identification. In an experimental setting, subjects were asked to identify musical instruments after hearing short sections spliced from tones. Experiments show that it is far easier to identify a musical instrument from its attack than from its steady-state. Thus, the time-varying parts of the attack spectrum are often crucial for maintaining musical quality. (This again probably has an evolutionary motivation.)

It is important to mention in passing the time-resolution capabilities of the human ear. At one level, it is possible for the ear to detect changes at the level of a few milliseconds. For example, if one uses the data of Figure 4.6 to resynthesize a tone, a careful listener can discern the absence of the small "blips" in the attack. If these are left out, many listeners discern the difference and complain about the blatty quality of the brass tone as well.

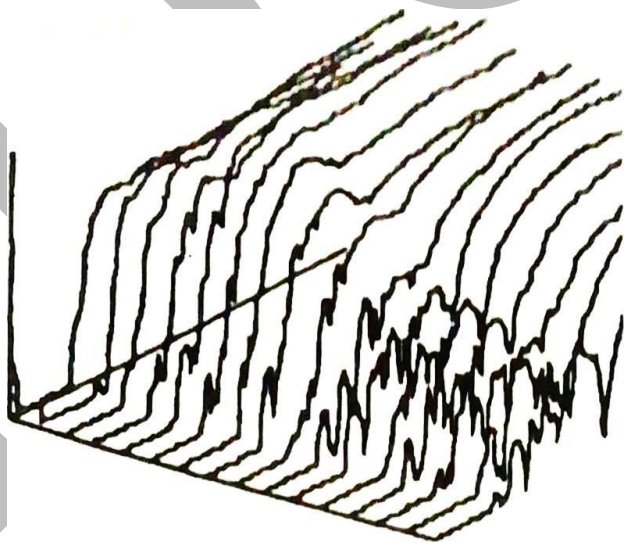


Figure 4.6 Time-varying amplitude of the first 16 harmonics of the attack of a trumpet tone. The fundamental is at the back. Amplitude is the vertical axis. Time runs from lower left to upper right

These phenomena have implications for the design of audio in digital systems. Mistakes can be disruptive. If your operating system has swapped out during a multimedia presentation when the synthesizer needs some new parameters, the attention of the listener .in the audience will suddenly be drawn instinctively to the resulting disturbance in the sound effects, even if the listener does not consciously notice a mistake in the sound. To avoid such mistakes in synthesizing high-quality sound, one might like to calculate the data rate for parameter updates to a synthesizer. However, since detail in the attack portion of sounds can be so important, one cannot get by with a calculation of throughput rate for parameter updates; one needs to take the burst rate into account.

Some parameters can be preloaded to accommodate the bursts. Or, if the system is just playing back pre-recorded sounds, they too can be preloaded for playback. There is another problem with that approach, however. Consider a multimedia conferencing system. In natural human communication, it must be possible for one speaker to interrupt another.

Likewise, in real-time music making, the performance needs to be modified as it happens because of adjustments made by the skilled performer. If too much audio data or parameters are queued, then one needs a mechanism to "jump to the head of the queue".

Masking

Masking may be easier to explain in the visual domain. We all know that if a bright light is shining in our eyes, such as headlights from an oncoming car, then other dimmer lights are impossible to see. There are similar phenomena in the auditory world. One sound can make it impossible to hear another, or one sound may shift the apparent

loudness of another. The masking of one sound may be total or partial. Also, parts of one sound can mask parts of another sound, even if we cannot consciously detect such masking in normal circumstances. Auditory masking effects can die away in a matter of milliseconds.

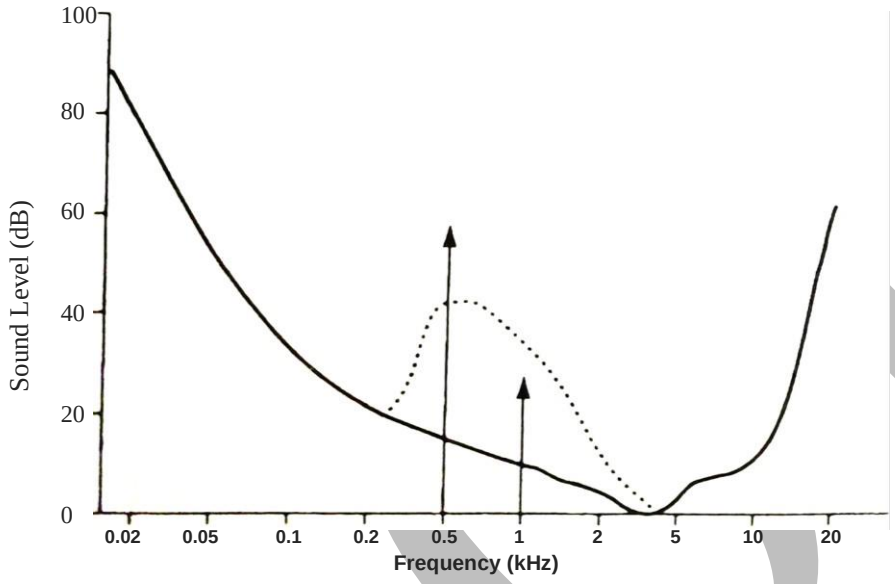


Figure 4.7: The threshold of audibility (solid line) shifted in the presence of a masker (left arrow).

If one tone is masking another, the effect depends on the separation in frequency. Figure 4.7 shows a typical plot of the masking effect (for example, for plots of maskers at different frequencies). The solid black line is the threshold of audibility. A spectral component on the left causes the threshold of audibility to be shifted upward, shown by the dotted line.

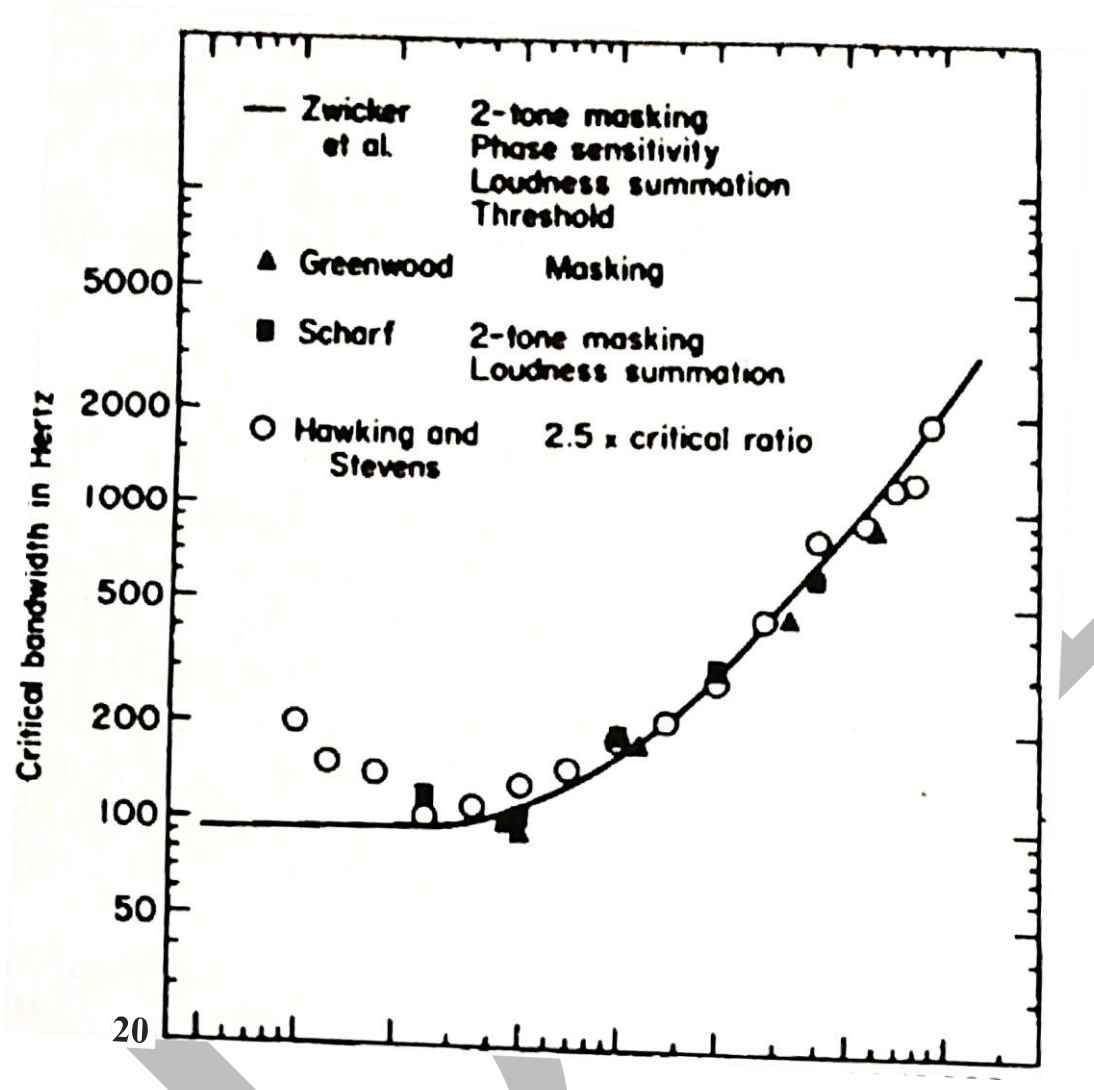


Figure 4.8 Critical bandwidth (y-axis) as a function of the frequency at the center of the critical band (x-axis). The solid line, circles, triangles, and squares are based on various measurements

A second spectral component, shown by the arrow on the right, is masked. Notice that the masking effect falls off more steeply to the right of the curve, that is, toward higher frequencies. Thus, a low tone can effectively mask more high frequencies, but a high tone affects relatively fewer lower frequencies. Masking can also happen if the tones do not happen simultaneously; that is, if a masking tone is short but occurs

before another tone, the masked tone can still be obscured. Masking effects happen whether the masker is a single tone or a broader band of noise.

Laboratory studies of such masking effects led to the notion of critical band; that is, a given frequency is surrounded by a band of frequencies within which various auditory phenomena can be shown to occur. For example, consider the loudness percept evoked when two tones are at the same frequency. Now move the tones apart. As the frequency between tones increases, experimental evidence shows that the perceived loudness does not increase until the distance between the tones exceeds a critical band.

A critical band is characterized by a center frequency and by bandwidth. The bandwidth is approximately one-third of an octave above perhaps, 300 Hz. Below that, the bandwidth is more or less fixed. The theory does not state that the ear has fixed bands; rather, the theory says that the band within which certain phenomena occur increases in size (frequency) as frequency increases. This property can be closely matched to physiological properties of the human ear. The implications of masking and critical band phenomena for communications technology are discussed below, especially in the section on MPEG.

Phase

One of the characteristics of a waveform not yet discussed here is its phase. Two waves with the same waveform are said to be in phase if they start at the same point and move in the same direction. For sinusoidal waveforms, such as in Figure 4.1, the waveform is said to be

180 degrees out of phase when the illustration is flipped top-to-bottom. There is some controversial evidence that humans can hear absolute phase. More importantly, phase problems occur in stereo transmission. If two signals are exactly out of phase, they cancel, resulting in the sound of silence. For normal listening, this rarely occurs. But consider the case when a stereo soundtrack from a movie is mixed to mono for transmission to conventional, monophonic televisions. It can and does happen that some sound engineer somewhere mixed up the phase relations on some sound effect or two.

There are true stories of the network executive heating up the telephone after the machine gun on his television suddenly spoke silently. Phase continues to be a practical problem in broadcast and will undoubtedly creep into, for example, multichannel conferencing systems.

Binaural Hearing and Localization

In most normal listening situations, we hear sounds coming from all directions around us, including above and below. The ear uses various factors to determine sound location. These include intensity (if the sound is louder on the left, it probably came from the left); timing (if the sound hits the right ear first, it probably came from the right); and spectrum (the head imposes filtering effects as sound wraps around from one ear to the other). It is now known that the outer ear imposes certain filter characteristics as well, depending on the direction and elevation of the sound.

The distance of a sound to the listener is affected by the reverberant field. In any enclosed room, the sound from the original source bounces off the walls, ceiling, and floor. Within a few tens of milliseconds after the

sound starts, there can be literally thousands of these mini-echoes, especially in a good concert hall. It is now well understood how to make artificial reverberation.

The effects of sound in space are subtle but important. In a teleconferencing system (or on a movie screen), if the speaker's head is shown in one position and the sound of the speaker's voice comes from another position, then the listener can be confused. For a discussion of the importance of this when the speaker is moving in a teleconferencing system.

DIGITAL REPRESENTATIONS OF SOUND

Time-Domain Sampled Representation

Sound in the analog world is said to be continuous in both time and amplitude. The sound's analog amplitude can be measured to an arbitrary degree of accuracy, and measurements can be taken at any point in time. A digital signal is different; the signal is defined only at certain points in time, and the signal may take on only a finite number of values.

To sample a signal means to examine it at some point in time. Sampling usually happens at equally separated intervals, at a rate called the sample frequency, determined by the Sampling Theorem. If a signal contains frequency components up to some frequency f, then the sample frequency must be at least 21 in order to reconstruct the signal properly. In practical systems, the sample frequency must be higher than 21. In the early days of digital audio, sample frequencies at 44.1 kHz and 48 kHz were adopted to handle the full 20-kHz range of human hearing. The sample frequency of 32 kHz is also common. In multimedia systems for

the PC market, sub-multiples of 44.1 (22.05 kHz, 11.025 kHz) are often found. The highest frequency that can be handled (i.e., one-half the sample rate) is often called the Nyquist frequency.

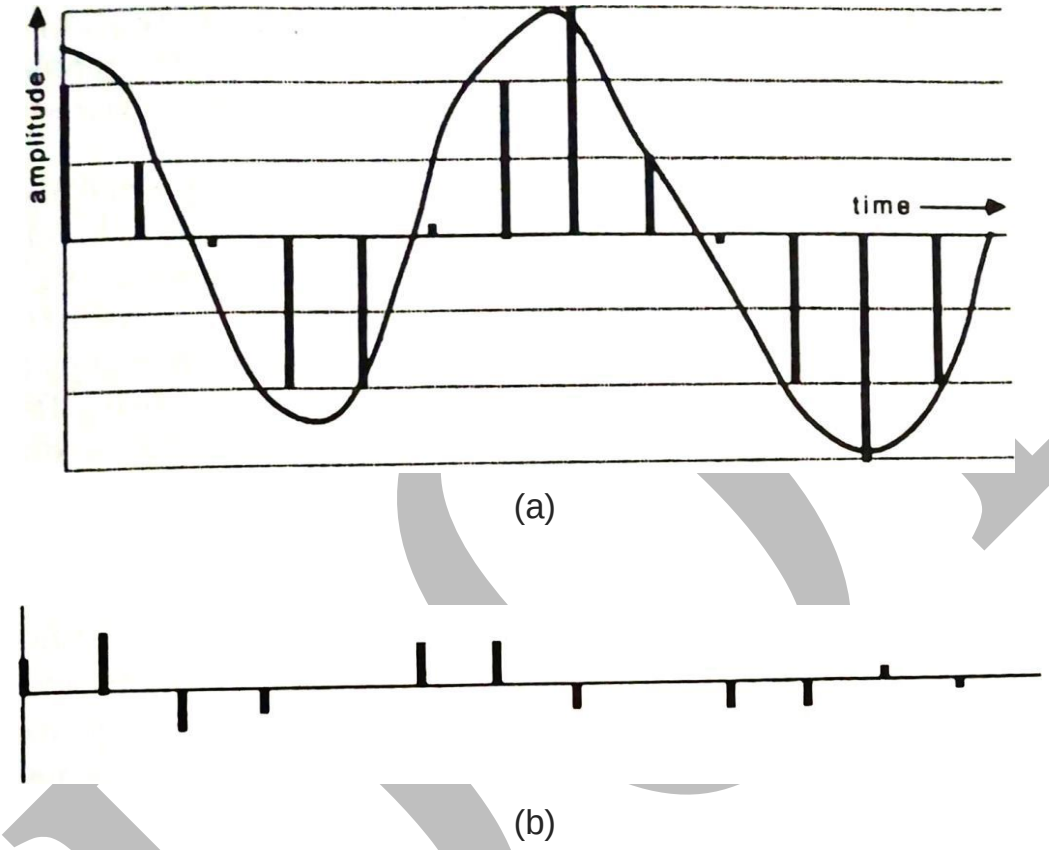


Figure 4.9 (a) A sampled and quantized signal (vertical bars) compared with an analog original signal (solid line), (b) The quantization error

To quantize a signal means to determine the signal's value to some arbitrary degree of accuracy. Figure 4.9 (a) shows an analog waveform (solid line) and the representation of that waveform as quantized samples (vertical bars). The digital signal is defined only at the times where the vertical bars occur. The height of each vertical bar can take on only certain values, shown by dashed lines, which are sometimes

higher and sometimes lower than the original signal. If the height of each bar is translated into a digital number, then the signal is said to be represented by pulse-code modulation, or PCM. In the world of digital music, a sampled and quantized signal shown by the vertical bars in Figure 4.9 (a) is called a sampled signal to distinguish it from the signals synthesized.

The difference between a quantized representation and an original analog signal is called the quantization noise. Figure 4.9 (b) shows the quantization noise for the signal in Figure 4.9 (a). Quantization noise differs from analog tape hiss. Quantization noise can be present only when there is some audio signal to be quantized, whereas tape hiss is always present. Basically quantization noise follows the signal like a halo. With more bits for quantization of a PCM signal, the signal sounds cleaner. Each additional bit of accuracy improves the signal-to-noise ratio by about 6dB. This is because each bit represents a factor of 2, and 20 log 2 approximately equals 6dB. For 8-bit systems, the quantization noise is quite audible. Early digital audio systems operated on 16 bits, given in part by the 16-bit word length common in computers and digital hardware. Audio CDs store 16 bits of data. As discussed above, the human ear can hear a wider range (perhaps 120 dB) than the 16 bits x 6 dB/bit = 96 dB maximum range provided by a 16-bit system. One obvious solution is to use more bits, such as 20.

There are hardware devices called digital-to-analog converters (DACs) and analog-to-digital converters (ADCs) for moving between the analog and digital domains. The hardware in and around an ADC follows the input analog signal; takes a snapshot of its value at the sample time; holds onto that value until conversion is finished; and outputs a number. The opposite happens in analog-to-digital conversion. The DAC accepts

a number; hangs onto that value until the value is converted to some electrical signal (such as current or voltage); and sends that signal to the outside world. If the converted signal is fed to a loudspeaker, for example, then the loudspeaker cone is made to move in a way that causes us to believe that we are hearing a recorded signal.

There is a filter, called the anti-alias filter, involved with the ADC. As required by the sampling theorem, this filter removes components above the Nyquist frequency. There is a corresponding filter, often called a reconstruction filter, associated with the DAC.

The filtering associated with DACs and ADCs can itself introduce un wanted distortions into the signal. Although these are often subtle, for high-quality work they should be minimized. One solution is to use so-called oversampling DACs. The DAC operates at four, eight, or some other multiple of the sample frequency. The requirements on the associated filter are not so stringent in the audio band. In effect, the associated with the filtering are pushed into higher-frequency ranges, they become inaudible.

Since we are working with the transition from the analog to the world, it is important to mention in passing that the digital world is unfriendly to analog signals. For example, the radio frequency bouncing around inside a PC leak easily into analog lines coming in the outside. Also, the usual variations in manufacturing, temperature, and line voltage can lead to nonlinearities in DACs and so that a 16-bit system is not really a 16-bit system after all. Another problem worth mentioning has to do with interrupting the digital stream on its way to the outside world.

The loudspeaker cone protests vigorously if several samples are inadvertently skipped. It can and does happen, even on the floors of professional audio trade shows, that someone has made an implementation mistake. Suddenly the loudspeaker cone is trying to move from its extreme inward position to its extreme outward position. The result is a big bang.

Those debugging digital audio in the early stages of algorithm development are advised to keep the volume on their audio gear turned down.

Transform Representations

A sampled and quantized signal such as shown in Figure 4.9 (a) can be transformed into another representation. Such transforms have been extensively studied. A primary motivation comes from communications Channels with narrow bandwidths.

Even in the early days of analog communications, researchers explored whether a transformed signal might be easier or more robust to transmit than the original signal. The signal would be transformed at the sender, the transform parameters would be transmitted at a lower bandwidth, and the receiver would reconstruct the signal.

Fourier Methods

One of the most common methods is to use a digital form of the Fourier transform. The PCM signal is referred to as the time-domain signal, and the Fourier coefficient representation is referred to as the frequency-domain signal. For a stationary time-domain signal lasting an infinite amount of time, a single set of Fourier coefficients is adequate to represent and reconstruct the signal. For real-world musical and speech signals that vary in time, as we have seen, we are technically dealing with the discrete short-time Fourier transform, popularly known as the phase vocoder. In the world of digital music, this corresponds to additive synthesis.

The phase vocoder and its close relatives have proven to be an invaluable tool for research into the nature of speech and sound. One simple example was given in Figure 4.6. Nonetheless, there are problems with the phase vocoder.

Depending on the parameters, there can be an explosion of data by a factor of 10 to 100 or more. Only recently, with the advent of systems such as the Silicon Graphics Indigo or the NeXT, has it become fairly easy to deal with these amounts of data.

There are other transform domains to be mentioned in passing. One of the closest to the digital world is the Walsh transform which has not yet, to my knowledge, been exploited in any commercial musical system. The problem with the Walsh representation is that there is no intuitive relationship between changing one of the Walsh coefficients and the auditory results of the change.

Subband Coding and MPEG Audio

Another clever method for compacting the required data stream is to exploit the masking properties of the ear discussed earlier. One way to do this involves subband coding, in which the signal is broken into bands which can be transmitted as a group at lower data rates than required to r the original signal There are many subband coding schemes for audio, such as the ATRAC scheme used in Sony's Mini Disc format or the Digital Compact Cassette (DCC) introduced by Philips. Subband coding has studied extensively in the speech community.

A standard known popularly as MPEG (Motion Picture Experts Group) Audio is given in the document ISO/IEC DIS 11172 promulgated jointly by the international standards organizations ISO and IEC The MPEG group is officially known as ISO-IEC/JTC1/SC29/WG11e The video encoder and decoder, which form part of the theoretical basis of the video portion, are given in full in the ISO/IEC. For more information on MPEG audio we now discuss MPEG in some detail, since MPEG provides a good example of subband coding theory and practice.

A full MPEG system includes an encoder and a decoder. As with MPEG video, the audio encoder is not strictly defined in the document, but functionality of the decoder is tightly specified. Some sample audio encoders are given in the document; they are briefly reviewed here. Some companies are working on chip implementations of MPEG coder/decoders (CODECs), such as the C-Cube CL450. Proprietary implementations of MPEG audio encoders are being patented and will be brought to the marketplace as well.

The MPEG audio standard is closely bound to the arena of mono and stereo audio to accompany images, as opposed to professional and consumer audio electronics. (A standard tied to professional audio would have envisioned an escape mechanism for more than two channels, for example.) There are four modes:

- single channel
- dual channel (two independent channels, for example in two languages, coded in one bit stream)
- stereo (two stereo channels coded in one bit stream)
- joint stereo (stereo pair coded exploiting redundancy between right and left channels)

There are three possible *layers*, or *degrees*, of processing. Regardless of which layer of coding is used, an audio frame consists of a header, optional bits for cyclic redundancy (CRC) error check, the audio data itself, and (in layers 1 and 2) optional ancillary data. The header contains a synchronization word, identification of the layer (1, 2, or 3), sample frequency (32, 44.1, or 48 kHz), a padding bit (to help with 44.1 kHz sample frequency), a bit for private use, some bits for emphasis, and copy protection bits.

The encoder for layer 1 (Figure 4.10) maps the digital audio input into 32 subbands; that is, the available frequency range is divided into 32 bands. The conversion to subbands is done for a frame of 384 audio samples. During each frame, the subbands are sampled 12 times. But not every subband is actually encoded.

Instead, a psychoacoustic model (three top right-hand boxes in the figure) determines allocation of the available bits to the perceptually strongest bands. A new allocation of bits among the subbands is

determined for every frame. Actually, two psychoacoustic models are available in the specification.

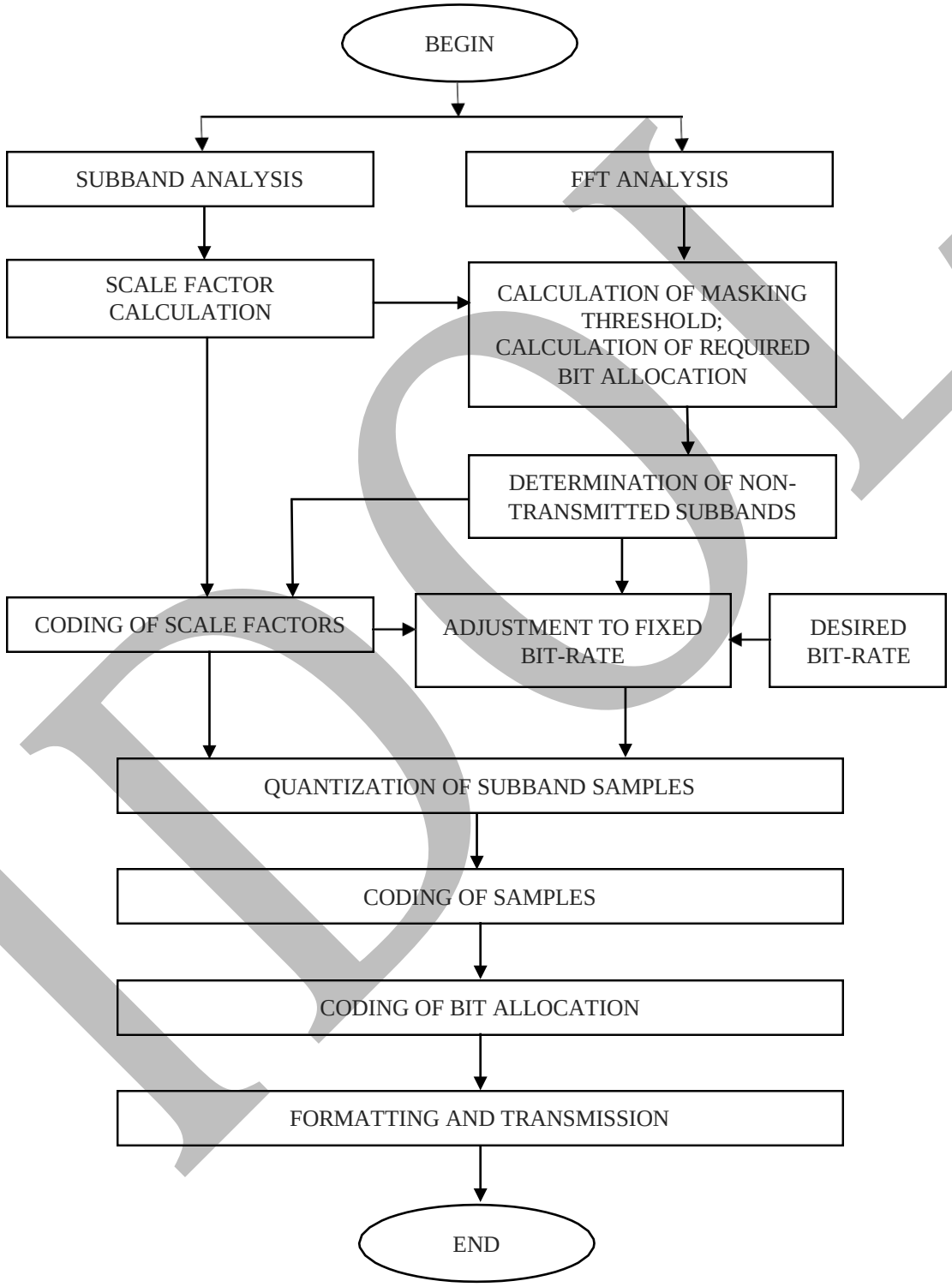


Figure 4.10: The structure of the encoder for MPEG layers 1 and 2

Only the first is described here. The first model separates the bands output by fast Fourier transform (FFT) analysis into those that are more sinusoidal (called tonal in the specification) and those that are more noise-like. The model determines which bands are masked by others, based on critical bands. The model removes bands that are irrelevant, that is, bands which fall below the absolute threshold, which is similar to the lower line in Figure 4.2. Then the model calculates the masking threshold for each band, based on the surviving maskers. In each band, the model further compares the signal in the band with the calculated masker. Using the resulting signal-to-mask ratio, the model determines the mask-to-noise ratio (MNR) in each band. An iterative procedure uses the MNR to assign coding bits to the bands with the lowest MNR. As long as bits are available, more subbands are encoded.

This means that some subbands remain uncoded, in which case a zero is transmitted as the bit allocation. A scale factor is also transmitted for each band when the number of bits is not zero, since the bands are scaled before quantization. Finally, for nonzero subbands, 2 to 15 data bits per subband are transmitted. One advantage of this scheme, by the way, is that noise in one band is independent of noise in another band.

User-defined ancillary data can also be transmitted. Since the ancillary data consume some bits that would otherwise be available for audio itself, using ancillary data could possibly degrade the audio quality.

Decoding, shown in Figure 4.11 ahead, is the opposite of encoding. The decoder determines from the bit stream which subbands have nonzero data. The data from those bands are read, and the bands are rescaled

according to scale factors. A re-synthesis filter creates an output PCM stream.

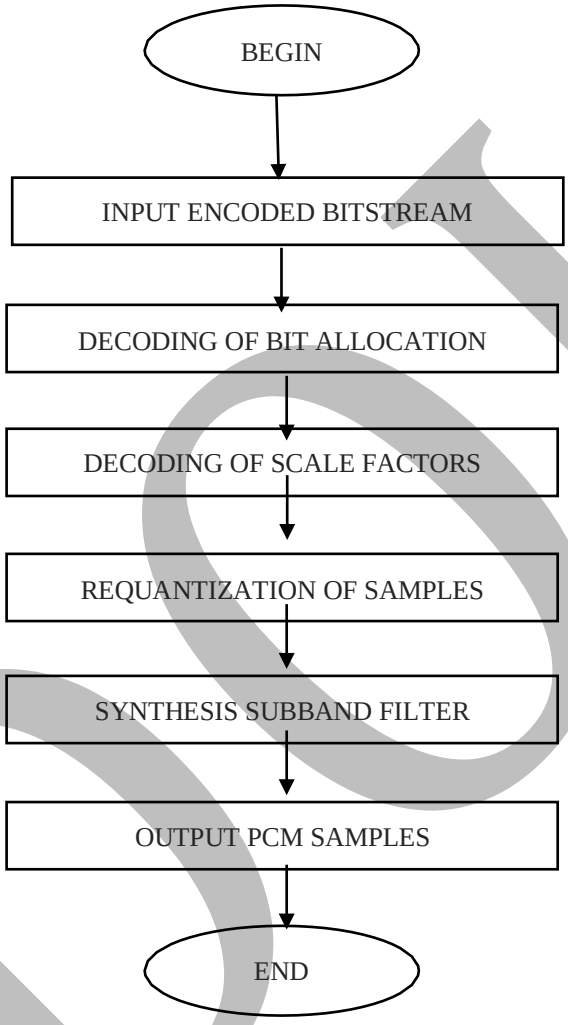


Figure 4.11: The structure of the MPEG decoder for layers 1 and 2

Layer 2 features additional coding of bit allocation, scale factors, and samples. There is a different frame rate of 1 152 input samples in one frame. As many as three scale factors are transmitted for each frame, for 36 Subband samples per frame. If the differences between successive scale factors are very small, only one scale factor needs to be transmitted, freeing up bits for encoding the subbands.

In level 3, there are 1152 input samples per encoded frame, and-the encoder is much more complicated. One difference is that a "hybrid" filter bank is used to increase frequency resolution. More precisely, the available frequency range is divided into two parts. The higher-frequency components above a certain point can be encoded with better time resolution. The lower-frequency components are encoded with better frequency resolution.

Level 3 also has a non-uniform quantizer and adaptive segmentation of data; that is, if there is a peak demand for bits in a given frame, the number of bits available for coding can be temporarily increased. The frame header mentioned above occurs at regular intervals, but the data for a frame of input samples can occur before and after the header corresponding to that frame.

Subtractive-based Representations

Another popular approach is to model the signal as a spectrally rich source followed by one or more filters. The most popular implementation of this model in the speech community is called linear prediction, discussed below. The linear predictor has found wide application in the computer music community. The advantage of the linear predictor for music composition and performance is that the fundamental frequency can be separated from the spectral characteristics. Still, the linear predictor is not a high-quality system in terms of musical quality. One reason is that the all-pole model used in traditional linear prediction cannot model closely the spectral shape of musical instrument tones.

Parametric Representations

Another class of representations has nothing to do with models of hearing or acoustics in particular, but exploits some numerical properties to achieve high-quality audio results. The first major breakthrough in this arena was frequency modulation, developed at Stanford by John Chowning. FM had been used for several decades for radio transmission. Chowning discovered that certain combinations of carrier and modulator frequencies could produce waveforms that mimicked the properties of musical waveforms. FM was used in digital synthesizers and other musical devices, mostly by Yamaha. Many PC plug-in cards also use FM chips for sound synthesis.

Other techniques such as granular synthesis, waveshaping, Chant, or the Karplus-Strong algorithm have been investigated in the computer music community and implemented in some synthesizers such as the Korg 01/2 Series (waveshaping). Techniques in the speech community such as vector quantization are discussed below.

TRANSMISSION OF DIGITAL SOUND

When one is sending signals from one piece of digital equipment to another, many problems can be encountered. There is always the fallback position of connecting the analog output of one digital machine to the analog input of another digital machine. A copy made in this manner will still sound very clean. But often in real-world work, several generations of copies must be made. If the signal is converted to the analog domain for every copy, unacceptable amounts of noise are gradually introduced.

One problem in connecting digital equipment is that there are several standard combinations of sample rate, number of channels, bit and byte order, and number of bits per channel used in digital audio recording. Some are intended for professional recording studios, other strictly for consumers, and some are called "semiprofessional." For example, the compact disc nominally stores stereo 16-bit signals at a 44. I-kHz sample rate; this consumer format is acceptable for semiprofessional use. Another format is the Digital Compact Cassette (DCC), which uses a different form of encoding, with sample rates of 32, 44.1, and 48 kHz. Other kinds of problems occur in recording studios, for example, where equipment with noisy fans must be isolated from the recording studio itself. A digital signal can degrade when transmitted over some kinds of long cables.

To solve the problem of differing sample rates, one uses a sample-rate converter. This process can also introduce subtle amounts of noise if the implementation is not handled properly. Chips have been introduced recently to simplify implementation.

Several formats have been standardized for connecting digital audio equipment. One basic format is known as the AES/EBU format (Audio Engineering Society/European Broadcast Union). The AES/EBU format transmits at least 16 bits of PCM audio, and can easily transmit up to 20 bits. Two channels are always transmitted at once.

Clock information is encoded with the data, so no extra connectors need to be provided for a clock signal, as is the case with some formats. Several "status" bits are transmitted as well. These can include the sample rate as well as user-programmed information. For work in large

recording studios, the AES/EBU format has been generalized to the multichannel audio digital interface (MADI) format 1501, which handles 56 channels of audio and up to 24 bits of audio data per sample.

Some companies have introduced their own formats, such as Sony-Philips digital interface format (SPDIF), Sony digital interface overview format (SDIF), is in Yamaha, and Mitsubishi Electric Company (Melco).

DIGITAL AUDIO SIGNAL PROCESSING

Digital Signal Processing (DSP) treats a signal as a series of numbers. Regardless of the content implied by the numbers, the numbers (the signal) can be treated (processed) in many ways. For example, if every number is replaced by the average of itself and its neighbour, the graph of the sequence of numbers is smoothed out. In the audio domain, this creates a low-pass filter.

Traditional digital audio signal processing is based on linear systems theory; that is, if you insert a signal with certain spectral components into the input of the system, the output has spectral components at the same frequencies, possibly delayed in time—no new frequencies are introduced. The limitations of linear theory become evident when one strives to characterize, for example, the behaviour of a clarinet reed. For many years there have been simple but practical uses for nonlinear systems, such as in the "fuzz box" (nonlinear circuit to enrich sound) popular with guitar players. With recent advances, nonlinear theory and practice are becoming more and more important.

Another characteristic of traditional digital signal processing is that it deals with causal systems; that is, there can be no output that happens before an input. Put another way, causal digital signal processing cannot "look ahead into the future." However, this causal nature is often thwarted in real-world applications. For example, in non-real-time systems for cleaning up tape hiss or background noise, the software "looks ahead" to see if it can characterize the background sound more precisely. Another example is in the side chain of audio compressors, discussed below.

In hardware and software, DSP is used to implement the usual functions needed to modify sound for recording and playback. Simple DSP algorithms can be chained together to create filters (high-pass, low-pass, band-pass, band-reject). Everyone knows some such controls from home entertainment equipment. Professional recording engineers' use these filters for more subtle effects, such as making the singer "stand out" from the background by changing the equalization on the singer's voice. Such filters can be grouped into equalizer banks. More sophisticated uses of DSP include time-warping signals, such as in the WordFit system. Briefly, in re-recording spoken passes for redubbing telephone and movie sound, it is necessary to make the re-recorded sound match in time the original spoken dialogue. WordFit finds the words in two sets of recordings based on the same text and then maps the timing of one recording to fit the timing of another.

For modifying level, gain control in DSP is a simple multiplication. More complicated level modification is given by a compressor, which reduces the dynamic range of a signal. In such a unit, the output level is equal to the input level until the input level reaches a certain threshold. Above that threshold, the output level increases by smaller amounts as the input level increases. If the signal is first compressed and then the whole signal

is amplified, the result is a signal which sounds louder overall. This can be useful in automobiles, for example, where the quiet parts of symphonic music get lost in traffic noise. A variant on the compressor is the limiter, which tries to limit the signal's output level to some more or less hard level. In devices such as compressors and limiters, the input signal can be delayed before being processed. This allows the unit to "look ahead," circumventing causality. For example, if a loud drum strike is coming soon, it may make musical sense to start lowering the level of the output signal now rather than later. If the amplitude is reduced only when the drum hits, the listener may hear an unnatural result.

If a signal is delayed and added back into itself, an echo can result. If many such echoes are generated at the right time and scaled at the right amplitudes, the effect of reverberation can be generated. Digital reverberators were the first commercially available digital audio units on the market.

There have been significant advances in DSP in general with tracking and identification of arbitrary signals. Submarine warfare is one hot topic. One wants to know from the acoustic signature of a submarine whether it is friend or foe. The concept of friend or foe is rarely encountered in musical structure, but tracking musical performances would be a useful aid to computer-based musicians. Unfortunately, a seemingly simple problem such as pitch tracking turns out to be difficult. For example, if the musician connects two notes with a glissando, where does one note end and another note begin? Even when the pitches of two successive notes are distinct, it is hard to know for certain when a new pitch has been reached. (Endpoint detection is more advanced for word recognition in speech than in music.) Musicians performing with pitch trackers practice hard and learn to avoid the pitfalls. As for identification of instruments,

even to identify accurately a solo instrument recorded separately is considered a difficult task today. A more complex problem, such as separating musical instruments from the total symphonic signal, is barely in its infancy. The DSP solutions to speaker identification in speech are mentioned below.

To prepare the sound track for a movie, video, or presentation, it is necessary to assemble sounds in sequence. In decades past this kind of work was done in professional and semiprofessional studios with audio tape. Recently, hard disk systems have become common in the professional and semiprofessional audio industry. For PCs, inexpensive boards are available for individual users that have the same capabilities. This brings the world of "desktop audio" one step closer to the kinds of gains that desktop publishing has achieved in the past 20 years. In hard systems, sampled sound is stored as files on a hard disk (as opposed to tape, the previous working medium in the audio industry). The sounds can be spliced together in a given sequence under software control. Often begin and end of a sound must be trimmed to remove, for example, unneeded silence. To chain two sounds, it is usually necessary to perform a crossfade across some number of milliseconds. Again, if two sounds are simply abutted, then it can easily happen that the digital sample values at the join have an unexpectedly large difference in amplitude, causing a very unpleasant pop.

As an aside, it deserves to be mentioned that the sophisticated notions developed for client/server relations in graphics have barely been extended to audio. Some audio and DSP resources are often expensive. Also, audio hardware resources are rarely needed for, say, hours at a time. It makes sense in many settings to have the audio resources centrally located. One implementation involves a large *N x M*

switch, with N the number of inputs and M the outputs. Under computer control, the switch sends outputs to individual workstations as needed. A mixer at the user's station, possibly computer-controlled as well, can set final output levels.

Returning, then, to the world of DSP, it is possible for add-in boards for even the Macintosh or the PC to perform significant signal processing with current digital signal processing chips, such as the Motorola DSP 56001 family or the Texas Instruments TMS 320 family. The NoNoise System from Sonic Solutions, for example, which runs on a Macintosh, can perform click removal and background noise cleanup.

Stereophonic and Quadrophonic Signal Processing Techniques

The first known stereo transmission of sound occurred more than 100 years ago, when the Paris opera could be heard in stereo over headphones at a remote site. The fundamental principle of most modern stereo recording and reproduction systems is to record the natural sound field with two microphones and reproduce those signals from two loudspeakers. For signals recorded with just one microphone, or for synthesized signals, it is possible to "pan" the signal from left to right simply by changing the relative amplitudes of the signals going to the left and right speakers. To make the sound appear more distant, reverberation can be added, and the ratio of direct to reverberant sound is an important control. For sound in motion, Doppler shift is also needed. To synthesize elevation, the filter response of the outer ears of test subjects can be averaged so that the frequency response corresponding to a given elevation can be added. All of these algorithms can be

implemented on commercially available processor cards. There are also commercially available (sometimes proprietary) systems such as the Roland Sound Space (RSS) that add location informati0ß to a signal. For a quick review of recent 3-D developments in the commercial audio industry.

Adding location information is more than just a game. For example, fighter pilots often listen to three or four continuous streams at once. If the perceived location of each stream can be separated in 3-D perceptual space, then it is easier for the pilot to separate which stream to pay attention to. Similarly, in audio augmentation of a computer graphic system, suppose that there is a sound associated with each window. As discussed by Ludwig et al., it makes sense for the apparent location of each sound to match the position of its associated window on the screen. When a new window pops up, say with a video image of another participant in teleconferencing, the sound should seem to come from the position of the speaker's head. More importantly, sounds can be processed so that they are treated hierarchically. If the window disappears, the sound can be turned off. If the user selects the window, the sound can be made to jump "into the foreground."

Architecture of an audio Signal Processing Library

Various researchers in the digital signal processing, speech, and computer music communities have dealt with the problem of sound representation during the past 20 years or so. Some of the efforts have fortunately been standardized and released for general use.

In dealing with audio signals, there is first the problem of storage. (We ignore tape solutions here, such as DAT players.) It may seem odd to

discuss file formats here, but disk access speed and DSP chip bandwidth are still such that one quickly comes up against real-world constraints. Typically, for a signal stored in PCM format, one wants random access to permit editing, and one wants real-time playback.

Until recently, playback from disk was a difficult problem that required careful planning and possibly even specially formatted disks. For a signal stored in the transform domain, one wants easy access for editing, but high throughput for possible real-time resynthesis. For time-varying Fourier analysis, editing is easier when each channel's Fourier representation is stored as one unit on the disk. For real-time resynthesis, such a format results in a high number of disk seeks, so a format in which all channels in each time slice are stored together is more convenient. One would also like to regularize the storage format so that ideally one editing program can read and edit sampled data, LPC, time-varying Fourier coefficients, and the like.

The ESPS system by Entropic Systems laboratories is typical. The file header, stored on disk with the data, specifies the format for storage. In practice, one format for sound files and another for data files typically becomes standard practice.

We now turn to components of signal processing libraries. The components obviously depend on the application. Perhaps the first main software signal processing library was put together by IEEE. The IEEE FORTRAN library contains implementations of Fourier transforms, linear prediction, and filter design, among others. The Numerical Recipes volumes were another big step, containing implementations of interpolation, linear equations, FITS, random number generators, and the like. For the Motorola DSP 56001 chip included in the NeXT

computer, there is a library bf hand-coded routines, most of which operate on vectors. Some examples of the contents of the library are given in Table 4.2.

Function	Description
cvconjugate	form a complex from the elementwise conjugate of a complex vector
cvtcv	pointwise multiplication of two complex vectors
fftr2a	radix 2 decimation-in-time FFV, complex input and output, in-place, output shuffled
mtm	multiply two two-dimensional matrices, creating an output matrix
sumvsq	sum squares of vector elements to a scalar vector move, bit reversed
vmovebr	Vector move, bit reversed
vramp	fill a vector with a ramp function
vtvmvtv	pointwise multiplication of two vectors, subtract pointwise multiply of two vectors

Table 4.2 Example Functions from the NeXT Signal Processing Library

At another level, the VCOS operating system from AT&T includes a Multimedia Module Library (MML) containing these components:

- Generate and decode TouchTone@ dual-tone multifrequency tones
- CCITT G.722 standard audio compression and decompression
- MPEG audio encoder and decoder subband encoder and decoder, compressing 8-kHz sampled speech to 16 or 24 kb/sec

- sample rate conversion at ratios of 1:3 and 3:1
- JPEG decoder

In other libraries one finds modules for music synthesis algorithms, standard filter structures, reverberators, time delays, implementations of vocoders, and the like.

Editing Sampled Sound

Hardware and software systems for editing digital audio have existed for several decades. Some started in the commercial world, especially with the Fairlight synthesizer. Others, often discussed in the pages of Computer Music Journal and the Proceedings of the International Computer Music conference' were developed in academia. One typical package for Sun workstations is in the ESPS system, available from Entropic and mentioned in the previous section.

Certain basic capabilities can be found in any sound editing system. One needs to see the waveform, sometimes up close (short time resolution), sometimes far away (overview).

Figure 4.12 shows a typical editing window. Time runs from left to right. The top layer of icons allows for operations such as playback, zoom, and cursor movement. Other options are available from menu items. The gray horizontal graph shows an overview of the recording, which lasts approximately two minutes. Below that are two large windows, each showing one channel of the stereo file. The highlighted areas in the stereo detail match one of the small outlined rectangles in the overview.

Figure 4.12: Typical sound display, taken from DigiDesign's Sound Designer II

The user wants to see the values of the sound file, listen to parts of a file, and perform operations (fade-in, fade-out, FFT, and the like). In the music world, composers and performers need to mark sections of a sound file for synchronization with video, using SMPTE code, for example. Cut, duplicate, and paste operations are often important. In speech processing, most systems allow the user to mark symbols from the international phonetic alphabet. Many systems allow other arbitrary text labels to be associated with given points on the waveform. It is useful to be able to compare several sound files at once, optionally forcing them all to follow the same time scale. Many systems allow the user to redraw the waveform by hand on the screen. The sound file can be played back transposed in frequency or faster and slower than normal. Sound files need to be converted from one format to another. If the sound is

transformed, then the transform data can be shown on the screen, with the cursor of the sound file tied to the Cursor in the transformed data.

DIGITAL MUSIC-MAKING

Musical Instrument Synthesizers

The core of a digital music synthesizer is some hardware for creating sound. This may be concentrated in a special-purpose chip set; it may be a general-purpose DSP chip running special software; or it may be a board containing several chips, each of which implements one or more "voices."

The earliest digital synthesizers were usually built as a single unit with a keyboard interface controlling the sound generator. In current single-unit keyboard synthesizers, the hardware enclosure typically features an LCD screen, switches, knobs, sliders, a joystick, track ball, and a "pitch bend wheel" or two, used for creating life-like effects such as glissando. Nowadays, it is common to have a musical controller separate from the tone-generation unit, which can be relegated to a backstage location or a different room in the recording studio. Typical controllers include a separate keyboard, guitar interface, breath controller, violin-style input device, or woodwind controller, to name the most common. Virtual reality research is producing some inventive controllers as well.

Some synthesizers, especially those from Roland, offer a video connector. At this writing, graphic user interfaces for the complicated controls of a digital -music synthesizer unfortunately remain primitive. Almost all synthesizers now on the market operate on a MIDI network, explained below, so one finds one or more MIDI connectors.

The line between the disk-based audio systems mentioned above and boards which can generate music with personal computers is now blurred
• Boards from companies such as Turtle Beach, Ariel, or DigiDesign can be used for sound editing or music synthesis and performance. The personal computer with add-on board and associated software thus serves as a digital synthesizer as well.

A sampler is a synthesizer that uses stored (rather than synthesized from scratch) sounds. Some samplers only play back stored sounds; others allow the user to record new sounds. The sampler may encode sound in some fashion, so it may not store PCM samples directly. Still, in sampling (used here in the music industry sense), the sounds are basically recorded by the musician and stored in the synthesizer. There are now fine sets of sampled sounds available on CDs, such as Denny Jaeger's CDs or the set from McGill.

In the narrow sense, a synthesizer uses a synthesis technique to generate sound (although people commonly speak of a sampling synthesizer). A synthesis technique is an algorithm for generating digital samples which, when played through appropriate conversion hardware and loudspeakers, sound more or less like the desired musical sounds. Techniques such as additive synthesis, FM, waveshaping, and granular synthesis were mentioned earlier in this chapter.

One view of the synthesizer hardware developed in the 1970s and early 1980s is that manufacturers competed to find new, unusual (and patentable) synthesis techniques that allowed for musical control while still offering low-cost implementation in hardware. Typical for this era, the Yamaha DX-7 was a pure FM synthesizer and a great commercial success in part because of its "FM" sound. Nowadays, it is more common

for a synthesizer to create a sound by overlaying several sounds or by chaining parts of sounds to make one note. Each layer or each chained part may come from a different synthesis technique.

Whether the synthesizer is based on sampling or on one or more synthesis techniques, it is a nontrivial task to "voice" a synthesizer. One starting point is steady-state or time-varying (Fourier) analysis of recorded waveforms. Another way to work is to study the physical properties of, say, the bowed string or the clarinet reed. Another is to explore parameter settings to find some interesting sounds which are then refined. Much trial and error is involved, with repeated listening (and/or comparison with recorded sounds) as settings are changed. Synthesizer manufacturers sometimes develop an in-house development system for voicing. For everyday musicians to voice a synthesizer using the LCD on the front panel of the synthesizer case is a frustrating experience, due to the small size of the LCD and the primitive user interface design usually found. Many of the software packages available for the Macintosh and PC computers now allow the user to voice a synthesizer on the screen.

MIDI Protocol

In the early 1980s, several music instrument manufacturers agreed on a networking standard for musical instruments called MIDI, the Music Instrument Digital Interface. The standard is now maintained by MMA, the MIDI Manufacturer's Association and disseminated by the International MIDI Association (IMA). The specification calls for certain hardware connections, using a 5-pin DIN connector. There are three kinds of connections allowed: in, out, and "thru." A thru connector

provides a direct copy of the input signal. I would like to mention in passing that the MIDI network, although it has been made to work, is not the computer scientist's model of a well-designed network. It is to be expected that some super-set of MIDI will appear on the market. Already companies like Lone Wolf have attempted a subset. to bring to the market an optical network which includes MIDI as a subset.

The MIDI software specification involves 8 data bits, a start bit, and a stop bit, for a total of 10 bits transmitted at a rate of 31.25 kbaud. A message consists of one Status byte followed by zero or more Data bytes.

Status Byte (hex)	**Data Byte 1**	**Data Byte 2**	**Meaning**
8n	0k	0v	Note Off
9n	0k	0v	Note On
An	0k	0v	Polyphonic Key Pressure (aftertouch)
Bn	0c	0v	control change
Cn	0p		program change
Dn	0v		Channel pressure (aftertouch)
En	0v	0v	Pitch changes, LSB + MSB

Notes:

n: Voice Channel Number (1-16)

k: Note Number (0-127, from bottom of the keyboard range to the top)

v: Velocity (0-127), or pressure value, or control value

c: Controller, such as breath controller, soft pedal, or sustain pedal

p: Program Number (0-127)

Table 4.3: MIDI Channel Voice Messages

MIDI devices, such as tone generators, can be connected in networks such as chains or trees. Each device can listen to one or more MIDI channels. All data and mode messages are sent to all receivers, but the messages include a channel number so that only some receivers may act on specific messages.

The messages defined for musical events, such as note on, note off, and pitch bend change, are summarized in Table 4.3. The key number represents keys from the bottom of the keyboard range to the top. Velocity means the speed with which the key is struck and generally controls attack characteristics, overall amplitude, and spectrum of the note. The polyphonic key pressure message is sent by devices such as keyboards that can measure the pressure applied as each key is held.

The pressure for each key can be sent separately so that individual notes can be modified in performance. A channel pressure message comes from a device that can measure the pressure from its sensors, but can send only one pressure detected (usually the maximum)..

A program change message causes the synthesizer to select one of 128 voices. In the early years of MIDI, each manufacturer assigned arbitrary voices to these program numbers. The recent General MIDI specification includes a 128-voice Instrument Pitch Map. A melody recorded on one General MIDI synthesizer's xylophone sound, for example) will also be played back using a xylophone, and not a tuba, on some other General MIDI synthesizer.

Four Mode messages (not shown in the table) determine, among other things, whether the instruments' voices will be assigned to incoming

notes in a monophonic (single melody) or polyphonic (several voices at once) fashion.

There is also provision for common messages (sent to all receivers), real-time messages (for synchronization), and for system exclusive (sysex) messages. System exclusive is essentially a generalized escape mechanism for messages of arbitrary length.

MIDI is not limited to hardware systems. Indeed, the acceptance of MIDI made possible the proliferation of software programs running on the Amiga, Macintosh, Atari, and PC. MIDI software includes sequencer programs, with which the musician can record, play back, view, and alter musical events, working with music notation, piano-roll notation, text displays of MIDI commands, and the like.

Figure 4.13 shows a simple melody. Table 4.4 shows the basic MIDI messages for playing back the melody from a synthesizer. In the figure, all of the messages are sent out over channel 0. The note numbers and velocities are given in decimal representation. A note-on message with a velocity of 0 is the same as a note-off message. Time in the first column is in milliseconds, with 90 quarter notes to the second. Note that the first note occurs after a 3-second delay from the start of playback.

The original MIDI specification dealt primarily with real-time music performance. To store a performed or composed sequence, the programmer must implement a representation of time. To represent time in music, there are basically two possibilities—absolute time and delta time. With delta time, the time interval elapsed since the previous event is recorded. With absolute time, time elapsed since the beginning of the composition is represented. In the most general terms, both kinds of time

are identical. But in practical implementation, delta time has the advantage that a whole sequence can be moved as one unit; only the start time of the unit must be changed.

The disadvantage of delta time is that it can lead to incremental errors. Suppose the composer specifies three notes, which together should occupy 1 second. An integer representation of 1/3 second is rounded off. After three such delta units, there is a small time discrepancy, which can build up in a composition lasting 10 or 15 minutes.

Absolute time coding avoids those errors, but makes editing harder. (The proposed MHEG standard for multimedia and hypermedia objects also allows for time to be represented in what are essentially absolute and delta times. Some way of representing time with rational numbers is also needed, as the example of three notes in 1 second shows.

Early software sequencers stored the recorded data in proprietary file formats. Ultimately, an extension to the MIDI specification called Standard MIDI Files was established. The Standard Midi File format adopted a delta time representation, with time specified for each MIDI event. The file header effectively specifies the tempo. The delta time is a variable-length number between 0 and 0*x*FFFFFFF that specifies the number of time units given in the file header.

MIDI has also been extended to control theater lighting (MIDI Show Control). Yavelow's "bible" is the best current source of information on MIDI, computers, and (Macintosh) software. To follow the current scene, one should consult magazines such as Keyboard, MIX, and Electronic Musician.

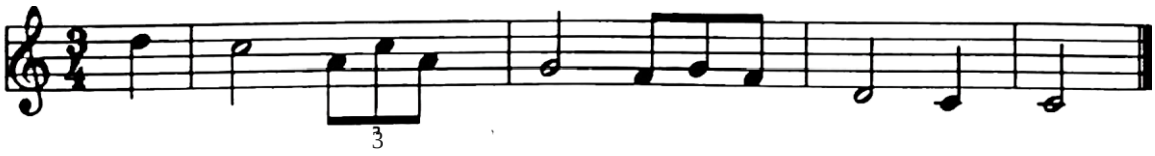


Figure 4.13 A short musical example ("Justin's Lullaby," © 1990 John Strawn).

Notation produced with the Finale program, courtesy of Robert Duisberg.

Table 4.4 gives the corresponding MIDI messages

Time	*MIDI Status Byte*	*MIDI Pitch Byte (Data Byte 1)*	*Velocity (Data Byte 2)*	*Meaning*	*Pitch*
3642	9	74	49	note on	D
4149	9	74	0	note off	
4222	9	72	49	note on	C
5307	9	72	0	note off	
5380	9	69	49	note on	A
5549	9	69	0	note off	
5573	9	72	49	note on	C
5742	9	72	0	note off	
5765	9.	69	49	note on	A
5934	9	69	0	note off	
5958	9	67	49	note on	G
7045	9	67	0	note off	
7117	9	65	49	note on	F
7286	9	65	0	note off	
7310	9	67	49	note on	G
7478	9	67	0	note off	
7503	9	65	49	note on	F
7672	9	65	0	note off	
7696	9	62	49	note on	D
8783	9	62	0	note off	
8856	9	60	49	note on	C
9362	9	60	0	note off	
9435	9	60	49	note on	C
10520	9	60	0	note off	

Table 4.4: MIDI Messages for Figure 4.13

SPEECH RECOGNITION AND GENERATION

Speech is one of the main channels for human communication and thus must be handled carefully in any multimedia communications system. In contrast to what has been discussed thus far about music, a major criterion in speech is intelligibility. For example, "telephone-quality" speech has a bandwidth limited to around 200-3400 Hz. An 8-kHz sample rate results in a 68 kb/sec bit rate for PCM speech, far smaller than required for music PCM.

Speech Production

The organs involved in speech include the larynx, which encloses loose flaps of muscle called vocal cords. The puffs of air that are released create a waveform which can be approximated by a series of rounded pulses. The waveform created by the vocal cords propagates through a series of irregularly shaped tubes, including the throat, the mouth, and the nasal passages. At the lips and other points in the tract, part of the waveform is transmitted further, and part is reflected. The flow can be significantly constricted or completely interrupted by the uvula, the teeth, and the lips.

A voiced sound occurs when the vocal cords produce a more or less regular waveform. The less periodic, unvoiced sounds involve turbulence in which some part of the whole tract is tightened.

Vowels are voiced sounds produced without any major obstruction in the vocal cavity. In speech, formants (introduced above) are created by the position of tongue and jaw, for example. In separating vowels, the first

three formants are the most significant. In the male, the fundamental frequency of voiced sounds is around 80-160 Hz, with three formants around 500, 1500, and 2500 Hz. The fundamental of the female is around 200 Hz and higher, with the formants perhaps 10 percent higher than those of the male. Consonants arise when the vocal tract is more or less obstructed. Sounds at the level of consonants and vowels are collectively known as phonemes, the most basic unit of speech differentiation, analysis, and synthesis. The next level up from phonemes is the diphthong and the syllable, then the word.

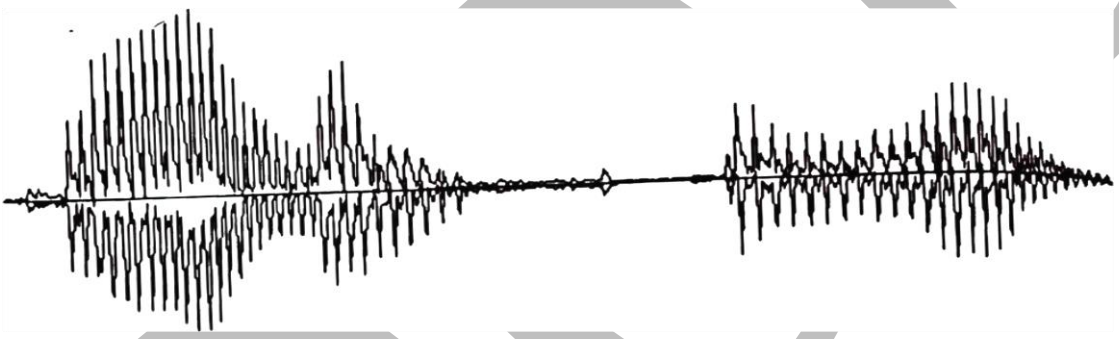


Figure 4.14 (a): The utterance "Golly, Scully"

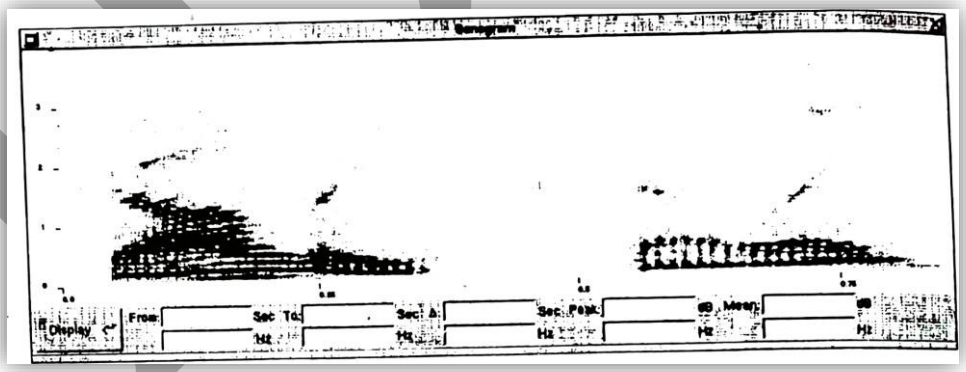


Figure 4.14 (b): A sonogram of the same utterance, prepared using the program Sonogram by Hiroshi Momose

Figure 4.14 (b) shows a sonogram, a time-varying representation of a speech signal. The regions of high energy appear dark. The vertical stripes in the dark region correspond to individual pulses from the vocal cords. The change in the position of the darkest areas from left to right corresponds to the changes in formants.

The SPASM system developed by Cook combines models of the glottal waveform and noise sources in the vocal tract with modelling of the tubes and obstructions in the vocal tract. The resulting articulatory model is implemented with a GUI, including cross-section of the head, to permit synthesis of spoken and singing voice.

Encoding and Transmitting Speech

The simplest way to encode speech is to use PCM, discussed above. The 8-bit, 8-kHz standard for speech is of significantly lower quality than what is required for music. Still, at the nominal 64-kb/sec rate for speech, if one bit per sample can be saved, then the total saving is 8 kb/sec. Methods for lowering the bit rate thus remain an active area of research. The ADPCM method discussed above can easily save 2 to 4 bits per sample.

PCM, ADPCM, and related methods attempt to describe the waveform itself. There are other methods, such as the subband coding discussed above under MPEG. We now turn to another class of methods, called voice coders, or vocoders. The human vocal tract can be simplified by assuming, for example, that the source of vibration for voiced sounds is not affected by the rest of the vocal tract. The series of filters that model the vocal tract can be modelled such that if one filter changes, there is no effect on the others. Under such conditions, we can calculate the voice model coefficients independently of the fundamental frequency or the

voiced/unvoiced decision. We can also reasonably assume that formants change quite slowly compared to the rate of individual pulses from the vocal tract and transmit the filter coefficients at a slower rate.

The channel vocoder pioneered by Dudley analyses speech as a bank of filters. The driving function for synthesis is noise or a series of pulses like those generated by the vocal cords. The filter coefficients, the fundamental frequency, and the voiced/unvoiced decision are transmitted. Research on the channel vocoder ultimately led to the phase vocoder implementation mentioned above.

Linear prediction, also mentioned above, models the vocal tract as a source followed by a series of filters. Those filters can be modelled as a series of tubes, and the tube parameters can be transmitted. There is, unfortunately, no intuitive relationship between tube parameters and, say, the spectrogram representation, but LPC is certainly adequate for compressing speech for reproduction in chips. One transmits the pitch period, gain, the voice/unvoiced decision, and a dozen or so filter coefficients.

In a different kind of system, both encoder and decoder can contain a lookup table. Each table entry is a vector containing a series of samples. Rather than transmit the samples, one can transmit just the index into the table. If the exact sequence of samples cannot be found, the closest vector is transmitted. This method can be used to transmit the waveform itself or sequences of coefficients for a vocoder.

As we have seen, the basic data rate is 64 kb/sec (CCITT G.211) for 8-bit PCM. With ADPCM, 4 to 6 bits per sample are transmitted, for 32 to 48 kb/sec there is a 32 kb/sec CCITT standard G.721 for ADPCM. Some

subband coding systems operate as low as 16 kb/sec. For higher-quality speech with subband coding, there is CCITT G.722 for 50-7000 Hz at a 64-kb/sec rate. For various methods of coding, bit rates can fall as low as 2400-bit/s, but with a corresponding reduction in quality. Improvements in quality and lowering bit rate are being driven (as always) by military research and the usual telephone companies, but also by factors such as the desire to incorporate voice with other data, such as in ISDN, or the need to scrunch more channels from cellular networks.

Speech Synthesis

A major driving force in speech synthesis has come from text-to-speech (TIS). A TIS system assumes that the text already exists in machine-readable form, such as an ASCII file. The machine-readable form is possibly obtained from optical character recognition. TTS converts the text symbols to a parameter stream representing sounds. This includes expanding common abbreviations, such as "Pres.," and symbols like Also, the system figures out how to handle numbers: 1492 can be read as a date, and even the dollar amount $1492.00 could be read starting with "fourteen hundred…" or "one thousand four hundred…" After creating a uniform symbol stream, the system creates initial sound parameter representation, often at the word level - some words may simply be looked up in a dictionary.

Other parts of the stream are broken down into morphemes, the syntactic basic units of the language. With luck, a group of text symbols corresponds to one morpheme. It is often the case that there is a regular mapping between the symbols of such a group and some sound, in which case the group can be turned, into sound. As a last resort, the system converts individual text symbols to sound using rules. The system synthesizes sounds from the parameter string based on an

articulatory model, or using sampled sounds, LPC, or formants. The synthesis system may store units at the level of phonemes, diphthongs, syllables, or words.

The sounds are concatenated. Then higher-level elements of speech such as prosody (the rise and fall of pitch), overall emphasis (e.g., whisper, shout), and glottal stops are added. Syntactical analysis provides the basis for adjustments in clause ends or sentence ends, e.g., the rise in pitch for a question.

Speech Recognition

A speech recognition system starts by breaking speech down into a parametric representation. The first step is to isolate speech segments in time. (One big problem, as in music, lies in the fact that in the acoustic signal, there are no discrete units.) The speech signal is parameterized as the outputs of a bank of bandpass filters, or as LPC coefficients, or some form derived from LPC coefficients, such as cepstral coefficients (the log of the transform of the spectrum). Individual frames of data typically last on the order of 10-30 msec. The frames are matched against a template, using some measure of goodness of fit between input and template. The template typically contains raw spectral data or vector quantized spectral data. The measure of fit can be improved by dynamic time warping, in which the time-varying input signal is measured against several time-varying templates. The time scale of the input is modified non-monotonically so that the representation of the input best matches the representation of the template. In this manner, subtle differences in timing that would otherwise throw off template matching can be removed. This is similar to the WordFit time adjustment scheme discussed in digital signal processing, above.

Another algorithm for improving identification involves Hidden Markov Models (HMM). The core of an HMM identification system is a finite-state machine, with probabilities associated with the transition from one state to another. But the states of the machine cannot be directly observed. Instead, a finite number of observations can be made about the current state of the state machine. The observations are stochastically related to the actual states. There is an algorithm for deriving the probability that a given sequence of observations was generated by a given sequence of states. In an HMM-based system, each element (e.g., phoneme) has one model representing it, that is, one state machine with associated probable initial state, transition probabilities, and observation probabilities. There is also an algorithm for deciding which of a set of models produced the speech being analysed. Recently, neural networks have come to be used for speaker identification and speaker verification.

Speech recognition systems have an easier job if all speakers speak the same text. Isolated words are easier, connected speech is harder. Handling any arbitrary speaker from the general population is harder. Allowing an arbitrary vocabulary makes the task harder still. Currently, typical systems achieve an accuracy in the mid 90 percent range for dictionaries of several hundred words spoken by different speakers.

DIGITAL AUDIO AND THE COMPUTER

It is now common to find audio capabilities in many computers, large and small. In this last section, we review the capabilities for audio and music at various levels of quality.

One fundamental capability is audio storage: 8-bit audio at an 8-kHz sample rate consumes 1 KB for 1 second of sound. CD-quality audio

(stereo, 16-bit linear PCM, 44.1-kHz sample rate) consumes 176 KB per second, and stereo sound at the professional rate of 48 kHz eats almost 200 kbytes per second. Sound storage on disk thus requires large disks. Sound playback and recording require a DAC and ADC, respectively. The 8-bit hardware on many computers is adequate for speech and basic algorithm testing, but not for professional music or recording. Plug-in boards with 16-bit DACs and ADCs are the solution if such hardware is not provided as part of the basic system. Even better audio quality can be had with external conversion units, using, for example, a SCSI connector.

For digital input and output, a connector for AES/EBU transmission and/or an SPDIF connector is required. Often, the DACs and ADCs on an external DAT player with an appropriate interface can be used instead of dedicated hardware.

For real-time processing of sound, plug-in cards and external units are available with commercially available DSP chips. With a single chip, typically a stereo stream can be input, processed, and output. In such a scheme, the sound does not necessarily go to disk. Such systems can also generate three-dimensional audio, for example. There is a tendency nowadays to use RISC or CISC chips instead of DSP chips for dedicated processing. In some systems, a single RISC chip is fast enough to provide the generalized compute power as well as extra cycles for sound processing. Using two RISC chips in parallel, one for general computing and one for real-time applications, has the advantage that the software development is the same for both. (When a DSP chip is used, the software environment is usually radically different from that for the computer's CPU.)

For music synthesis, there are also plug-in cards currently available implementing FM synthesis, Sampling, and other synthesis techniques. Editing sound requires a good graphics system; for editing transform data, graphics accelerators are often recommended, as the amount of data can be enormous. Sound can be edited on the screens of portable computers, but a large crisp colour monitor is to be preferred.

Given all the power in and fanfare surrounding UNIX workstations, such as the Sun or SPARC, one would expect them to support music easily. But UNIX is not a real-time operating system, and musicians complain when their music stops dead while the operating system services something else.

Finally, a remark about sound in personal computers and multimedia is in order. For many years, the basic capabilities implied in this chapter have been included in plug-in boards for the PC, such as the SoundBlaster. With the release of documents such as Multimedia PC Specification Version 1.0 (Microsoft), we can expect the computer industry to follow a path well known to digital audio and computer music specialists. The software protocols will become standardized, as will music and sound exchange formats. The quality of the sound coming into and out of the system will improve. The capabilities of the system will be expanded to include more and more sophisticated techniques.

SUMMARY

- One of the disadvantages of audio compared with other media is the ephemeral nature of sound

- The lower limit of the dynamic range of human hearing is at the threshold of audibility, and the upper limit is the threshold of pain (or damage).

- A sampler is a synthesizer that uses stored (rather than synthesized from scratch) sounds

- The spectrum and other parts of an auditory event give rise to the percept labelled timbre

- A critical band is characterized by a center frequency and by bandwidth.

- The difference between a quantized representation and an original analog signal is called the quantization noise

- There is always the fallback position of connecting the analog output of one digital machine to the analog input of another digital machine.

- If a signal is delayed and added back into itself, an echo can result. If many such echoes are generated at the right time and scaled at the right amplitudes, the effect of reverberation can be generated

- The MIDI software specification involves 8 data bits, a start bit, and a stop bit, for a total of 10 bits transmitted at a rate of 31.25 kbaud

- Sounds at the level of consonants and vowels are collectively known as phonemes, the most basic unit of speech differentiation, analysis, and synthesis. The next level up from phonemes is the diphthong and the syllable, then the word

- 8-bit audio at an 8-kHz sample rate consumes 1 KB for 1 second of sound

- For digital input and output, a connector for AES/EBU transmission and/or an SPDIF connector is required

UNIT END EXCERCISES

1. What is a sampler?
2. How can one identify a quantization noise?
3. Write a note on hearing of a human being.
4. What is MIDI?
5. How is a time-domain sampled represented?
6. What is binaural hearing?
7. Describe with the help of diagram/s encoder and decoder of MPEG.
8. What are stereophonic and quadrophonic signal processing techniques?
9. Write a short note on speech recognition.
10. Explain the role of sigital audio in computers.

ADDITIONAL REFERENCE

1. MULTIMEDIA SYSTEMS, John F. Koegel Buford, University of Massachusetts Lowell, Pearson, Fourteenth Impression

5

Video Technology

Unit Structure

Objectives
Introduction
Raster Scanning Principles
Sensors for TV Cameras
Color Fundamentals
Color Video
Video Performance Measurements
Analog Video Artifacts
Video Equipment
Worldwide Television Standards
Summary
Unit End Exercises
Additional Reference

OBJECTIVES

In this unit, you will understand:

- Video Resolution and its types
- Color Mixing fundamentals
- NTSC, PAL and SECAM Video Formats
- Performance of Video and how to measure resolution, response, noise and color performance
- Analog Video artefacts and video equipment

INTRODUCTION

Principles of analog video as they exist in the television industry needs to be understood in order to pursue the discussion of multimedia. A good knowledge of analog video nomenclature, characteristics, performance and limitations will be essential and this chapter will explain analog video fundamentals as they relate to the uses of video in digital formats.

Most things are analog in nature – real images and sounds are based on light intensity and sound pressure values, which are continuous functions in space and time. For television, we must convert images and sounds to electrical signals. This is done by appropriate use of sensors, called transducers. Sensors for converting images and sounds to electronic signals are typically analog devices, with analog outputs. The world of television and sound recording is based on these devices.

RASTER SCANNING PRINCIPLES

Raster

The purpose of a video camera is to convert an image in front of the came into an electrical signal. An electrical signal has only one value at any instant in time—it is one-dimensional, but an image is two-dimensional having many values at all the different positions in the image.

Conversion of the two-dimensional image into a one-dimensional electrical signal is accomplished by *scanning* that image in an orderly pattern called a *raster*.

Video signal

With scanning, we move a sensing point rapidly over the image-fast enough to capture the complete image before it moves too much. As the sensing point moves,

the electrical output changes in response to the brightness or colour of the image point beneath the sensing point.

The varying electrical signal from the sensor then represents the image as a series of values spread out in time—this is called a *video signal*.

Raster scanning pattern

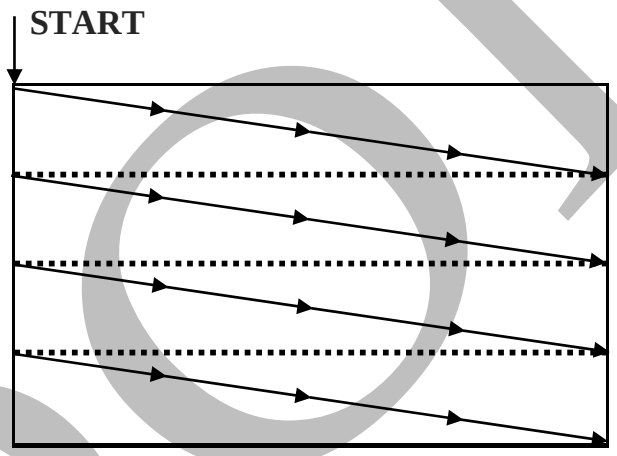


Fig 5.1: Raster Scanning

Figure shows a raster scanning pattern. Scanning of the image begins at the upper left corner and progresses horizontally across the image, making a scanning line. At the same time, the scanning point is being moved down at a much slower rate. When the right side of the image is reached, the scanning point snaps back to the left side of the image. Because of the slow vertical motion of the scanning point, it is now a little below the starting point of the first line. It then scans across again on the next line, snaps back, and continues until the full height of the image has been scanned by a series of lines.

During each line scanned, the electrical output from the scanning sensor represents the light intensity of the image at each position of the scanning point. During the snap-back time (known as the horizontal blanking interval) it is customary to turn off the sensor so zero-output (or blanking level) signal is sent out. The signal from a complete scan of the image is a sequence of line signals, separated by horizontal blanking intervals. This set of scanning lines is called a ***frame***.

Aspect Ratio

An important parameter of scanning is the aspect ratio—it is the ratio of the length of a scanning line horizontally on the image, to the distance covered vertically on the image by all the scanning lines. Aspect ratio can also be thought of as the width-to-height ratio of a frame. In present-day television, aspect ratio is standardized at 4:3. Other imaging systems, such as movies, use different aspect ratios ranging as high as 2:1 for some systems.

Sync

If the electrical signal from a scanning process is used to modulate the brightness of the beam in a cathode-ray tube (CRT) that is being scanned exactly the same way as the sensor, the original image will be reproduced.

This is what happens in a television set or a video monitor. However, the electrical signal (s) sent to the monitor must contain some additional information to ensure that the monitor scanning will be in synchronism with the sensor's scanning. This information is called sync information, and it may be included within the video signal itself during the blanking intervals, or it may be sent on a separate cable (or cables) just for the sync information.

Resolution

Resolution is the ability of a television system to reproduce fine detail in the scene. It is expressed separately for *horizontal* and *vertical* directions.

Horizontal Resolution

As the scanning point moves across one line, the electrical signal output from the sensor changes continuously in response to the light level of the part of the image that the sensor sees. One measure of scanning performance is the *horizontal resolution* of the pickup system, which depends on the size of the scanning sensitive point. A smaller sensitive point will give higher resolution.

Thus, a system that is said to have a horizontal resolution of 400 lines can reproduce 200 white and 200 black lines alternating across a horizontal distance corresponding to the height of the image.

Scanning across a pattern of vertical black and white lines produces a high-frequency electrical signal. It is important that the circuits used for processing or transmitting these signals have adequate bandwidth for the signal. Without going into the details of deriving the numbers, broadcast systems require a bandwidth of 1 MHz for each 80 lines of horizontal resolution. The North American broadcast television system is designed for a bandwidth of 5.5 MHz, and this has a theoretical horizontal resolution limit of 360 lines.

Vertical Resolution

The vertical resolution of a video system depends on the number of scanning line used in one frame. The more lines there are, the higher is the vertical resolution. Broadcast television systems use either 525 lines (North America and Japan) or 625 lines (Europe, etc.) per frame. In a sense, the vertical resolution response of television is not an analog process because a discrete number of samples are

taken vertically—one for each scanning line. The result is that the vertical resolution response of television often displays sampling artefacts such as aliasing.

A small number of lines out of each frame (typically 40) are devoted to the vertical blanking interval. Both blanking intervals (horizontal and vertical) were originally intended to give time for the scanning beam in cameras or monitors to retrace before starting the next line or the next frame. However, in modern systems they have many other uses, since these intervals represent non-active picture time where different information can be transmitted along with the video signal.

Interlace

For motion video, many frames must be scanned each second to produce the effect of smooth motion. In standard broadcast video systems, normal frame rates are 25 or 30 frames per second, depending on the country you are in. However, these frame rates - although they are high enough to deliver smooth motion—are not high enough to prevent a video display from having flicker. In order for the human eye not to perceive flicker in a bright image, the refresh rate of the image must be higher than 50 per second. However, to speed up the frame rate to that range while preserving horizontal resolution would require speeding up of all the scanning, both horizontal and vertical, therefore increasing the system bandwidth. To avoid this difficulty, all television systems use ***interlace***.

SENSORS FOR TV CAMERAS

It is possible to make a television camera as described above with a single light-sensitive element; however, that proves not to be an effective approach because the sensor only receives light from a point in the image for the small fraction of time that the sensor is looking at that point. Light coming from a point while the sensor

is not looking at that point is wasted, and this is most of the time. The result is that this type of video sensor has extremely poor sensitivity—it takes a large amount of light to make a picture. All present-day video pickup devices use an i*ntegrating* approach to collect all the light from every point, of an image all the time. The use of integration in a pickup device increases the sensitivity thousands of times compared to non-integration pickup. With an integrating pickup device, the image is optically focused on a two-dimensional surface of photosensitive material, which is able to collect all the light from all points of the image all the time, continuously building up an electrical charge at each point of the image on the surface. This charge is then read out and converted to a voltage by a separate process which scans the photosensitive surface.

Without going into all possible kinds of pickup devices, there are two major types in use today which differ primarily in the way they scan out the integrated and stored charge image. Vacuum-tube pickup devices (vidicon, saticon, etc.) collect the stored charge on a special surface deposited at

COLOR FUNDAMENTALS

However, most real images are in colour, and what we really want is to reproduce the image in colour. Colour video makes use of the tri-stimulus theory of colour reproduction, which says that any colour can be reproduced by appropriate mixing of three primary colours. In grade school we learned to paint colours by doing just that—mixing the three colours: "red," "blue," and yellow. (The use of quotes on "red" and "blue," but not yellow, is deliberate and will be explained below.) These paint colours are used to create all possible colours by mixing them and painting on white paper. This process is known technically as subtractive colour mixing - because we are starting with the white paper, which reflects all colours equally, and we are adding paints whose pigments filter the reflected white light to subtract

certain colours. For example, we mix all three paint primaries to make black—meaning that we have subtracted all the reflected light (the paper looks black when no light at all is being reflected).

There is a different system of primary colours that is used when we wish to create colours by mixing coloured lights. This is the additive primary system, and those colours are red, green, and blue. If we mix equal parts of red, green and blue lights, we will get white light. Note that two of the additive primary colour names seem to be the same as two of the subtractive primaries—"red" and "blue." This is not the case—red and blue are the correct names for the additive primaries, but the subtractive paint primaries "red" and "blue" should technically be named, respectively, magenta, which is a red-blue colour, and cyan, which is a blue-green colour.

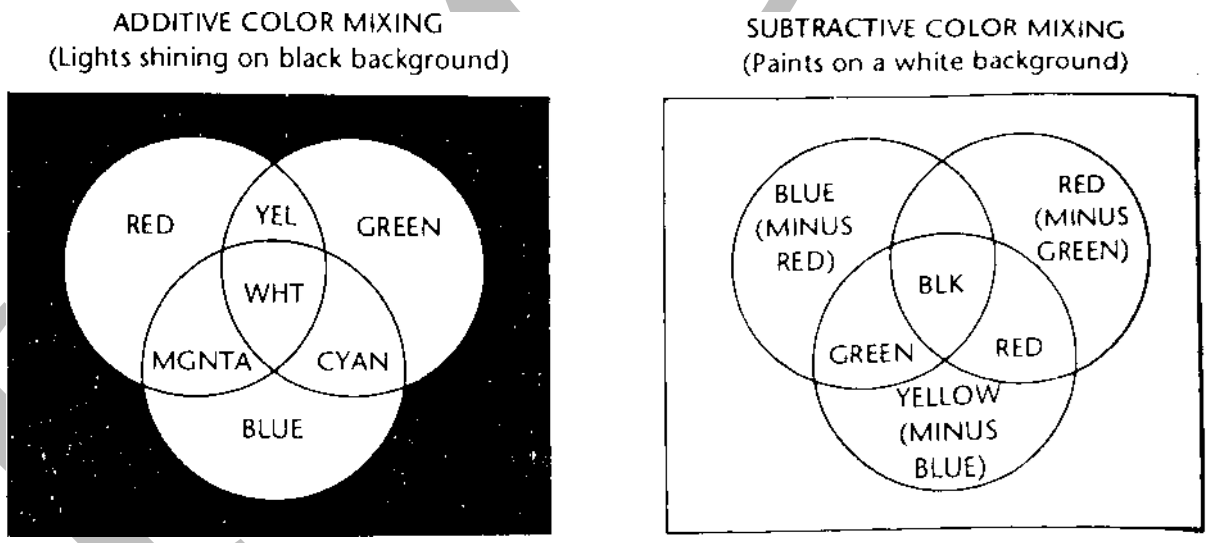


Figure 5.2: Additive and Subtractive Colour Mixing

Subtractive "blue" filter, anything red will appear black—it has been filtered out. Similarly, the subtractive "red" removes green light, and the subtractive yellow removes blue light. Therefore, when we mix two subtractive colours, such as "blue"

and yellow, we have removed both the red and the blue from the reflected light—leaving the green light—so mixing "blue" and yellow paint makes green. You can try the other combinations yourself to convince you that it agrees with what you learned in grade school.

COLOR VIDEO

Let's return to the additive system, because that is the basis for colour video systems. Video is an additive colour system because the colour CRT used for display creates three light sources which are mixed to reproduce an image. A colour CRT mixes red, green, and blue (RGB) light to make its image. The colours are produced by three fluorescent phosphor coatings, which are on the faceplate of the CRT. Typically, these are scanned by three electron guns, which are arranged so that each of them impinges on only one of the phosphors. (There are many ways to do this.) The intensity of each of the guns is controlled by an electrical signal representing the amount of red, green, or blue light needed at each point of the picture as the scanning progresses.

Three-Sensor Colour Camera

So a colour video camera needs to produce three video signals to control the three guns in a colour CRT. A conceptually simple way to do this is to split the light coming into a colour video camera into three paths, filter those paths to separate the light into red, green, and blue, and then use three video pickup devices to create the necessary three signals. In fact, many video cameras do exactly that, as shown in Figure.

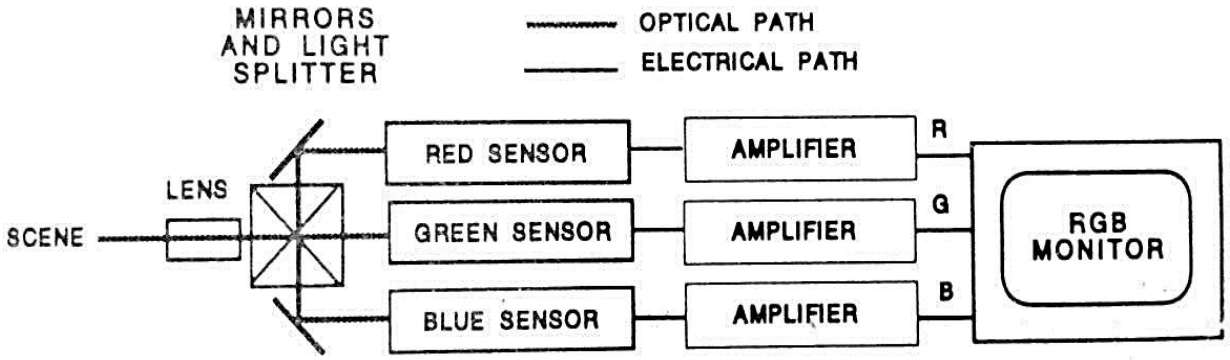


Figure 5.3: Block diagram of a three-sensor RGB colour video camera

This kind of camera, known as a three-tube or three-sensor camera, is complicated and expensive because of the three parallel paths. A lot of the difficulty arises because the three sensors must be scanned in exact synchronism and exact physical relationship so that at any instant of time the three output signals represent the colour values from exactly the same point in the image. This calls for extremely demanding electrical and mechanical accuracy and stability in a three-sensor camera design. The process for obtaining these exact relationships is known as registration. If a camera is not in exact registration, there will be colour fringes around sharp edges in the reproduced picture. In spite of these difficulties, three-sensor cameras produce the highest quality images, and this approach is used for all the highest performance cameras.

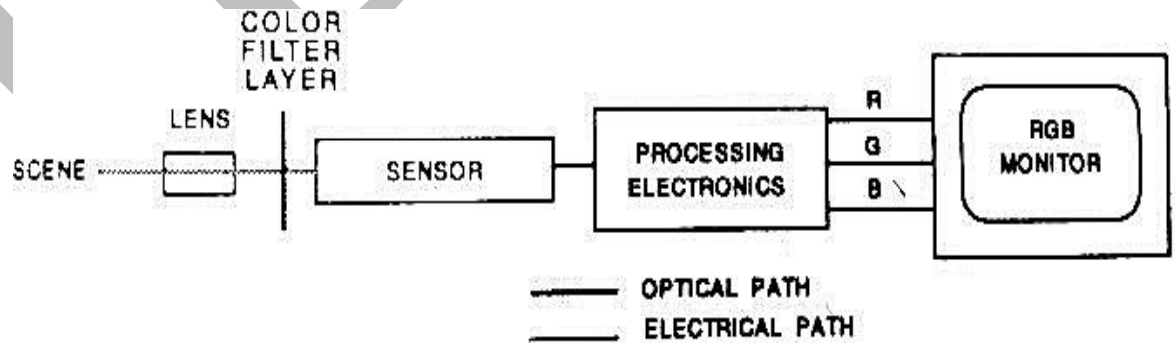


Figure 5.4: Single-sensor colour video camera

Single Sensor Colour Cameras

There also are single-sensor colour cameras, as shown in Figure 5.4. These use a system of filtering that splits the incoming light into spots of coloured light, which appear side by side on the surface of the sensor. When the sensor is scanned in the normal way, the electrical output for each spot in the image will consist of three values coming in sequence, representing the red, green, and blue values for that point. Because of the critical relationship required between the sensor and the colour filter, it is customary to build the colour filter right on top of the sensor's storage layer. Electronic circuits are then used to separate the sequential output from the sensor as it is scanned into the required three separate signals. This approach is effective if the three spots of colour can be small enough that they will not reduce the resolution of the final reproduction. Because that requires a threefold increase in the resolution of the sensor used (and that is difficult to come by) single-sensor cameras often are a compromise with respect to resolution. However they are still the simplest, lowest cost, and most reliable cameras, and therefore single-sensor colour cameras are widely used. All home video cameras are of the single-sensor type. Solid-state sensors are particularly suited to making single-sensor cameras. Because the resolution capability of solid-state sensors is steadily improving, single-sensor cameras are getting better.

Colour Television Systems—Composite

The colour cameras just described were producing three output signals—red, green, and blue. This signal combination is called RGB. Most uses of video involve more than a single camera connected to a single monitor, as we had in Figures and 5.4. The signal probably has to be recorded; we may wish to combine the outputs of several cameras together in different ways, and almost always we will want to have more than one viewing monitor. Therefore, we will usually be concerned with a colour video system, containing much more than cameras.

In RGB systems, all parts of the system are interconnected with three parallel video cables, one for each of the colour channels. However, because of the complexities of distributing three signals in exact synchronism and relationship, most colour TV systems do not handle RGB (except within cameras), but rather the camera signals are encoded into a composite format which may be distributed on a single cable. Such composite formats are used throughout TV studios, for video recording, and for broadcasting. There are several different composite formats used in different countries around the world—NTSC, PAL, SECAM—and they will be covered specifically in the next section. Here we will concentrate on some of the conceptual aspects of composite colour video systems.

Composite colour systems were originally developed for broadcasting of colour signals by a single television transmitter. However, it was soon found that the composite format was the best approach to use throughout the video system, so it is now conventional for the composite encoding to take place inside the camera box itself before any video signals are brought out. Except for purposes such as certain video manipulation processes, RGB signals do not exist in modern television plants.

All composite formats make use of the luminance/chrominance principle for their basic structure. This principle says that any colour signal may be broken into two parts—luminance, which is a monochrome video signal that controls only the brightness (or luminance) of the image, and chrominance, which contains only the colouring information for the image. However, because a tri-stimulus colour system requires three independent signals for complete representation of all colours, the chrominance signal is actually two signals, called colour differences.

Luminance and chrominance are just one of the many possible combinations of three signals which could be used to transmit colour information. They are obtained by a linear matrix transformation of the RGB signals created in the camera. The matrix transformation simply means that each of the luminance and chrominance signals is an additive (sometimes with negative coefficients) combination of-the original RGB signals, in a linear transmission system there are an infinity of possible matrix transformations that might be used; we just need to be sure that we use the correct inverse transformation when we recover RGB signals to display on a colour monitor. Psych visual research (research into how images look to a human viewer) has shown that by carefully choosing an appropriate transformation, we can generate signals for transmission which will be affected by the limitation of transmission in ways that will not show as much in the reproduced picture.

The colour printing world uses another version of the luminance /chrominance system that has many similarities to that used in colour television. That is called the hue-saturation-value (HSV) system or the hue-saturation-intensity (HSI) system. In these systems, value or intensity is the same as luminance—it represents the black and white component of the image, and hue and saturation are the chrominance components. Hue refers to the colour being displayed, and saturation describes how deep that colour is. In a black and white image, saturation is zero (and hue is meaningless), and as the image becomes coloured, saturation values increase. The same terms, hue and saturation, are used with the same meaning in colour television.

In a composite system, the luminance and chrominance are combined by a scheme of frequency interleaving in order to transmit them on a single channel. The luminance signal is transmitted as a normal monochrome signal on the cable or broadcast channel, and then the chrominance information is placed on a high-

frequency subcarrier located near the top of the channel bandwidth. If this carrier frequency is correctly chosen, very little interference will occur between the two signals. This interleaving works because of two facts:

1. The luminance channel is not very sensitive to interfering signals that come in near the high end of the channel bandwidth. This is especially effective if the interfering signal has a frequency that is an odd multiple of half the line scanning rate. In this case, the interfering frequency has the opposite polarity on adjacent scanning lines, and visually the interference tends to cancel out. The selection of carrier frequency for the chrominance ensures this interlace condition.

2. The eye is much less sensitive to colour edges than it is to luminance edges in the Picture. This means that the bandwidth of the chrominance signals can be reduced without much visual loss of resolution. Bandwidth reductions of 2 to 4 for chrominance relative to luminance are appropriate.

Colour Video Formats—NTSC

The NTSC colour TV system is the standard broadcasting system for North America, Japan, and a few other countries. NTSC is an acronym for National Television Systems Committee, a standardizing body which existed in the 1950s to choose a colour TV system for the United States. The NTSC system is a composite luminance/chrominance. An important objective for the NTSC system was that it had to be compatible with the monochrome colour system which was in place with millions of receivers long before colour TV began. This objective was met by making the luminance signal of NTSC be just the same as the previous monochrome standard—existing monochrome receivers see the luminance signal only. Furthermore, the colour signal present at the top of the bandwidth does not

show up very much on monochrome sets because of the same frequency-interleaving that reduces interference between luminance and chrominance.

In NTSC the luminance signal is called the V signal and the two chrominance signals are I and Q. I and Q stand for in-phase and quadrature, because they are two-phase amplitude-modulated on the colour subcarrier signal (one at 0 degrees, and one at 90 degrees—quadrature). The colour carrier frequency is 3.579545 MHz which must be maintained very accurately. The tables at the end of this chapter give the matrix transformation for making Y, I, and Q from RGB. As already explained, the I and Q colour difference signals have reduced bandwidths. While the luminance can utilize the full 5.5-MHz bandwidth of a TV channel, the I bandwidth is only 1.5 MHz, and the Q signal is chosen so that it can get away with only 0.5 MHz bandwidth. (In fact, pretty good results are obtained if both I and Q only use 0.5 MHz—most TV receivers and VCRs in the United States have 0.5-MHz bandwidth in both chrominance channels.)

Colour Video Formats—PAL and SECAM

The PAL and SECAM systems, which originated in Europe are also luminance/chrominance systems. They differ from NTSC primarily in the way in which the chrominance signals are encoded. In PAL, chrominance is also transmitted on a two-phase amplitude-modulated subcarrier at the top of the system bandwidth, but it uses a more complex process called **Phase Alternating Line (PAL),** which allows both of the chrominance signals to have the same bandwidth (1.5 MHz). Because of the different bandwidths, a different set of chrominance components is chosen, called U and V instead of I and Q.

In addition, PAL signals are more tolerant of certain distortions that can occur in transmission paths to affect the quality of the colour reproduction. The tables at the end of this chapter give numbers for the PAL system.

The **SECAM** system (**Sequentiel Couleur avec Memoire**), developed in France, uses an FM-modulated colour subcarrier for the chrominance signals, transmitting one of the colour difference signals on every other line, and the other colour difference signal on alternate lines. Like the PAL system, SECAM is also more tolerant of transmission path distortions.

Video Performance Measurements

Analog television systems cause their own particular kinds of distortion to any signal passing through. Remember, analog systems are never perfect. Analog distortions also accumulate as additional circuits are added, and in a large system all the parts of the system must be of higher quality if the picture quality is to be maintained. A single component intended for a large system has to be so good that its defects become extremely difficult to observe when the component is tested by itself. However, when the component is used repeatedly in cascade in a large system, the accumulation of small distortions becomes significant. Many sophisticated techniques have been developed for performance measurement in analog television systems.

All analog video measurements depend on either looking at images on a picture monitor or making measurements of the video waveform with an oscilloscope or waveform monitor (which is just a special oscilloscope for television measurements). Because monitors also have their own distortions, looking at images on picture monitors tends to be suspect. Image based measurements also involve judgment by the observer and therefore are subjective. On the other hand,

oscilloscopic evaluation of waveforms can be more objective and therefore waveform-based approaches have been developed for measuring most parameters. However, picture monitors are good qualitative tools for performance evaluation, particularly when the characteristics of the monitor being used are well understood and the observer is skilled. Of course, the fundamental need for picture monitors in a TV studio is artistic—there is no other way to determine that the correct scene is being captured with the composition and other features desired by the producer and director.

One measurement on video waveforms that must always be done is the measurement of video levels. The television system is designed to operate optimally when all video signals are kept to a particular amplitude value or level: If signals get too high, serious distortions will occur and display devices may become overloaded. Similarly, if levels are too low, images will be faded out and the effect of noise in the system will become greater. Most video systems are designed for a standard video voltage level, such as 1 volt peak-to-peak, and all level-measuring equipment is calibrated for that level.

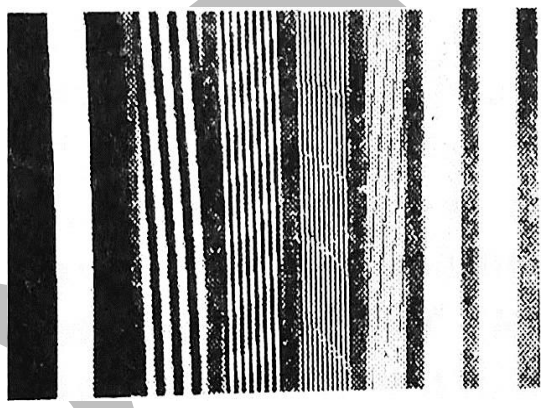
To simplify going between different systems that have different actual voltage standards, most oscilloscopes and other level indicators are calibrated in IRE levels. (IRE is an acronym for Institute of Radio Engineers, one of the forerunners of today's worldwide electrical engineer's professional society, the Institute of Electrical and Electronics Engineers.) This refers to a standard for video waveforms which specifies that blanking level will be defined as 0 IRE units and peak white will be 100 IRE units. Other aspects of specific signals can then be defined in terms of this range.

Most video performance measurements are based on the use of test patterns. These are specialized images which show up one or more aspects of video

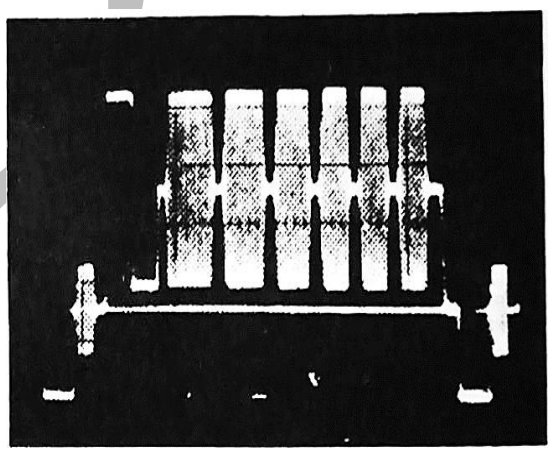
performance. Test patterns may be charts, which are placed in front of a camera, or they may be artificially generated signals, which are introduced into the system after the camera. Because a camera has its own kinds of impairment, a camera usually cannot generate a signal good enough for testing the rest of the system. Therefore, a camera is tested by itself with test charts, and then the rest of the system is tested with theoretically perfect signals, which are electronically generated by test signal generators.

Measurement of Resolution

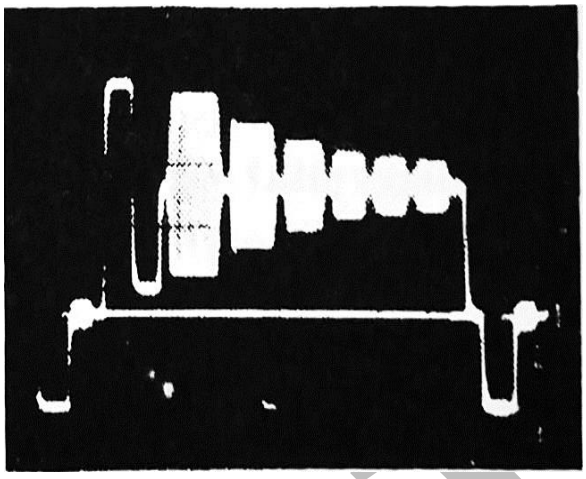
(a) NORMAL



(b) NORMAL



(c) LOSS OF HIGH FREQUENCIES



(d) LOSS OF MID FREQUENCIES

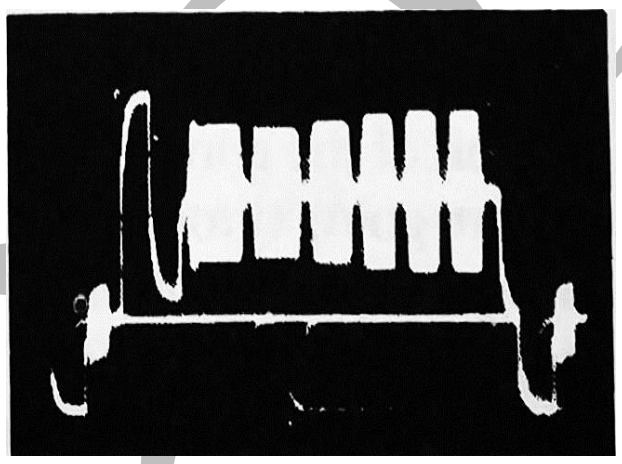


Fig 5.5: Testing frequency response with a multiburst pattern

Let's begin with the characterization of resolution. One test pattern for resolution is the multiburst pattern—it can be either artificially generated or made into a test chart, and it is used to test horizontal resolution of a system.

A multiburst pattern is shown in Figure 5.5. It consists of sets of vertical lines with closer and closer spacing, which give a signal with bursts of higher and higher frequency. Figure 5.5 also shows a line waveform for correct reproduction of a

multiburst pattern and another waveform from a system that has poor high-frequency response. This latter system would cause vertical lines to appear fuzzy in an image.

Another more subtle impairment is shown by the third waveform in Figure 5.5—in this case there is a mid-frequency distortion, which would make images appear smeared.

The multiburst pattern only tests horizontal resolution. To test vertical resolution as well, a resolution wedge test pattern is used. Figure 5.6 shows some resolution wedge patterns for testing both horizontal and vertical resolution.

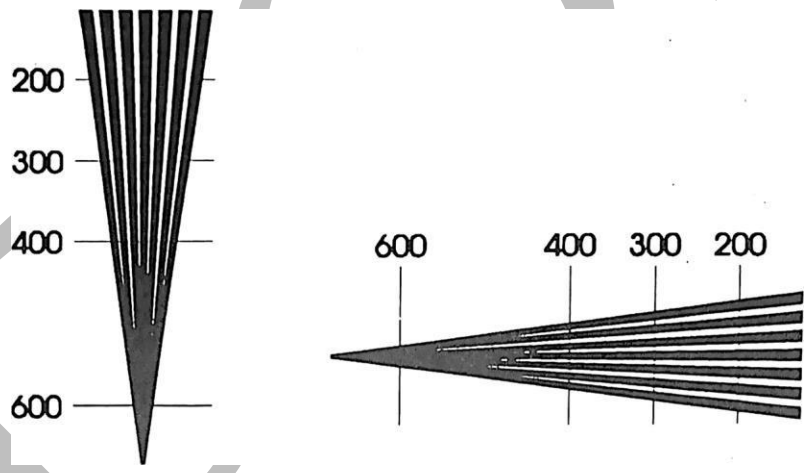


Figure 5.6: Resolution wedge patterns

A resolution wedge is used by observing where the lines fade out as the wedge lines become closer together. The fadeout line number can be estimated from the numbers beside the wedge—this would be the resolution performance of the system or camera being tested.

A common resolution test chart is the EIA Resolution Test Chart, designed to test many parameters in addition to resolution. It is usually placed in front of a camera. (EIA stands for Electronic Industries Association, an industry group in the United States which is very active in standards for television in that country.) There are wedges for both horizontal and vertical resolution at different locations in the image and various other blocks and circles to test camera geometric distortion (linearity) and gray scale response.

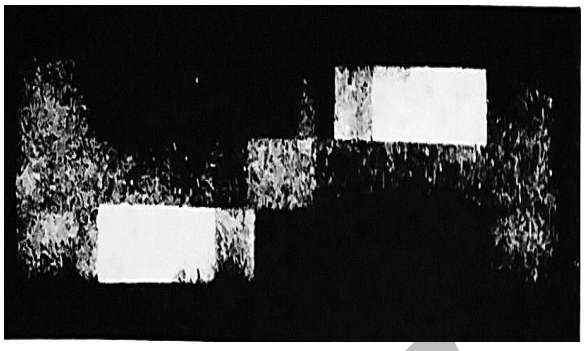
Measurement of Gray Scale Response

The gray scale test has its own special pattern, called the stairstep. This may be either a camera chart or an electronic generator. Figure 5.7 shows one version of the gray scale chart, with examples of the signals created by the pattern going through various systems.

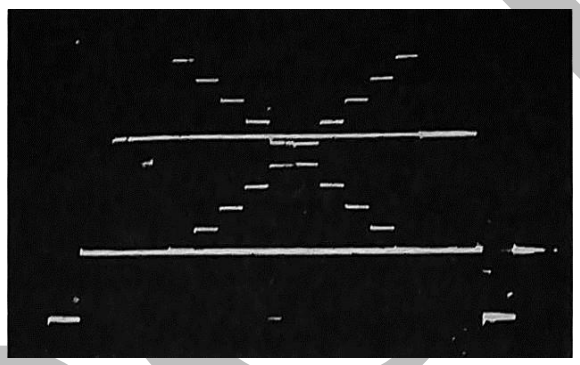
An important consideration regarding gray scale reproduction in TV systems is that an equal-intensity-step gray scale test chart should not produce an equal-step electrical signal. The reason for this is that the electrical signal will ultimately drive a CRT display and the brightness versus voltage characteristic of a CRT is not linear. Therefore, television cameras include gamma correction to modify the voltage transfer characteristic of the signal to compensate for an average CRT's brightness versus voltage behaviour. This is written into most television standards because the CRT was the only type of display that existed at the time of standardization.

New-type displays such as LCDs may have different gamma characteristics for which they must include their own correction if they are to properly display standard television signals.

(a)



(b)



(c)

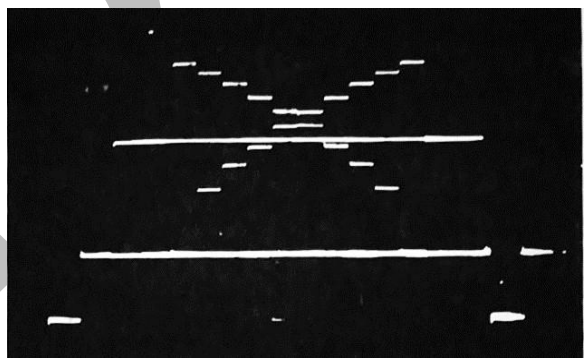


Figure 5.7: (a) Gray scale test chart, (b) Normal Waveform, (c) Distorted Waveform showing black stretch

A typical CRT has a voltage-to-intensity characteristic close to a power of 2.2. To correct for this CRT gamma characteristic, a signal from a camera sensor having a linear characteristic (gamma of 1.0, which is typical for sensors) must undergo gamma correction by a circuit which takes the 2.2 Foot of its input signal. If gamma correction is left out of a system, the effect is that detail in the black regions of the picture will be lost, and pictures appear to have too much contrast regardless of how you adjust the display. In dealing with computer-generated images, it is important to provide for the gamma characteristic to create realistic looking pictures on CRT displays.

Measurement of Noise

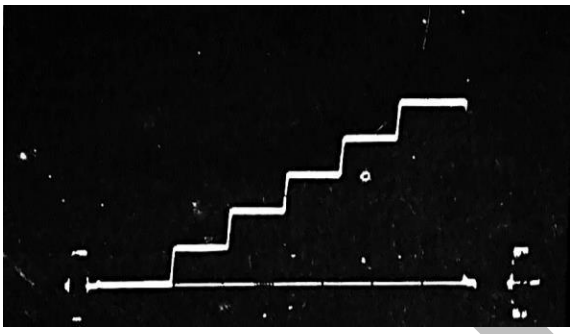
Measurement of noise in a television system is an art in itself. Most specifications are in terms of signal-to-noise ratio (SIN), which is defined as the ratio between the peak-to-peak black-to-white signal and the rms value (rms means root-mean-square—a kind of averaging) of any superimposed noise.

S/N numbers are commonly given in decibels (dB), which are logarithmic units specifically designed for expressing ratios. The bel represents a power ratio of 10:1, the decibel is one-tenth of that. Since signal-to-noise ratios are usually voltage ratios, not power, a 10:1 signal-to-noise ratio is 20 decibels because the power ratio goes as the square of voltage ratio. Because the decibel is logarithmic, doubling the dB value is the same as squaring the ratio-thus, 40 dB would be a signal-to-noise ratio of 100:1.

A good SIN ratio for a system is around 200:1 (46 dB), which means that the rms noise in that system is 200 times less than the maximum black-to-white video level the system is designed to handle. Note that the measurement of rms noise requires an integrating kind of meter, and for this purpose there needs to be a region of the

image for measurement that does not have any other signal present. There are many different kinds of test patterns for noise measurements.

(a)



(b)

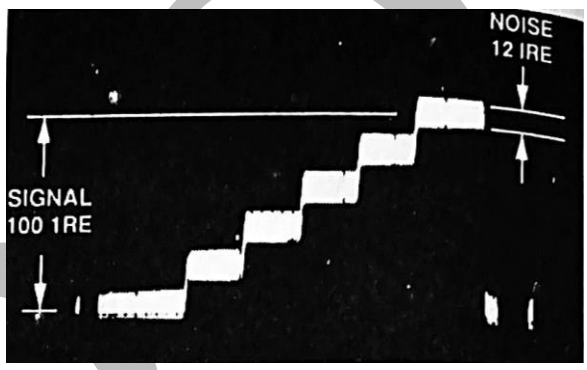


Figure 5.8: Noise measurement using an oscilloscope:

(a) Signal without noise

(b) Signal with noise. S/N ration = 600/12 = 50:1 =34dB

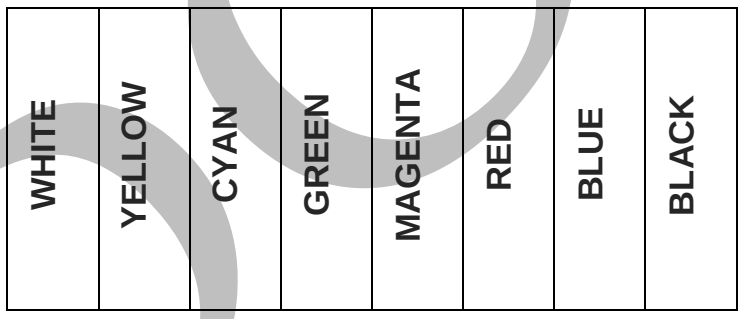
It is also possible to get an approximate S/N measurement by looking at video signals with a waveform monitor or oscilloscope, as shown by Figure 5.8. Noise appears on a video waveform as a fluctuating fine grain fuzz, which is usually evident on all parts of the active picture area of the waveform. (Noise usually does not appear during the blanking intervals because video equipment often regenerates the blanking interval, replacing sync and blanking—which may have gotten noisy in transmission or recording—with clean signals.) The peak-to-peak value of the noise fuzz can be estimated as a percentage of the total video black-

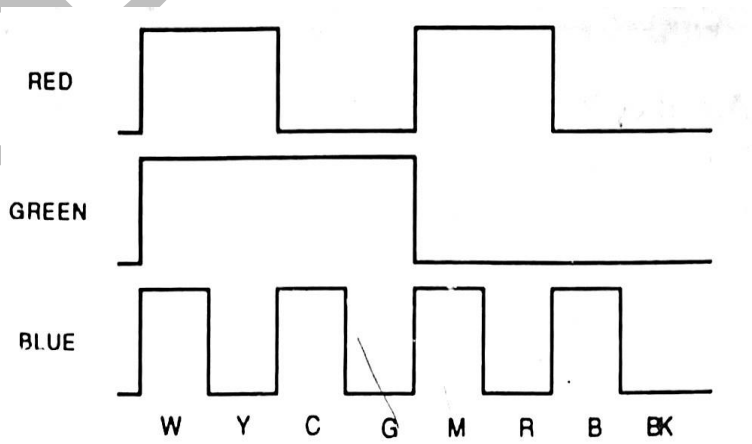
to-white range. Then an approximate S/N ratio may be calculated by dividing 600 by the percentage of noise fuzz. This is based on the assumption that typical noise has an rms value six times less than its peak-to-peak value. Therefore, a 46-dB system would show about 3 percent peak-to-peak noise fuzz on its signals.

Measurement of Colour Performance

Another class of measurement is involved with the colour performance of a system. Measurement of colour rendition of cameras is beyond our scope here, but there are simple means to test the NTSC parts of the system. This is most often done with an electronically generated signal called a colour bar test pattern.

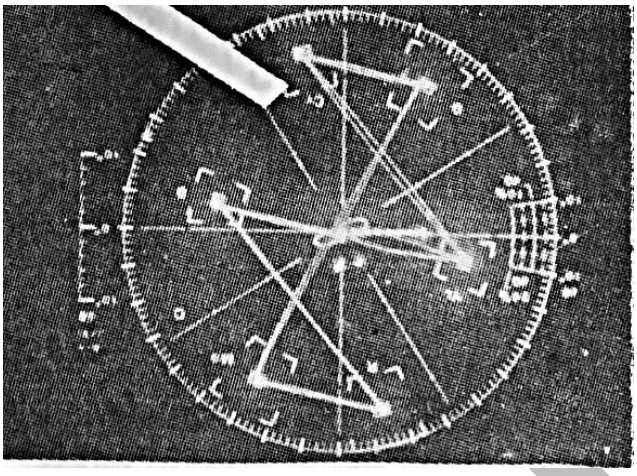
(a) SCREEN DISPLAY





(b) RGB VIDEO WAVEFORM

(c)



(d)

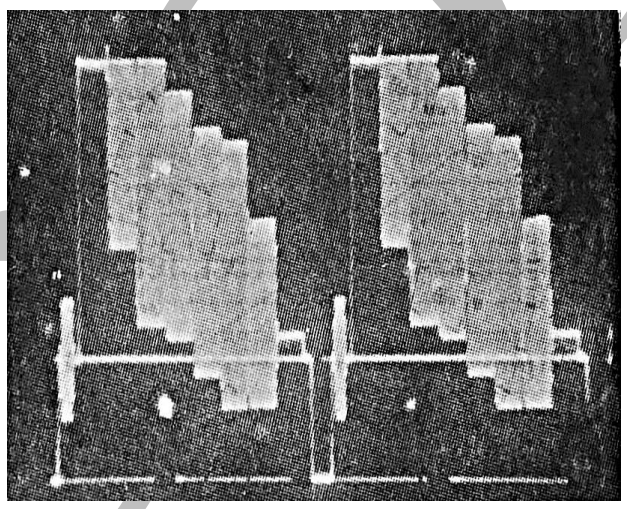


Figure 5.9: (a) Colour Test Pattern, (b) RGB Video Waveforms (c) NTSC Video Waveform, (d) NTSC vectorscope display

Figure 5.9 shows a colour bar pattern and the resulting waveform display used widely in NTSC systems. The particular bar sequence chosen arranges the colours in order of decreasing luminance values, which makes an easy-to-remember waveform. The colour bar signal can also be examined with a special oscilloscope made for studying the chrominance components - this instrument is

called a vectorscope. Figure 5.9 also shows what an NTSC colour bar signal looks like on a vectorscope. There are also split-field versions of the colour bar pattern, which put the bar pattern at the top of the screen and a different pattern (such as a stairstep pattern) at the bottom. With this arrangement, several different tests can be made using only one test signal. Available test signal generators provide many combinations of the test signals, to be used individually or simultaneously in split-field combinations.

Colour distortions in NTSC systems can occur as the result of amplitude shifts or phase shifts in the chrominance components of the signal. Amplitude changes the saturation (intensity) of the colour, and phase shifts change the hue (tint) of the colours. Both of these distortions can be observed by the proper use of a vectorscope.

There are many other kinds of video performance measurements. For our purpose in understanding the interfaces between analog and digital video, the few that are covered in the foregoing paragraphs are sufficient.

Analog Video Artifacts

Archaeologists use the word artifact to refer to an unnatural object which has been found in nature—thus it is presumably manmade. In the video world, artifacts are unnatural things which may appear in reproduction of a natural image by an electronic system.

In order to appreciate the vagaries of analog video, and particularly to appreciate the different things which an analog system does to video compared to what a digital system does, we need to give attention to these analog image distortions - artifacts. A skilled observer of video images can often recognize things that the

casual Viewer will never notice - but being able to recognize them is important when you are responsible for the system. Often the artifacts are clues to something that is starting to go wrong, recognizable before it becomes catastrophic. So, at the risk of destroying your future enjoyment of television images that may be less than perfect, let's look at analog video artifacts.

Noise

The most common form of noise is what we refer to on our television receiver as snow. The speckled appearance of snow is caused by excessive amounts of noise rather uniformly distributed over the bandwidth of the video signal, which is often called white noise or flat noise. Flat noise is readily observable when the signal-to-noise ratio falls below about 40 dB. A good three-sensor camera will generate a signal where the S/N is nearer to 50 dB. However, this signal may be subsequently degraded by various transmission paths or by video recording (see later discussion on recorders).

There are other types of noise which look quite different. For example, noise that is predominantly low-frequency (in the vicinity of the line frequency or lower) will appear as random horizontal streaks in the picture. The eye is quite sensitive to this kind of noise, but fortunately it is unusual in properly operating systems. When this kind of noise is seen, something in the system may have become intermittent or is about to fail catastrophically.

Another noise phenomenon that looks different from snow is color noise. This is noise in the transmission path for a composite signal which appears in a frequency band close to the color subcarrier. Because of the relatively narrow bandwidth of the chrominance channels, color noise appears as moderately large streaks of varying color. Color noise is primarily an artifact of video recording.

RF Interference

Various kinds of coherent (not random) interferences can creep into video signals from other sources. The appearance of these interferences will depend on the exact frequency relationship they have to the scanning frequencies and to the colour subcarrier. The degree to which they produce moving patterns depends on whether (or how closely) they are synchronized with the main signal. You have probably seen the interference which consists of two vertical bars spaced about 10 percent of the picture width which move slowly across the picture, taking several seconds to go all the way across. This is interference from another colour television signal on the same standard but not synchronized with the main signal. It is quite common in signals received over the air, and it also may be a problem in a television studio system that has several sources of signals that are not synchronized.

Interference from single-frequency non-television signals will produce diagonal- or vertical-line patterns, either stationary or moving. The relationship to the horizontal scanning frequency controls the exact pattern produced. Interference from multi-frequency sources will produce more complex patterns. For example, interference getting into video from an audio signal will produce a pattern of horizontal bars which changes in size and position with the sound. The relationship is obvious if you are able to hear the sound while you watch the patterns in the video.

Interference from coherent sources is much more visible than is random interference or noise. This is because the coherent interference creates some kind of pattern, which repeats over and over in the same (or a slowly moving) location. Patterns of bars may have any spacing and may range from vertical bar patterns through diagonal patterns to horizontal bar patterns. Coherent interferences are

often visible if they exceed about 0.5 percent peak-to-peak relative to the black-to-white video range.

Loss of High Frequencies

In a composite color television system, the first effect of loss of high frequencies is that the color saturation (vividness of color) will be reduced, or color may be lost entirely. (Except in the SECAM system: In SECAM the FM nature of the color subcarrier will retain color saturation. The effect becomes one of increasing color noise, or finally loss of color.) More severe loss of high frequencies will noticeably affect the sharpness of vertical edges in the image. In an RGB system, loss of high frequency will only affect sharpness, although if the loss is not the same in all three channels, it will also show as color fringes on vertical edges.

Smear

Smear shows up as picture information which is smeared to the right (usually). Figure 5.10 shows an image with smear. It is caused by a loss of amplitude or a phase shift at frequencies near or somewhat above the horizontal line frequency. Many tube-type cameras have a high-peaker adjustment, which can cause this kind of distortion if not properly set. It is also caused by long video cables which are not properly equalized.

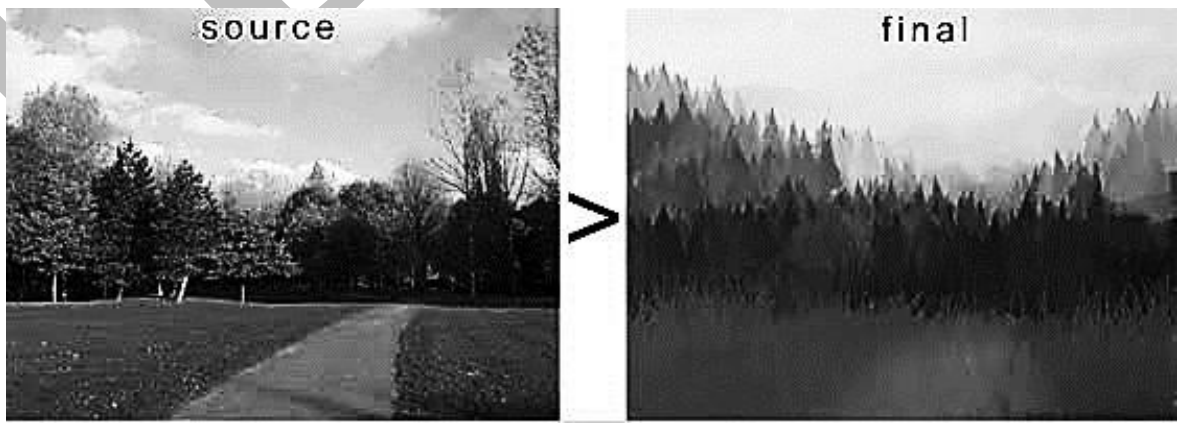


Figure 5.10 Image showing the smear artefact

Streaking

Streaking is present when a bright object in the image causes a shifting of brightness all the way across the image at the vertical location of the bright object. Figure 5.11 shows what it looks like. Streaking is usually caused by video information which gets into the blanking interval and then interferes with the level-setting circuits (called clamps) present in many pieces of video equipment. It may occur when a signal with an extreme case of smear passes through equipment containing clamps, or it can happen due to loss of frequency response at frequencies below the line frequency. The latter is usually caused by a component failure somewhere in the system.

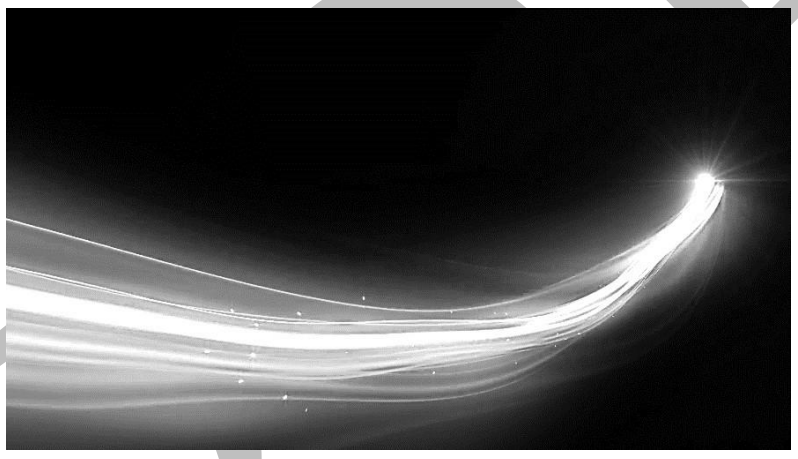


Figure 5.11: Image showing analog streaking artifact.

Color Fringing

Color fringing is present when edges in the picture have colors on them that were not in the original scene. This may be over all of the picture, or it may be confined to only one part of the picture. The principal cause of color fringing is registration errors in cameras. However, there are some less frequent kinds of distortions in the frequency domain or in recording systems which will cause similar effects. In these cases the entire picture is usually affected, and it will most likely occur only on vertical edges.

Color Balance Errors

In an RGB system, the most likely color errors are those caused by the video levels not being the same in the three channels. This is a color balance error, and it shows up as a constant color over the entire image. It is independent of what colors are in the scene. For example, if the red level is too low, white areas in the image will have a minus red cast—that is, they will appear cyan. All parts of the image will have cyan added to the correct color. Correct reproduction of white areas is a good test for color balance. In a composite system, color balance errors usually originate at the camera, where they may be caused by balance errors in the RGB circuits of the camera or errors in the composite encoder built into the camera. In NTSC and PAL systems, the subcarrier output goes to zero in white areas of the picture, so looking for zero subcarrier on a white bar is the test for proper balance of an NTSC or PAL signal. There are more subtle color balance errors where the color balance varies as the brightness level of the image changes. This is tested by observing the color balance on all the steps of a gray scale test pattern or test signal. The most common source of this kind of error is the camera.

Hue Errors

In RGB systems, hue errors are unusual if the color balance is correct. However, in composite systems, particularly NTSC, hue errors are caused by improper decoding of the color subcarrier; in particular, the phase of the color subcarrier controls hue in an NTSC system. NTSC television receivers have a hue control which adjusts this parameter. In a color origination system, however, the signals must all be set to a standard, so that the viewer will not feel that the hue control has to be readjusted every time a different camera is used. In the PAL system, the phase-alternating line-encoding approach makes the hue much less sensitive to the phase of the color subcarrier, and PAL receivers usually do not have a hue control.

Color Saturation Errors

It has already been explained that in composite systems, color saturation errors are most commonly caused by incorrect high-frequency response, which leads to incorrect color subcarrier amplitude.

Flag-Waving

In low-cost video recorders, there can be a problem of synchronization instability at the top of the picture called flag-waving. It shows up as vertical edges at the top of the screen moving left to right from their correct position. In the recorder it is caused when the video playback head of the recorder has to leave the tape at one edge and come back onto the tape at the other edge (which happens during the vertical blanking interval, so this it is not seen).

If the tape tension is not correctly adjusted, flag-waving may appear. Some video recorders have a control called skew, which is an adjustment to minimize this effect.

Jitter

If the entire picture shows random motion from left to right (called jitter), the motion usually is caused by time-base errors from video recorders. In a video recorder the smoothness of the mechanical motion of the recorder's head drum is very critical in order to reproduce a stable picture.

Higher-priced recorders contain a time base corrector (TBC) to correct this problem but TBCs are expensive, and it costs less in a low-priced recorder to try to make the mechanical motion stable enough without a TBC. Occasionally jitter effects will be caused by other kinds of defects or interferences getting into the synchronizing signals or circuits.

Video Equipment

The video business is a mature, fully developed industry around the world. There are many manufacturers selling competitively to worldwide markets and because there are good standards for video signals and broadcasting, there is a wide range of equipment made for every imaginable purpose in video production, recording, postproduction, broadcasting, and distribution.

The markets for video equipment range from the most sophisticated network or national broadcasting companies through industrial and educational markets, to the home (consumer) market. In the discussion that follows, we will simplify the market structure to just three categories:

1. **Broadcast**: Equipment used by large-market TV broadcast stations and networks. This equipment has the highest performance, is intended for large system application, and is the most expensive.

2. **Professional**: Equipment for use in educational and industrial applications (and smaller broadcasters) who cannot afford full broadcast-level equipment but still need a lot of performance and features.

3. **Consumer**: Home equipment where price is the first consideration. Must have simple operation and high reliability for use by a nontechnical user. The intended system application is very simple—usually one-camera, one-recorder systems.

Live-Pickup Color Cameras

Broadcast-level video cameras for live pickup are generally of the three-sensor variety. The highest performance cameras are large units for studio use, and they

have large-format sensors. They also support a very wide range of lenses for extreme zoom range, wide angle, very long telephoto, etc. These cameras can also be taken in the field when the highest possible performance is needed, but their large size hampers portability.

There are smaller broadcast cameras, which are designed to be truly portable, and no expense is spared to keep the performance as high as possible. Broadcast cameras today generally contain computers which control their setup adjustments and many features of operation. However, the signal processing is usually analog.

Professional-level video cameras are also of the three-sensor type, but they typically use smaller sensors and smaller optics to reduce the cost. They usually have a simpler system design with fewer special features; therefore, they are lower in cost and easier to operate and maintain. The compromises in picture quality and flexibility resulting from these changes have been chosen to be acceptable to the markets for this class of equipment. Applications are in education, training, and institutional uses.

Consumer-level video cameras go to the ultimate in cost reduction. Thy use the single-sensor format, which is a compromise in resolution performance compared to the other cameras, but it yields a small, reliable, low-cost unit. Consumer cameras are designed for high-volume production and cost do not have a lot of special features. Very simple system application is in. tended. However, their performance and features have been good enough to create a mass consumer market, and they are an outstanding value in terms of what you get for the price. In the consumer market, you can no longer find video cameras by themselves - they are all combined with video recorders as camcorders. This is a great convenience and also a cost saving, so camcorders have replaced separate cameras. Camcorders are also available in the broadcast and professional

categories; however, they have not completely replaced rate equipment in these fields, primarily because broadcast and professional users often will use several cameras with a single recorder, or maybe with no recorder at all if they are broadcasting live. Therefore, there will always be separate cameras for these uses.

Color Cameras for Pickup from Film

Cameras specifically designed for television pickup from motion picture film or slides are called telecine cameras. Getting good television pictures from film is not as simple as it sounds because of two key problems.

The first is that the frame rate for film is usually 24 frames per second, whereas television frame rates are 25 or 30 Hz. In the parts of the world where television frame rate is 25 Hz, film is shown on television simply by speeding the film frame rate up to match the television frame rate - a 4 percent increase. This amount of speedup is usually acceptable. However, for 30-Hz television systems, the 20 percent speedup required would be unacceptable. In these systems, an approach called 3:2 pulldown is used to resolve the different frame rates. In the 3:2 approach, one film frame is scanned for three television fields and the next film frame is scanned for only two television fields; that is, two film frames are shown in five television fields, which is 2.5 television frames. This ratio of 2.5:2 is exactly the same as the 30:24 frequency ratio, so the average film frame rate can be the correct value.

There are artifacts from 3:2 pulldown, such as a certain jerkiness in motion areas of the image—most television viewers in North America have become used to this effect and accept it. Another 3:2 pulldown artifact is the wheels appearing to turn backwards on a car or wagon. When you see this effect' you can be sure that the program or commercial was originally shot on film. The second film-television

problem is a mismatch between film and television with regard to color reproduction capability. This arises because television is an additive color system, whereas film is a subtractive color system. Film images typically have more contrast than a television system can handle, and film shows its best colors in dark parts of the image (where the dye concentrations are the highest).

Television gives its best colors in bright regions where the CRT is turned on fully and can overcome the effect of stray ambient light on the tube face. Both of these effects may be mostly overcome by using excess gamma correction (much more than is needed to correct for the CRT characteristic) to bring up the dark areas of the image from film. Telecine cameras have elaborate gamma circuits to provide this feature. In addition, since high gamma correction tends to also bring up sensor noise effects, telecine cameras need to start with a higher signal-tonoise ratio from the sensors in order to withstand the high gamma correction.

There are other problems of color rendition in reproducing film, which lead to a need in telecine cameras for much more flexible color adjustment circuits as well. It is common for telecine systems to contain very elaborate color correctors to deal with faded film, incorrect color balance, and other kinds of color errors. Because of the complexities of television from film, there is not much telecine equipment on the market outside of the broadcast field. Even in the broadcast field, telecine has become a specialized capability - used only for film-to-tape transfer. That is because videotape is much easier than film to deal with in a broadcast operation. Most film you see on television was transferred to tape some time before it is broadcast, often immediately after the film was processed.

Video Recording Equipment

For our use in digital video systems we will almost always be dealing with recorded video as the input to the digital system. Video will be shot with analog cameras,

recorded, and often processed extensively before it is digitized. A whole industry, referred to as video postproduction, has grown up to take recorded video material and put it together into finished programs. Techniques and facilities for postproduction are highly developed and are serving large markets for television and other video production. We can expect that for some time these approaches will be the best way to create video for any use, including digital video.

Analog video recording is mostly based on magnetic technology. (An exception is the laser videodisc, which is optical.) Magnetic recording is not a good medium for use directly in analog recording, because it is highly nonlinear. Magnetic recording is in fact better as a digital medium, where a domain of magnetic material can be considered either magnetized or not magnetized. In video recording systems, this is dealt with by modulating the video signal onto an FM carrier before recording. The FM carrier is very well matched to the characteristics of magnetic recording, because it does not require sensitivity to different levels of magnetization; rather, it depends on the size and location of magnetized regions.

One of the considerations of video recording for producing a program is that creation of the finished program requires going through the recording system several times. Original video is shot by using a camera with one recorder to get each of the scenes separately. A SMPTE time code signal is
also recorded with all the material for use in controlling later processes. Recording scenes one at a time is done because it is much more efficient from the staging and talent viewpoint to not try to put scenes together in real time. This process of capturing the scenes one at a time is called production.

To put together a program involving scenes from several cameras and often several locations, all the original tapes will be taken to a postproduction studio where the desired shots will be selected for assembly into the final program. Time

code locations of all critical points will be tabulated. Then each of the scenes is run from its original tape under time code control and re-recorded in the proper sequence on a new tape to create an edited master. In that process various transition effects between scenes can also be introduced, such as dissolves, fades, wipes, etc. The edited master is usually backed up by making another edited master (called a protection master) or, more commonly, by re-recording copies from a single edited master (called a protection copy). Re-recording of videotape is called dubbing, and the resulting tape is called a dub.

If the dub from the edited master is the copy we use to digitize, you can see that we have gone through the analog recording process at least three times—once in original production, again in making the edited master, and a third time to back up the edited master. This is referred to as three generations. Since analog distortions will accumulate, a recording system that will deliver good pictures after three generations must have considerably higher performance than a recorder we will only go through once. In fact, three generations is almost a minimum number - there are often additional steps of video postproduction which can lead to needing five or six generations before the final copy is made.

Most video recorders are designed to record the composite video signal—NTSC, PAL, or SECAM. However, there are several new systems that use an approach called component recording. In a component recorder, the signal is recorded in two parallel channels, usually with luminance on one channel and chrominance on the other channel. Some of the recording artifacts can be reduced by this approach.

If a component recorder is used with a composite camera, the composite signal must be decoded at the input of the recorder to create the component format. In this case, there is little performance advantage for a single generation because

both component and composite signal degradations will be present. However, if the component recorder is either combined with a camera or used with a camera having component outputs, then the composite encoding does not have to occur until after recording, and the system performance can be improved.

There are even some attempts to build component postproduction facilities where composite encoding does not occur until after postproduction. Such a facility is able to go through more generations and therefore can perform fancier postproduction effects is because of the need for multiple generations in video recording, there a move in the television industry to develop digital video recorders.

A digital recorder and a digital postproduction system could use unlimited generations, just as we are used to doing with computer recording devices. (In a computer we never worry about repeated loading and saving of data because we have confidence that the digital system will make no errors.) There are digital video recorders coming out in the broadcast field in both composite and component formats. The composite recorders use the same analog formats we have been talking about—they digitize the analog composite signal at their input. However, the digital component recorder takes a digitized YUV input format.

The standard of broadcast-level analog composite video recorders is the Type C recorder, using one-inch videotape in a reel-to-reel format. This equipment is the workhorse of broadcast television around the world and delivers the best analog recording performance available today. The basic Type C machine is a large unit weighing somewhat more than 100 pounds and intended for fixed or transportable use. Type C recorders will deliver good performance after three generations and are usable up to five or six generations. Another broadcast-level format, which is somewhat less used in the United States but is found extensively in Europe is the Type B system. The Type B recorders also use one-inch tape, but their format is

different in a way which allows smaller machines to be built. Type B performance is equivalent to Type C.

The two broadcast-level component recording formats, mentioned earlier, are the Betacam format and the Type Mll format. Both of these use half-inch tape in a cassette and have performance that is close to Type C level. Because of the small tape size, very compact machines can be built, including a camcorder format, which combines camera and VCR in one hand-held unit.

In professional-level recording, the three-quarter-inch U-Matic format is the workhorse. This format uses a cassette with three-quarter-inch tape, and machines come in rack-mounted, tabletop, and portable configurations. The three-quarter-inch system uses a different way of getting the composite signal onto the tape, which requires that the luminance and chrominance be taken apart and then put back together inside the recorder. Doing that to a composite signal introduces some inherent degradations so that the picture quality of the three-quarter-inch system is not as good as Type C, particularly with respect to color sharpness and luminance bandwidth. The three-quarter-inch system typically can go only two generations with acceptable pictures.

Recently, two other formats for the professional market have been introduced. These are the S-VHS and the Hi-8 formats, derived from the VHS and 8-mm consumer formats. These are component formats whose perforrmance is highly competitive with U-Matic, the equipment is smaller, and they will probably take over the market in the future.

In the consumer-level recording field we have the familiar VHS and 8-mm formats. These systems all use a method of recording similar to the three-quarter-inch systems, in which separating of luminance and chrominance is required inside the

recorder. However, because of the smaller tape sizes and lower tape speeds, bandwidths are much lower, and the pictures are noticeably impaired by the recorder. However, they have proven themselves to be good enough for consumer entertainment use—witness the proliferation of consumer VCRs around the world.

As a medium for input to a digital system, however, the consumer equipment is not very satisfactory. The reason is that these VCRs introduce artifacts that are different from the digital artifacts, and, therefore, when their signal is digitized by a low-cost digital system, both kinds of artifacts can appear. That is usually just too much. Note, however, that broadcast- and professional-level component systems (Betacam, Type Mll, S-VHS, or Hi-8) are based on the same consumer half-inch or 8-mm tape technologies, but the component recording technique allows much better performance—at a higher price, of courses The component systems generally are quite satisfactory for recording source material for digital systems.

Video Monitoring Equipment

Monitoring equipment for analog television includes picture monitors and waveform monitors, which also come in various price/performance levels. Broadcast-level video monitors cost several thousand dollars and come as close as possible to being transparent, which means that the picture you see depends on the signal and not the monitor. Broadcast monitors will also include various display modes, which allow different aspects of the signal (in addition to the picture content) to be observed. One common feature is the pulse-cross display, which shows the synchronizing signal part of a composite video signal.

Broadcast monitors are designed to be capable of being matched, so that a group of monitors will show the same signal the same way. Matching is important when several monitors are going to be used to set up signals that may eventually be

combined into the same program. In such a case, it is important that the color reproduction of al l signals match as closely as possible. Broadcast monitors always have inputs for composite signals. Some monitors also have RGB inputs, but this is unusual, because RGB signals seldom exist today in broadcast studios or postproduction facilities.

Professional-level monitoring equipment is designed to a slightly lower price/performance point - pictures are still very good but some features may be sacrificed in the interest of price.

Consumer-level monitoring is mostly done with the ubiquitous television receiver. All TV receivers have an antenna input for receiving the composite signal in RF form as it is broadcast on a TV channel. Recently' TV sets are also adding video inputs for a baseband composite signal, which may come from a consumer camera, VCR, or home computer. (A baseband composite signal is a video signal that has not been modulated up to a TV channel.) Rarer is the RGB input, although this will appear in more televisions as the use of computers grows in the home.

Broadcasting of the composite video signal also introduces some characteristic performance problems, which we sometimes observe on our TV sets. In a wired system, it is unusual to get extremely noisy (snowy) signals or signals containing ghost images; however, these are common defects in over-the-air transmission. TV receivers are designed to deal as well as possible with these problems and still make an entertaining picture. Because broadcasting in a fixed channel bandwidth puts an absolute limit on the bandwidth for the video signal, pictures received on a TV set will not look as good as they do in the studio. Also, there is a lot of competition in TV receiver manufacturing, and they are available at several points on the price/performance curve.

Worldwide Television Standards

This section presents a summary of the major parameters of the principal television systems in the world—NTSC, PAL, and SECAM. There are minor differences between the same systems implemented in different countries.

The information given here is for NTSC-M (United States), PAL-B (West Germany), and SECAM-L (France).

Scanning Parameters

Scanning	NTSC	PAL	SECAM
Lines/frame	525	625	625
Frames/second	30	25	25
Interlace ratio	2:1	2:1	2:1
Aspect ratio	4:3	4:3	4:3
Color subcar. (Hz)	3,579,545	4,433,619	Note 1
sc/h ratio	455/2	1135/4	Note 1

Sync Waveforms: Horizontal Timing

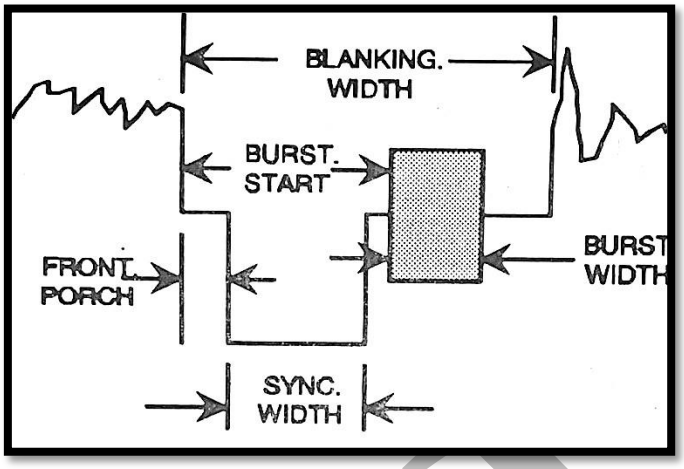
Refer to Figure 5.12 for nomenclature. All values given are nominal. Horizontal sync values are in microseconds.

Horizontal Timing

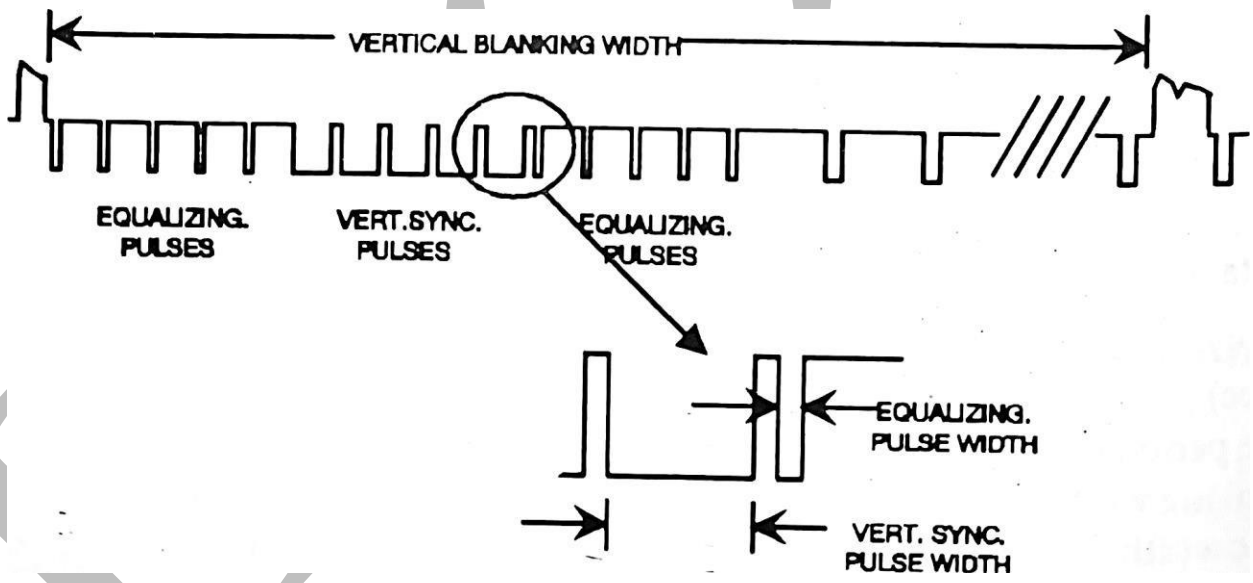
(micro-sec)	NTSC	PAL	SECAM
Line period (H)	63.55	64.0	64.0
Blanking width	10.9	12.0	12.0
Sync width	4.7	4.7	4.7
Front porch	1.2	1.2	1.2
Burst start	5.1	5.6	Note 2
Burst width	2.67	2.25	Note 2
Equalising width	2.3	2.35	2.35
Vert. sync width	27.1	27.3	27.3

Sync Waveforms: Vertical Timing

Vertical Sync	NTSC	PAL	SECAM
Blanking width	20H	25H	25H
Num. equalizing	6	5	5
Num. vert. sync	6	5	5
Burst suppression	9H	Note 3	Note 4



(a) Horizontal blanking interval



(b) Vertical blanking interval

Figure 5.12 Sync waveform nomenclature

NTSC

Color Matrix Equations	**Chrominance Bandwidth (-3 dB)**
Y = 0.30 R + 0.59 G + 0.11 B	
I = 0.60 R - 0.28 G - 0.32 B	1.3 MHz
Q = 0.21 R -0.52 G +0.31 B	0.45 MHz

PAL

Color Matrix Equations	**Chrominance Bandwidth (-3 dB)**
Y = 0.3 R + 0.59 G + 0.11 B	
U = 0.62 R – 0.52 G – 0.10 B	1.3 MHZ
V = -0.15 R – 0.29 G + 0.44 B	1.3 MHz

SECAM

Color Matrix Equations	**Chrominance Bandwidth (-3 dB)**
Y = 0.30 R + 0.59 G + 0.11 B	
DR = -1.33 R + 1.11 G +0.22 B	1.3 MHz
DB = -0.45 R -0.88 G + 1.33 B	1.3 MHz

Sync Waveform Nomenclature Color

Modulation Parameters

NTSC

The chrominance subcarrier is suppressed-carrier amplitude modulated by the I and Q components, with the I component modulating the subcarrier at an angle of 0 degrees, and the Q component at a subcarrier phase angle of 90 degrees. The reference burst is at an angle of 57 degrees with respect to the I carrier.

PAL

The chrominance subcarrier is suppressed-carrier amplitude modulated by the U and V components, with the U component modulating the subcarrier at an angle of 0 degrees, and the V component at a subcarrier phase angle of 90 degrees. The V component is alternated 180 degrees on a line-by-line basis. The reference burst also alternates on a line-by-line basis between an angle of +135 degrees and -135 degrees relative to the U carrier.

SECAM

The chrominance subcarrier is frequency modulated by the D_R and D_B signals on alternate lines. At the same time, the subcarrier frequency changes on alternate lines between SC_R and SC_B. The color burst also alternates between the two frequencies.

Notes

Note 1: SECAM uses two FM-modulated color subcarriers transmitted on alternate horizontal lines. SC_R is 4.406250 MHz ($282*f_H$), and SC_B is 4.250000 MHz ($272f_H$).

Note 2: SECAM places a burst of SC_R or SC_B on alternate horizontal back porches, according to the subcarrier being used on the following line.

Note 3: Because the PAL burst alternates by 90 degrees from one line to the next, the 8.5-line burst suppression during vertical sync must be shifted according to a four-field sequence (called meandering) in order to ensure that each field begins with the burst at the same phase.

Note 4: SECAM color burst is suppressed during the entire vertical blanking interval. However, a different burst (called the bottle signal) is inserted on the 9 lines after vertical sync.

Summary

- Conversion of the two-dimensional image into a one-dimensional electrical signal is accomplished by *scanning* that image in an orderly pattern called a *raster*.

- The varying electrical signal from the sensor then represents the image as a series of values spread out in time—this is called a *video signal*.

- Resolution is the ability of a television system to reproduce fine detail in the scene. It is expressed separately for *horizontal* and *vertical* directions.

- All composite formats make use of the luminance/chrominance principle for their basic structure

- The colour printing world uses another version of the luminance /chrominance system that has many similarities to that used in colour television. That is called the hue-saturation-value (HSV) system or the hue-saturation-intensity (HSI) system

- The **SECAM** system (**Sequentiel Couleur avec Memoire**), developed in France, uses an FM-modulated colour subcarrier for the chrominance signals, transmitting one of the colour difference signals on every other line, and the other colour difference signal on alternate lines

- The multiburst pattern only tests horizontal resolution. To test vertical resolution as well, a resolution wedge test pattern is used.

- Most specifications are in terms of signal-to-noise ratio (SIN), which is defined as the ratio between the peak-to-peak black-to-white signal and the rms value (rms means root-mean-square)

- The colour bar signal can also be examined with a special oscilloscope made for studying the chrominance components - this instrument is called a vectorscope

- Flat noise is readily observable when the signal-to-noise ratio falls below about 40 dB

- Color fringing is present when edges in the picture have colors on them that were not in the original scene

- If the entire picture shows random motion from left to right (called jitter), the motion usually is caused by time-base errors from video recorders.

- Most video recorders are designed to record the composite video signal—NTSC, PAL, or SECAM

- The two broadcast-level component recording formats, mentioned earlier, are the Betacam format and the Type Mll format

Unit End Exercises

1. Write a note on Video Resolution and its types.
2. Explain Color Mixing fundamentals.
3. Explain NTSC, PAL and SECAM Video Formats.
4. How to measure resolution, response, noise and color performance?
5. Write a short note on video equipment.

Additional Reference

1. MULTIMEDIA SYSTEMS, John F. Koegel Buford, University of Massachusetts Lowell, Pearson, Fourteenth Impression

6

Digital Video and Image Compression

Unit Structure

6.1 Objectives
6.2 Introduction
6.3 Video Compression Techniques
6.4 Standardization of Algorithm
6.5 The JPEG Image Compression Standard
6.6 ITU-T Recommendations
6.7 The MPEG Motion Video Compression Standard
6.8 DVI Technology
6.9Summary
6.10 Unit End Exercises
6.11 Additional Reference

6.1 OBJECTIVES

In this Chapter you will understand

- General techniques for video and image compression
- Several standardized compression systems, including JPEG, MPEG, p*64, and DVI Technology

6.2 INTRODUCTION

Reducing the amount of data needed to reproduce images or video (compression) saves storage space, increases access speed, and is the only way to achieve digital motion video on personal computers.

Evaluating a Compression System

In order to compare video compression systems, one must have ways to evaluate compression performance. Three key parameters need to be considered:

- Amount or degree of compression
- Image quality
- Speed of compression or decompression

In addition, we must also look at the hardware and software required by each compression method.

How Much Compression?

Compression performance is often specified by giving the ratio of input data to output data for the compression process (the compression ratio). This measure is a dangerous one unless you are careful to specify the input data format in a way that is truly comparable to the output data format.

A much better way to specify the amount of compression is to determine the number of bits per displayed pixel needed in the compressed bitstream. For example, if we are reproducing a 256 x 240 pixel image from a 15,000-byte bitstream, we are compressing

(bits) / (pixels)

(15,000 x 8) / (256 x 240) = 2 bits per pixel

How Good Is the Picture?

In this about picture quality performance of a compression system, it is helpful to divide the world of compression into two parts – lossless compression and lossy compression. Lossless compression means that the reproduced image is not changed in any way by the compression/decompression process; because we can use more efficient methods of data transmission than the pixel-by-pixel PCM format that comes from a digitizer.

On the other hand, lossy compression systems by definition do make some change to the image— something is different. Lossy compression systems may introduce any of the digital video artifacts, or they may even create some unique artifacts of their own. None of these effects is easy to quantify, and final decisions about compression systems, or about any specific compressed image, will usually have to be made after a subjective evaluation

How Fast Does It Compress or Decompress?

In most cases of storing still images, compression speed is less critical than decompression speed-since we are compressing the image ahead of time to store it, we can usually take our time in that process. On true other hand, decompression usually takes place while the user is waiting for the result, and speed is much more important. With motion video compression there is a need for fast compression in order to capture motion video in real time as it comes from a camera or VCR. In any case, compression and decompression speed is usually easy to, specify and measure.

What Hardware and Software Does It Take?

Some amount of compression and decompression can be done in software using standard PC hardware. Except with very simple algorithms, this approach quickly runs into speed problems—the process takes too long, simple algorithms do not provide the best compression. This is a moving target with time because of the continued advance in the processing power of PCs.

Redundancy

Redundancy in a digital video image occurs when the same information is transmitted more than once. For example:

- In any area of the picture where the same color spans more than one pixel location, there is redundancy between pixels, since adjacent pixels will have the same value. This applies both horizontally and vertically.

- When the scene or part of the scene contains predominantly vertically oriented objects, there is a possibility that two adjacent lines will be partially or completely the same, giving us redundancy between lines. These two types of redundancy (pixel and line) exist in any image and are called spatial redundancy.

- When a scene is stationary or only slightly moving, there is a further possibility of redundancy between frames of a motion sequence—adjacent frames in time are similar, or they may be related by a simple function such as translation. This kind of redundancy is called temporal redundancy.

Compression schemes may exploit any or all of these aspects of redundancy.

6.3 VIDEO COMPRESSION TECHNIQUES

A great deal of research has been done in image and video compression technology, going back more than 25 years. Many powerful techniques have been developed, simulated, and fully characterized in the literature; in fact, today it is quite difficult to invent something new in this field—it has been so well researched.

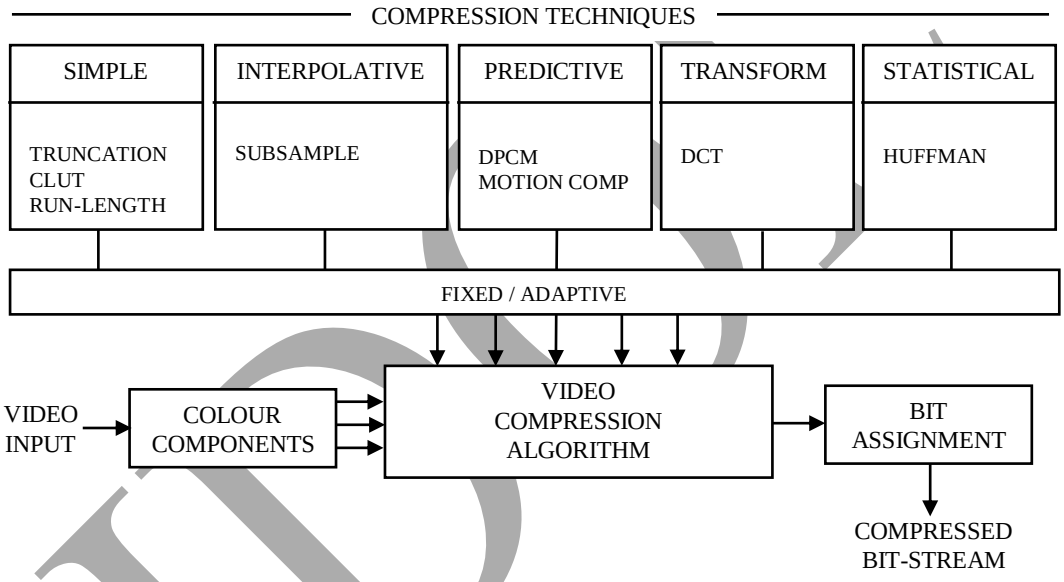


Figure 6.1: Compression techniques

However, broad application of the more sophisticated video compression approaches has not been practical because of the cost of the hardware required. That is now changing because of the power of high-performance digital signal processing chips and custom VLSI devices.

In this discussion, we will use the word technique to refer to a single method of compression—usable by itself, but possibly also used in combination with other techniques. On the other hand, an algorithm

refers to the collection of all the techniques used by any particular video compression system. Figure 6.1 is a block diagram of how techniques are used to create an algorithm.

We will assume that the input to the compression system is always a PCM digitized signal in color component (RGB, YUV, etc.) form. Most compression systems will deal with the color components separately, processing each one by itself. In decompression, the components similarly are separately recovered and then combined into the appropriate display format after decompression. Note, however, that there is nothing that requires that the individual color components be processed in the same way during compression and decompression—in fact, there are sometimes significant advantages to handling the components of the same image by different techniques. This brings up immediately that we must choose the color component format to use, and that choice could make a big difference in what performance is achieved by the system. Two obvious choices that we have already discussed are RGB components or luminance/chrominance components. Where it is relevant, the significance of the color component choice will also be covered.

Similarly, there is always a possibility of making any technique adaptive, which means that the technique can change as a function of the image content. Adaptivity is not a compression technique itself; rather, it is a way to cause any given technique to be more optimized locally in the image or temporally in the frame sequence. Almost all the compression techniques we will be discussing can be made adaptive, but of course this adds complexity. Where adaptivity is an important aspect, it will be discussed with each technique.

The output of a compression process is a bitstream—it is usually no longer a bitmap and individual pixels may not be recognizable. The structure of the bitstream is important, however, because it can also affect the compression efficiency and the behavior of the system when errors occur in transmission or storage. Therefore, the figure shows a separate box called bit assignment—this is where the bitstream structure is imposed on the compressed data. It may be a task which is subsumed in the algorithm, or it may be a separate step in the process.

6.3.1 Simple Compression Techniques

A good example of simple compression is truncation—reducing data through arbitrary lowering of the bits per pixel. This is done by throwing away some of the least significant bits for every pixel. If we go too far with truncation, we will begin to see contouring, and our image will start looking like a cartoon. However, many images can stand this up to a point; so, for example, we can usually truncate to 16 bpp with good results on real images. 16 bpp is usually done by assigning bits to color components such as R:G:B 5:5:5 or Y:V:U 6:5:6. In the R:G:B 5:5:5 case, the 16^{th} bit could be used as a flag for some other purpose, such as a keying signal. Truncation is attractive because its processing is extremely simple.

Another simple compression scheme, which creates a different kind of artifact, is the color lookup table (CLUT) approach. With a CLUT, the pixel values in the bitmap represent an index into a table of colors, but the table of colors will have much greater bpp than the pixel values. It is usually done with pixels having no more than 8 bpp, which means that the entire picture must be reproduced with 256 or fewer colors at a time. The colors in the CLUT are chosen from a palette represented by the color depth in the lookup table. For some kinds of images, that is not as

bad as it sounds-if the 256 colors are carefully chosen. However, that means each image must be processed ahead of time to choose the 256 best colors for that image (I unique CLUT must be created for each image), and that is a nontrivial amount of preprocessing. Going higher than 8 bpp with CLUT (more colors) will of course give better results, but by the time we get to 16 bpp, it will probably be better to simply use the truncation approach of the previous paragraph because the processing for truncation is much simpler.

A third simple technique is run-length (RL) coding. In this technique blocks of repeated pixels are replaced with a single value and a count of many times to repeat that value. It works well on images which have areas of solid colors - for example, computer-generated images, cartoons can CLUT images. Depending entirely on the kind of image, RL coding can achieve large amounts of compression—well below 1 bpp. However, its effectiveness is limited to images (or other data streams) that contain large numbers of repeated values, which is seldom the case for real images from a video camera.

6.3.2 Interpolative Techniques

Interpolative compression at the pixel level consists of transmitting a subset of the pixels and using interpolation to reconstruct the intervening pixels. Within our definition of compression, this is not a valid technique for use on entire pixels because we are effectively reducing the number of independent pixels contained in the output, and that is not compression. The interpolation in that case is simply a means for reducing the visibility of pixellation, but the output pixel count is still equal to the subset. However, there is one case where interpolation is a valid technique. It can be used just on the chrominance part of the image while the luminance part is not interpolated. This is called color

subsampling, and it is most valuable with luminance-chrominance component images (YUV, YIQ, etc.).

The color components I and Q of the YIQ format (in NTSC color television) were carefully chosen by the developers so that they could be transmitted at reduced resolution. This works because a viewer has poor acuity for color changes in an image, so the lower resolution of the color components really is not noticed. The same is true for YUV components, which are used in PAL television systems.

For example, in a digital system starting with 8 bits each of YUV (24 bpp total), we can subsample the U and V components by a factor of 4 both horizontally and vertically (a total ratio of 16:1). The selected U and V pixels remain at 8 bpp each, so we still are capable of the full range of colors. When the output image is properly reconstructed by interpolation, this technique gives excellent reproduction of real pictures. The degree of compression works out to 9 bpp:

bpp = (luminance) 8 + (UV) 16/(subsamp. ratio) 16 = 9

Please note that we have used the term "real" images when talking about the advantages of color subsampling and interpolation. It is not as effective on "nonreal," i.e., computer-generated images. Sometimes a computer-generated image using color subsampling and interpolation will have objectionable color fringes on objects, or thin colored lines may disappear. This is inherent in the technique.

Interpolation can also be applied between frames of a motion sequence. In this case, certain frames are compressed by still compression or by predictive compression; the frames between these are compressed by doing an interpolation between the other frames and sending only the

data needed to correct the interpolation. This will be covered further when discussing motion video compression algorithms.

6.3.3 Predictive Techniques

Anyone who can predict the future has a tremendous advantage - that applies to video compression as much as it applies to the stock market. In video compression, the future is the next pixel, or the next line, or the next frame. We said earlier that typical scenes contain a degree of redundancy at all these levels - the future is not completely different from the past Predictive compression techniques are based on the fact that we can store the previous item (frame, line, or pixel) and use it to help build the next item. If we can identify what is the same from one item to the next, we need only transmit the part that is different because we have predicted the part that is the same.

DPCM

The simplest form of predictive compression operates at the pixel level with a technique called differential PCM (DPCM). In DPCM, we compare adjacent pixels and then transmit only the difference between them. Because - adjacent pixels often are similar, the difference values have a high' probability of being small and they can safely be transmitted with fewer pits than it would take to send a whole new pixel. For example, if we are compressing 8-bit component pixels, and we use 4 bits for the different value, we can maintain the full 8-bit dynamic range as long as there is never a change of more than 16 steps between adjacent pixels. In this case, the DPCM step size is equal to one quantization step of the incoming signal.

In decompression, the difference information is used to modify the previous pixel to get the new pixel. Normally the difference bits would

represent only a portion of the amplitude range of an entire pixel, meaning that if adjacent pixels did call for a full-amplitude change from black to white, the DPCM system would overload. In that case, it would take i) number of pixel times (16, for the example of the last paragraph) before the output could reach full white, because each difference pixel only represents a fraction of the amplitude range. This effect is called slope overload, and it causes smearing of high-contrast edges in the image.

ADPCM

The distortion from slope overload may be reduced by going to adaptive DPCM (ADPCM). There are many ways to implement ADPCM, but one common approach is to adapt by changing the step size represented by the difference bits. In the previous example, if we knew that the black-to-white step was coming, we could increase the step size before the b-w step came, so that when we got there, the difference bits would represent full range, and a full-amplitude step could then be reproduced. After the step had been completed, the adaptive circuit would crank the step size back down in j order to better reproduce fine gradations. This changes the artifact from slope overload's smearing to edge quantization—an effect of quantization noise surrounding high-contrast edges. You might have a hard time deciding which is better.

In the previous example of ADPCM, we glossed over the problem of how the decompression system knows what step size to use at any time. This information must somehow be coded into the compressed bitstream. There are lots of ways for doing that (which we will not go into here) but you should be aware that using adaptation with any algorithm will add the problem of telling the decompression system how to adapt.

A certain amount of overhead data and extra processing will always be required to implement adaptation.

The DPCM example also highlights a problem of predictive compression techniques in general. What happens if an error creeps into the compressed data? Since each pixel depends on the previous pixel, one incorrect pixel value will tend to become many incorrect pixel values after decompression. This can be a serious problem. A single incorrect pixel would normally not be much of a problem in a straight PCM image, especially a motion image; it would just be a fleeting dot that a viewer might never see. However, if the differential system expands a single dot error into a line that goes all the way across the picture (or maybe even into subsequent lines), everyone will see it. Therefore, predictive compression schemes typically add something else to ensure that recovery from an error is possible and that it happens quickly enough that error visibility will not be objectionable. A common approach is to make a differential system periodically start over, such as at the beginning of each scanning line or at the beginning of a frame.

After all the previous discussion, it shouldn't be a surprise to say that DPCM or ADPCM are not widely used by themselves for video compression. The artifacts of slope overload and edge quantization become fatal as we try to achieve more than about 2:1 compression. The techniques, however, do find their way into more complex compression algorithms that combine other more powerful techniques with some form of differential encoding.

Other Predictive Techniques

Continuing with predictive compression schemes and moving to the next higher level, we should talk about prediction based on scanning

line redundancy. However, line-level prediction is not often used by itself; rather, it tends to be subsumed in the two-dimensional transform techniques which very neatly combine pixel and line processing in one package.

Prediction is also a valuable technique at the frame level for motion video compression. We will discuss it later.

6.3.4 Transform Coding Techniques

A transform is a process that converts a bundle of data into an alternate form which is more convenient for some, particular purpose. Transforms are ordinarily designed to be reversible—that is, there exists an inverse transform which can restore the original data. In video compression, a "bundle of data" is a group of pixels—usually a two-dimensional array of pixels from an image, for example, 8x8 pixels. Transformation is done to create an alternate form which can be transmitted or stored using less data. At decompression time, the inverse transform is run on the data to reproduce the original pixel information.

6.3.5 A Simple Transform Example

In order to explain how a transform works, we will make up a very simple example. Consider a *2* x 2 block of monochrome (or single-color component) pixels, as shown in Figure 6.2.

We can construct a simple transform for this block by doing the following:

1. Take pixel A as the base value for the block. The full value of pixel 4 will be one of our transformed values.
2. Calculate three other transformed values by taking the difference^ between the three other pixels and pixel A.

The following figure shows the arithmetic for this transformation, and it also shows the arithmetic for the inverse transform function. Note that we now have four new values, which are simply linear combinations of the four original pixel values. They contain the same information.

Now that we have made this transformation, we can observe^ that the redundancy has been moved around in the values so that the different values may be transmitted with fewer bits than the pixels themselves would have required. For example, if the original pixels were 8 bits each, the 2 x 2 block then used 32 bits. With the transform, we might assign 4 bits each for the difference values and keep 8 bits for the base pixel—this would reduce the data to only 8 + (3 x 4) or 20 bits for the 2x2 block (resulting in compression to 5 bits/pixel). The idea here is that the transform has allowed us to extract the differences between adjacent pixels in two dimensions, and errors in coding of these differences will be less visible than the same errors in the pixels themselves.

2x2 ARRAY OF PIXELS

A	**B**
C	**D**

TRANSFORM INVERSE TRANSFORM

X0= A	A_n = X0
X1 = B - A	B_n = X1 + X0
X2 = C - A	C_n = X2 + X0
X3 = D - A	D_n = X3 + X0

Figure 6.2:Example of simple transform coding

This example is not really a useful transform—it is too simple; Useful transforms typically operate on larger blocks, and they perform more complex calculations. In general, transform coding becomes more effective with larger block sizes, but the calculations also become more difficult with larger blocks. The trick in developing a good transform is to make it effective with calculations that are easy to implement in hardware or software and will run fast. It is beyond our scope here to describe all the transforms that have been developed for image compression, but you can find them in the literature. The Discrete Cosine Transform (DCT) is especially important for video and image compression and is covered in detail below.

The Discrete Cosine Transform

The DCT is performed on a block of horizontally and vertically adjacent pixels—typically 8x8. Thus, 64 pixel values at a time are processed by the transform; the output is 64 new values, representing amplitudes of the two-dimensional spatial frequency components of the 64-pixel block. These are referred to as DCT coefficients. The coefficient for zero spatial frequency is called the DC coefficient, and it is the average value of all the pixels in the block. The remaining 63 coefficients are the AC coefficients, and they represent the amplitudes of progressively higher horizontal and vertical spatial frequencies in the block.

Since adjacent pixel values tend to be similar or vary slowly from one to another, the DCT processing provides opportunity for compression by forcing most of the signal energy into the lower spatial frequency components. In most cases, many of the higher-frequency coefficients will have zero or near-zero values and can be ignored.

A DCT decoder performs the reverse process—spatial frequency coefficients are converted back to pixel values. Theoretically, if DCT encoding and decoding is done with complete precision, the process of encoding followed by decoding would be transparent. However, in a real system there will be slight errors because the signals have been quantized with finite numbers of bits, and the DCT algorithm involves transcendental mathematical functions, which can only be approximated in any real system. Thus, the process will not be perfectly transparent. The trick is to choose the quantizing parameters so that the errors are not visible in the reproduced image. This is successfully done in the standards discussed later, but the small remaining errors explain why DCT cannot be used for lossless compression.

6.3.6 Statistical Coding

Another means of compression is to take advantage of the statistical] distribution of the pixel values of an image or of the statistics of the data created from one of the techniques discussed above. These are called statistical coding techniques, or sometimes entropy coding, and they may be contained either in the compression algorithm itself, or applied separately as part of the bit assignment following another compression technique. The usual case for image data is that all possible values are not equally probable—there will be some kind of non-uniform distribution of the valued Another way of saying that is: Some data values will occur more frequently than other data values. We can set up a coding technique which codes the more frequently occurring values with words using fewer bits, and the less frequently occurring values will be coded with longer words. This results in, a reduced number of bits in the final bitstream, and it can be a lossless technique. One widely used form of this coding is called Huffman coding.

The above type of coding has some overhead, however, in that we must tell the decompression system how to interpret a variable-word-length bitstream. This is normally done by transmitting a table (called a code book) ahead of time. This is simply a table which tells how to decode the bitstream back to the original values. The code book may be transmitted once for each individual image, or it may even be transmitted for individual blocks of a single image. On the compression side, there is overhead needed to figure out the code book—the data statistics must be calculated for an image or for each block.

6.3.7 Motion Video Compression Techniques

In the still-image compression techniques that we discussed above, we gave little consideration to the matter of compression or decompression speeds With still images, processing only needs to be fast enough that the user does not get bored waiting for things to happen. However, when one begins to think about motion video compression systems, the speed issue becomes overwhelming. Processing of a single image in one second or less is usually satisfactory for stills. However, motion video implies a high enough frame rate to produce subjectively smooth motion, which for most people is 15 frames per second or higher. Full-motion video as used here refers to normal television frame rates—25 frames per second for European systems, and 30 frames per second for North America and Japan. These numbers mean that* our digital video system must deliver a new image every 30-40 milliseconds. If the system cannot do that, motion will be slow or jerky, and the system will quickly be judged unacceptable.

At the same time that we need more speed for motion compression, we also need to accomplish more compression. This comes about because

of data rate considerations. Storage media have data rate limitations, so they cannot simply be speeded up to deliver data more rapidly. For example, the CDROM's continuous data rate is fixed at 153,600 bytes per second—there is no way to get data out faster. If CD-ROM is being used for full-motion video at 30 frames per second, we will have to live with 5,120 bytes per frame. Therefore, we face absolute limits on the amount of data available for each frame of motion video (at least on the average); this will determine the degree of compression we must achieve.

For CD-ROM at 5,120 bytes of data per frame (40,960 bits per frame) and at a resolution of 256 x 240 pixels, the required compression works out to be 0.67 bits per pixel. Some still compression systems can work down to this level, but the pictures are not very good, and 256 x 240 already is a fairly low pixel count. Therefore, we should look at motion video to see if there are possibilities for compression techniques which can be used in addition to the techniques we discussed for stills.

Fortunately, motion video offers its own opportunities to achieve additional compression. There is the redundancy between adjacent frames—a motion video compression system can (or must) exploit that redundancy. Techniques for dealing with this are prediction and interpolation or a special technique called motion compensation. We will discuss motion compensation shortly.

Another concept that comes into play with motion video systems is the idea of symmetry between compression and decompression. A symmetric compression/decompression system will use the same hardware for both compression and decompression and perform both processes at roughly the same speed. Such a system for motion video

will require hardware that is too expensive for a single-user system, or else it will have to sacrifice picture quality in favour of lower-cost hardware. The reason is that a symmetric system must digitize and compress motion video in real time, which implies that the system must process data rates that can exceed 20 Mb per second.

However, this problem can be effectively bypassed by the use of an asymmetric system where the compression is performed on expensive hardware, but the decompression is done by low-cost hardware. This works in situations where the single-user system needs only to play back compressed video which has been prepared ahead of time—it will never have to do compression.

In fact, most Interactive video applications do not require that the end-user system contains a compression capability—only decompression. Motion video for this class of application can be compressed (once) during the application design process, and the final user only plays back the compressed video. Therefore, the cost of the compression process is shared by all the users of the application. This concept can lead to the establishment of a centralized compression service which performs compression for many application developers, thus sharing the costs even further.

Motion Compensation

Consider the case of a motion video sequence where nothing is moving in the scene. Each frame of the motion video should be exactly the same as the previous one. In a digital system, it is clear that all we need to do Is transmit the first frame of this scene, store that and simply display the same frame until something moves. No additional information needs to be sent during the time the image is stationary.

However, if now a dog walks across our scene, we have to do something to introduce this motion. We could simply take the image of the walking dog by itself, and send that along with the coordinates of where to place it on the stationary background scene sending a new dog picture for each frame. To the extent that the dog is much smaller than the total scene, we are still not using much data to achieve a moving picture.

The example of the walking dog on a stationary background scene is an overly simplified case of motion video, but it already reveals two of the I problems involved in motion compensation:

- How can we tell if an image is stationary?
- How do we extract the part of the image which moves?

We can try to answer these questions by some form of comparison of adjacent frames of the motion video sequence. We can assume that both the previous and the current frames are available to us during the compression process. If we do a pixel-by-pixel compare between the two frames, the compare should produce zero for any pixels which have not changed, and it will be nonzero for pixels which are somehow involved in motion. Then we could select only the pixels with nonzero compares and send them to the decompressing system. Of course, we would have to also send some information which tells the decompressing system where to put these pixels.

However, this very simple approach, which is a form of frame-to-frame DPCM, is really not too useful because of several problems. First, the pixel compare between frames will seldom produce a zero, even for a completely I stationary image, because of analog noise or quantizing

noise in the system. This could be alleviated by introducing a threshold that would let us accept small comparison values as zero, but there is a more serious problem—images from video or film cameras are seldom stationary. Even if the scene itself contains no motion (which is unusual in natural scenes) the camera may be moving slightly, causing all pixel compares to fail. Even partial pixel, movements will create changes large enough to upset the comparison) technique.

Therefore, more sophisticated techniques are needed to do the motion; detection for the purpose of motion compensation. This problem is usually addressed by dividing the image into blocks, just as we did with still image* for transform coding. Each block is examined for motion, using approaches which consider all of the pixels in the block for motion detection of that block. If the block is found to contain no motion, a code is sent to the decompressor to leave that block the way it was in the previous frame. If the block does have motion, a transform may be performed and the appropriate bits sent to the decompressor to reproduce that block with the inverse transform.

If enough computing power is available for the compression process, still more sophisticated approaches can be pursued. For example, blocks which contain motion can be further examined to see if they are simply a translation of a block from the previous frame. If so, only the coordinates of the translation (motion vectors) need to be sent to tell the decompressor how to create that block from the previous frame. A variation of this approach is used in the MPEG video compression standard. Even more elaborate techniques can be conceived to try to create the new frame using as much as possible of the information from the previous frame instead of having to send new information.

6.4 STANDARDIZATION OF ALGORITHM

The preceding discussion of techniques introduced the building blocks available for creating algorithms. An actual algorithm consists of one or more techniques which operate on the raw digitized image to create a compressed bitstream. The number of algorithm possibilities is nearly infinite. However, practical applications require that all users who wish to interchange compressed digital images or video must use exactly the same algorithm choice. Further, sophisticated algorithms will benefit from the development of special hardware or processing chips, where the algorithm and its options may be cast in the silicon. All this expresses the need for a standard to allow the orderly growth of markets which utilize image or video compression technology.

A successful example of a digital imaging market that took off once a standard was developed is the Group 3 facsimile machine, which is standardized under CCITT Recommendation T.4.1980. However, this Recommendation applies only to bilevel images (one bit per pixel), whereas here we are interested in continuous-tone images, typically represented by 8 bits per pixel or more, and often in color. Applications such as desktop publishing, graphic arts, color facsimile, wirephoto transmission, medical imaging, computer multimedia, and others, have a serious need for a continuous tone image compression standard.

Driven by these needs, there has been a strong effort to develop international standards for still image and motion video compression algorithms, under way for several years in the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). There are two working parties for algorithm standardization in a joint ISO/1 EC Committee (called JTC1). These

working parties are the Joint Photographic Expert Group (JPEG), which considers still image standards.

6.5 THE JPEG IMAGE COMPRESSION STANDARD

The JPEG (Joint Photographic Experts Group) became an international standard in 1992. In the sequential mode JPEG compression process is composed of following steps:

- Preparation of data
- Source encoding steps involving forward DCT and quantization
- Entropy encoding steps involving RLE and Huffman encoding

DECOMPRESSION PROCESS

- Entropy decoding steps involving RLD and Huffman decoding
- Source decoding steps involving inverse DCT and de-quantization

6.5.1 BLOCK PREPARATION

An image is represented by 1 or more 2D array of pixel values these blocks are in preparation of next steps where DCT is applied to each block instead of the entire image.

6.5.2 DISCRETE COSINE TRANSFORM (DCT)

The objective of this is to transform each block from the spatial domain to the frequency domain. We know that the synthesis equation of the DFT is given by the relation.

$$x[i] = \sum_{k=0}^{N/2} \mathrm{Re}\ \bar{X}[k] \cos(2\pi ki / N) + \sum_{k=0}^{N/2} \mathrm{Re}\ \bar{X}[k] \sin(2\pi ki / N)$$

where *N* is the total number of samples, *k* is the variable, *i* is the variable indicating the number of input sample considered in the time or(space) domain, x[i] is the actual time(or space) domain signal, $\mathrm{Re}\,\bar{X}[k]$ is proportional to the k-th entry and the frequency domain. If we ignore the imaginary components and consider the entire time (or space) domain signal to be compiled of real numbers we get

$$x[i] = \sum_{k=0}^{N-1} \mathrm{Re}\,\bar{X}[k] \cos(2\pi k i / N)$$

The above expression is called the synthesis equation of a one dimensional DCT, also known as inverse DCT. The forward DCT is just the reverse of the above relation, where we express the frequency domain in terms of time (or space) domain signal. Mathematically it can be expressed as

$$\mathrm{Re}\,X[k] = \sum_{i=0}^{N-1} x[i] \cos[\pi i(k + 1/2)/N)$$

or,

$$\mathrm{Re}\,X[k] = \sum_{i=0}^{N-1} x[i] \cos[\pi i(2k + 1)/2N)$$

The expression for 2D forward DCT is written as

$$F[i, j] = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} P[x, y] \cdot \cos[\pi i(2x + 1)/2N] \cdot \cos[\pi j(2y + 1)/2N]$$

6.5.3QUANTIZATION

The next step in the process is quantizing the coefficient numbers that were derived from the luminance and chrominance values by the DCT. Quantizing is basically the process of rounding off the numbers.

This is where the file compression comes in. How much the file is compressed depends on the quantization matrix.

The quantization matrix defines how much the information is compressed by dividing the coefficients by a quantizing factor. The larger the number of the quantizing factor, the higher the quality (therefore, the less compression). This is basically what is going onin Photoshop when you save as JPEG and the program asks you to set the quality; you are simply defining the quantizing factor.

Once the numbers are quantized, they are run through a binary encoder that converts the numbers to the ones and zeros computers love so well. You now have a compressed file that is on average about one-fourth of the size of an uncompressed file.

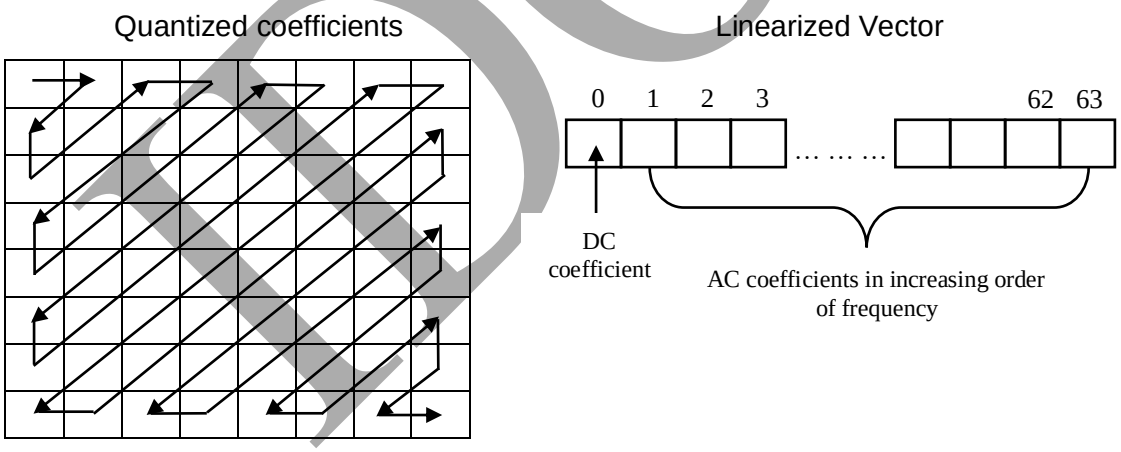


Figure 6.3 Quantization

6.5.3 ZIGZAG SCAN

After the DCT stage, the remaining stages involve entropy encoding. The entropy coding algorithms operate on one dimensional string of values i.e. a vector. The output of the quantization stage is a 2D array,

hence to apply an entropy scheme, the array is to be converted to a 1D vector. This operation is known as vectoring.

6.5.4 DPCM encoding

There is one DC co-efficient per block because of the small physical area covered by each block, the DC co-efficient varies from one block to the next. To exploit this similarity, the sequence of DC co-efficient is encoded in DPCM mode. This means the difference between the DC co-efficient of each block and the adjacent block is computed and stored.

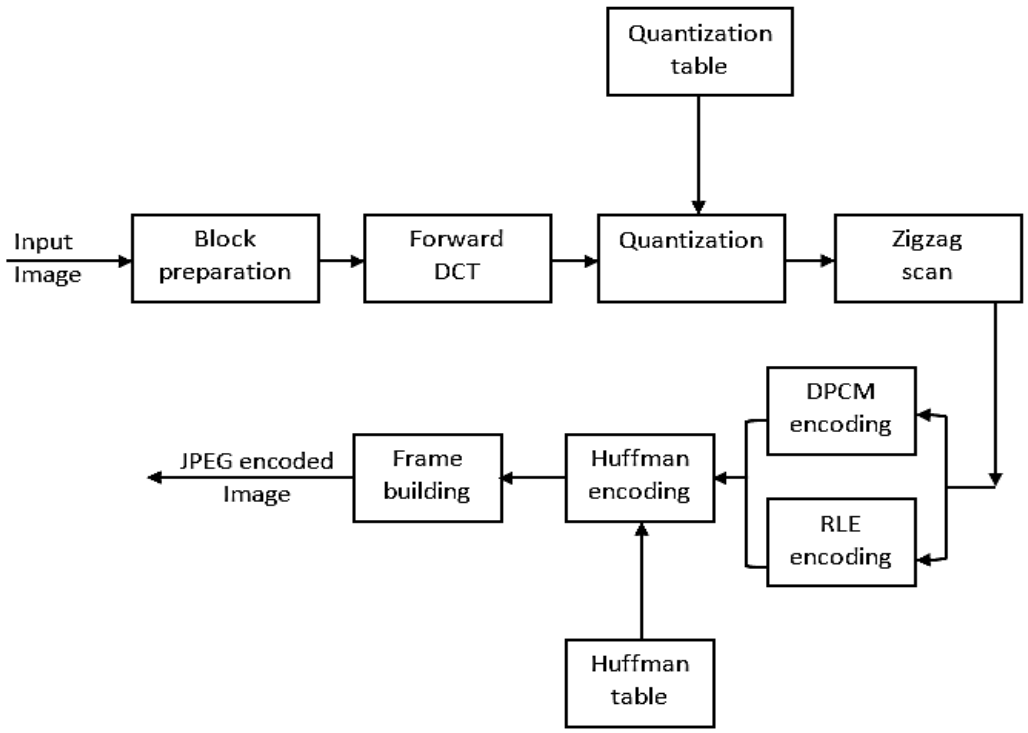


Figure 6.4 JPEG Encoder

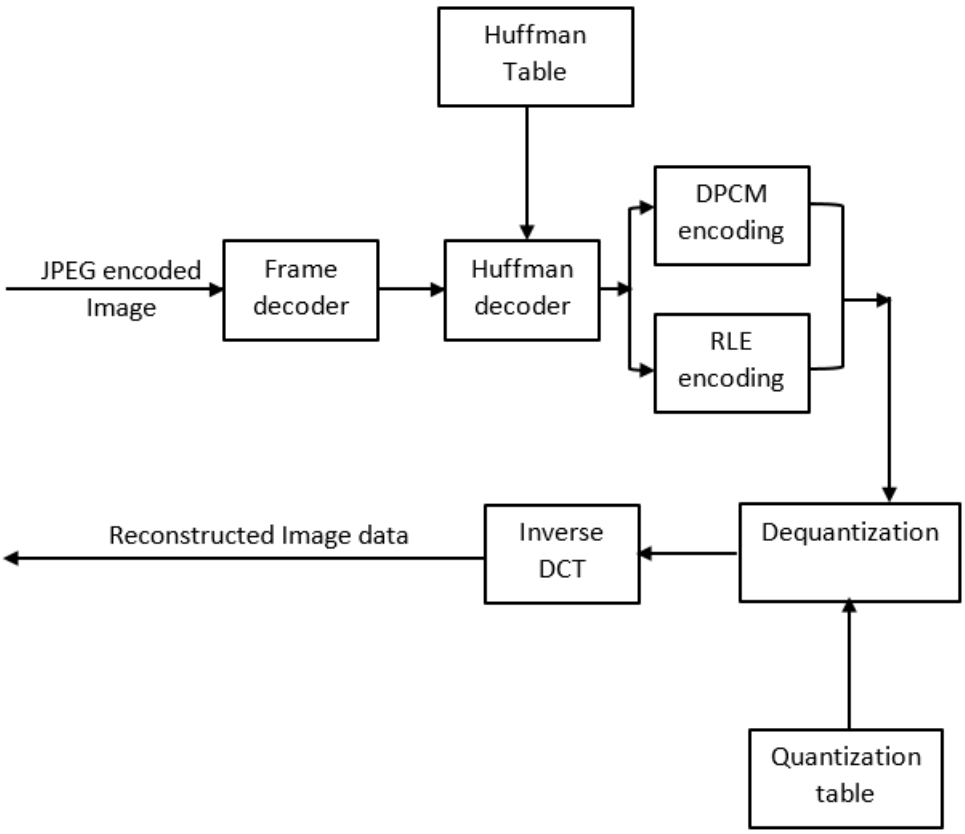


Figure 6.5 JPEG Decoder

6.6 ITU-T Recommendations

The H.261 standard developed in 1988-90 was a fore runner to the MPEG-1 and was designed for video conferencing applications over ISDN telephone lines. The baseline ISDN has a bit-rate of 64 Kbits/ sec and at the higher end, ISDN supports bit-rates having integral multiples (p) of 64 Kbits/sec. For this reason, the standard is also referred to as the p x 64 Kbits/sec standard.

In addition to forming a basis for the MPEG-1 and MPEG-2 standards, the H.261 standards offers two important features:

- Maximum coding delay of 150 msec. It has been observed that delays exceeding 150 msec do not provide direct visual feedback in bi-directional video conferencing

- Amenability to VLSI implementation, which is important for widespread commercialization of videophone and teleconferencing equipment.

6.6.2 Picture formats and frame-types in H.261

The H.261 standard supports two picture formats:

Common Intermediate Format (CIF), having 352 x 288 pixels for the luminance channel (Y) and 176 x 144 pixels for each of the two chrominance channels U and V. Four temporal rates: 30, 15, 10 or 7.5 frames/sec are supported. CIF images are used when p ≥ 6, that is for video conferencing applications.

Quarter of Common Intermediate Format (QCIF) having 176 x 144 pixels for the Y and 88 x 72 pixels each for U and V. QCIF images are normally used for low bit-rates applications like videophones (typically p =1). The same four temporal rates are supported by QCIF images also.

H.261 frames are of two types

1. I-frames: These are coded without any reference to previously coded frames.

2. P-frames: These are coded using a previous frame as a reference for prediction.

6.6.3 H.261 Bit-stream structure

The H.261 bit-stream follows a hierarchical structure having the following layers:

- Picture-layer, that includes start of picture code (PSC), time stamp reference (TR), frame-type (I or P), followed by Group of Blocks (GOB) data.

- GOB layer that includes a GOB start code, the group number, a group quantization value, followed by macroblocks (MB) data.

- MB layer, that includes macroblock address (MBA), macroblock type (MTYPE: intra/inter), quantizer (MQUANT), motion vector data (MVD), the coded block pattern (CBP), followed by encoded block.

- Block-layer, that includes zig-zag scanned (run, level) pair of coefficients, terminated by the end of block (EOB)

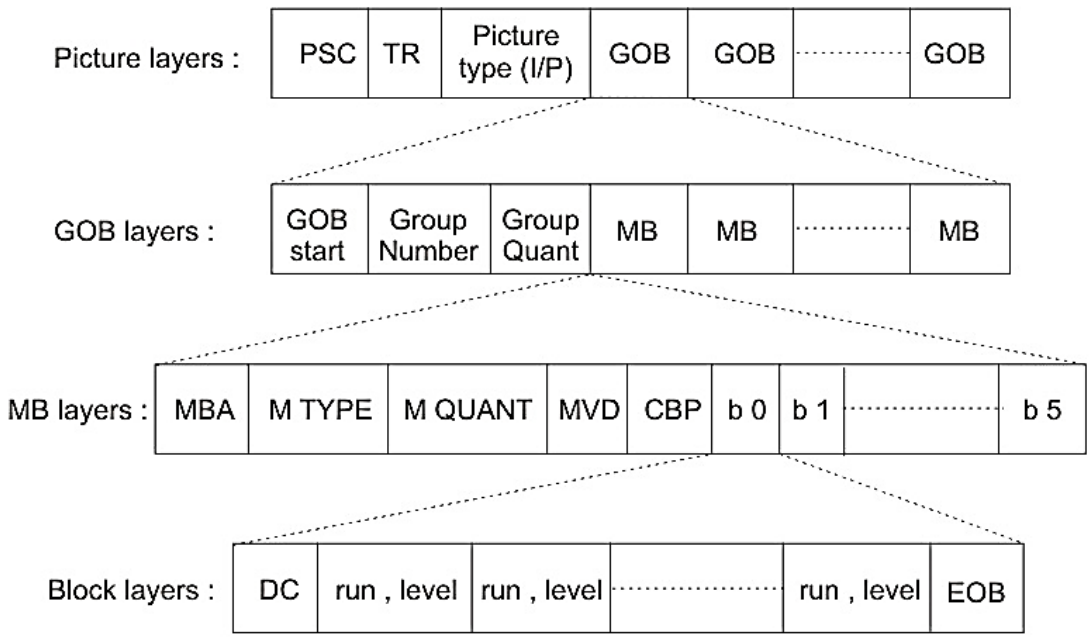


Figure 6.6: H.261 bit-stream structure

It is possible that encoding of some GOBs may have to be skipped and the GOBs considered for encoding must therefore have a group number, as indicated. A common quantization value may be used for the entire GOB by specifying the group quantizer value. However, specifying the MQUANT in the macroblock overrides the group quantization value. Major elements of the hierarchical data structure are discussed below:

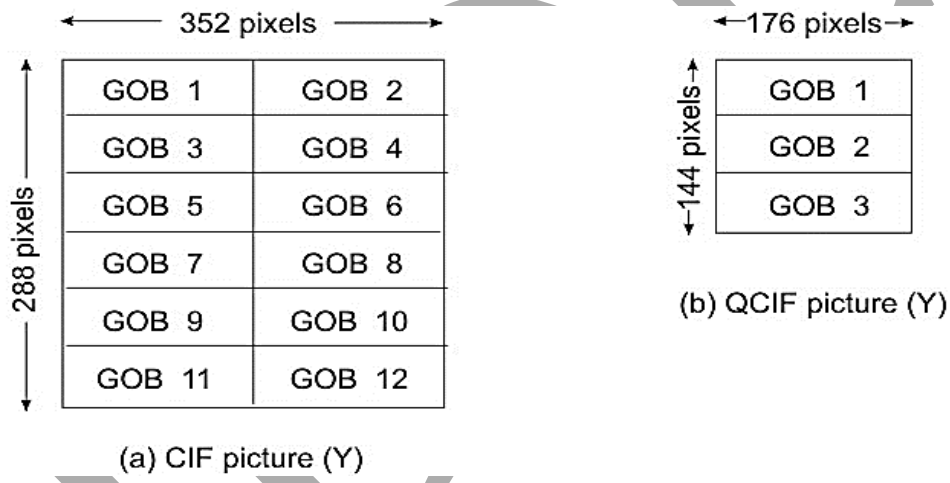


Figure 6.7: Arrangement of GOBs in (a) CIF Resolution and (b) QCIF Resolution

Each GOB thus relates to 176 pixels by 48 lines of Y and 88 pixels by 24 lines each of U and V. Each GOB must therefore comprise of 33 macroblocks - 11 horizontally and 3 vertically, as shown in Figure 6.8. Each box corresponds to a macroblock and the number corresponds to the macroblock number.

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

Figure 6.8: Composition of a GOB.

Data for each GOB consists of a GOB header followed by data for macroblocks. Each GOB header is transmitted once between picture start codes in the CIF or QCIF sequence.

6.6.4 Macroblock layer:

As already shown in Figure 6.8, each GOB consists of 33 macroblocks. Each macroblock relates to 16 x 16 pixels of Y and corresponding 8 x 8 pixels of each U and V, as shown in Figure 6.9.

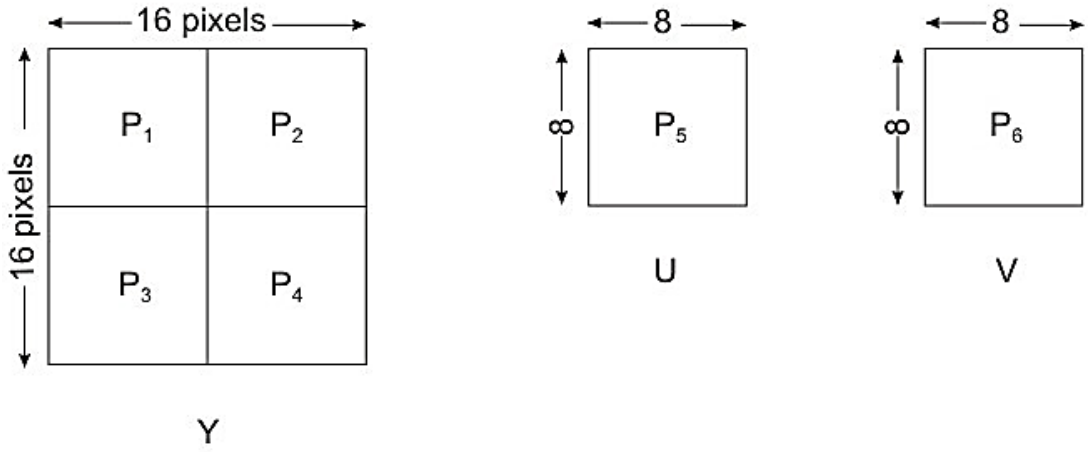


Figure 6.9: Composition of a macroblock

Since block is defined as a spatial array of 8 x 8 pixels, each macroblock therefore consists of six blocks - four from Y and one each

from U and V. Each macroblock has a header that includes the following information:

Macroblock address (MBA)

It is a variable length codeword indicating the position of a macroblock within a group of blocks. For the first transmitted macroblock in a GOB, MBA is the absolute address. For subsequent macroblocks, MBA is the difference between the absolute address of the macroblock and the last transmitted macroblock.

MBA is always included in transmitted macroblocks. Macroblocks are not transmitted when they contain no information for that part of the picture they represent.

Macroblock type (MTYPE)

It is also variable length codeword that indicates the prediction mode employed and which data elements are present. The H.261 standard supports the following prediction modes:

- intra modes are adopted for those macroblocks whose content change significantly between two successive macroblocks
- inter modes employ DCT of the inter-frame prediction error
- inter + MC modes employ DCT of the motion compensated prediction error
- inter +MC + fil modes also employ filtering of the predicted macroblock

Quantizer (MQUANT)

It is present only if so indicated by the MTYPE. MQUANT overrides the quantizer specified in the GOB and till any subsequent MQUANT is specified, this quantizer is used for all subsequent macroblocks.

Motion vector data (MVD)

It is also a variable length codeword (VLC) for the horizontal component of the motion vector, followed by a variable length codeword for the vertical component. MVD is obtained from the macroblock vector by subtracting the vector of the preceding macroblock.

Coded block pattern (CBP)

CBP gives a pattern number that signifies which of the block within the macroblock has at least one significant transformation coefficient. The pattern number is given by

$$32P_1 + 16P_2 + 8P_3 + 4P_4 + 2P_5 + P_6$$

Where, P_n=1 if any coefficient is present for block n, else 0. The block numberings are as per Figure 6.9.

6.6.5 Block Layer

The block layer does not have any separate header, since macroblock is the basic coding entity. Data for a block consists of codewords of transform coefficients (TCOEFF), followed by an end of block (EOB) marker.

Transform coefficients are always present for intra macroblocks. For inter-coded macro blocks, transform coefficients may or may not be

present within the block and their status is given by the CBP field in the macroblock layer.

TCOEFF encodes the (RUN, LEVEL) combinations using variable length codes, where RUN indicates run of zero coefficient in the zig-zag scanned block DCT array.

6.7 THE MPEG MOTION VIDEO COMPRESSION STANDARD

MPEG-4, formally the standard ISO/IEC 14496, was ratified by ISO/IEC in March 1999 as the standard for multimedia data representation and coding. In addition to video and audio coding and multiplexing, MPEG-4 addresses coding of various two- or three-dimensional synthetic media and flexible representation of audio-visual scene and composition.

As the usage of multimedia developed and diversified, the scope of MPEG-4 was extended from its initial focus on very low bit-rate coding of limited audio-visual materials to encompass new multimedia functionalities.

Unlike pixel-based treatment of video in MPEG-1 or MPEG-2, MPEG-4 supports content-based communication, access, and manipulation of digital audio-visual objects, for real-time or non-real-time interactive or non-interactive applications.

MPEG-4 offers extended functionalities and improves upon the coding efficiency provided by previous standards. For instance, it supports variable pixel depth, object-based transmission, and a variety of networks including wireless networks and the Internet.

Multimedia authoring and editing capabilities are particularly attractive features of MPEG-4, with the promise of replacing existing word processors. In a sense, H.263 and MPEG-2 are embedded in MPEG-4, ensuring support for applications such as digital TV and videophone, while it is also used for web-based media streaming.

MPEG-4 distinguishes itself from earlier video coding standards in that it introduces object-based representation and coding methodology of real or virtual audio-visual (AV) objects. Each AV object has its local 3D+T coordinate system serving as a handle for the manipulation of time and space. Either the encoder or the end-user can place an AV object in a scene by specifying a co-ordinate transformation from the object's local co-ordinate system into a common, global 3D+T co-ordinate system, known as the scene co-ordinate system.

The composition feature of MPEG-4 makes it possible to perform bit stream editing and authoring in compressed domain.

One or more AV objects, including their spatio-temporal relationships, are transmitted from an encoder to a decoder. At the encoder, the AV objects are compressed, error-protected, multiplexed, and transmitted downstream.

At the decoder, these objects are demultiplexed, error corrected, decompressed, composited, and presented to an end user. The end user is given an opportunity to interact with the presentation. Interaction information can be used locally or can be transmitted upstream to the encoder.

The transmitted stream can either be a control stream containing connection setup, the profile (subset of encoding tools), and class definition information, or be a data stream containing all other information.

Control information is critical, and therefore it must be transmitted over reliable channels; but the data streams can be transmitted over various channels with different quality of service.

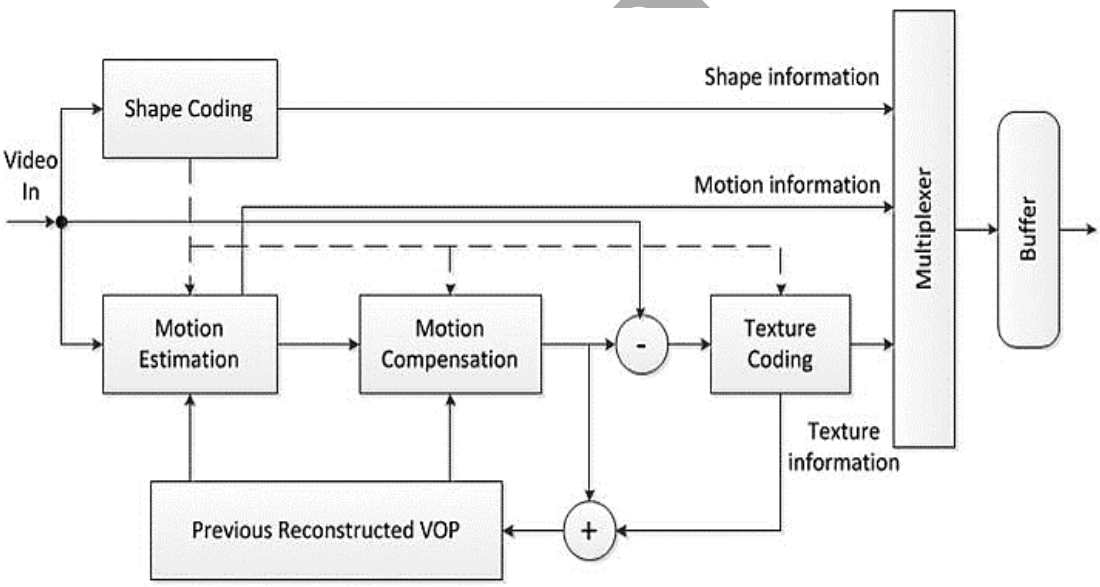


Figure 6.10: Video Object encoder structure - MPEG-4

6.8 DVI TECHNOLOGY

Digital Video Interactive (DVI) was the first multimedia desktop video standard for IBM-compatible personal computers. It enabled full-screen, full motion video, as well as stereoaudio, still images, and graphics to be presented on a DOS-based desktop computer. The scope of Digital Video Interactive encompasses a file format, including a digital

container format, a number of video and audio compression formats, as well as hardware associated with the file format.

Development of DVI was started around 1984 by Section 17 of The David Sarnoff Research Center Labs (DSRC) then responsible for the research and development activities of RCA.When General Electric purchased RCA in 1986, GE considered the DSRC redundant with its own labs, and sought a buyer. In 1988, GE sold the DSRC to SRI International,but sold the DVI technology separately to Intel Corporation.

DVI technology allowed full-screen, full motion digital video, as well as stereo audio, still images, and graphics to be presented on a DOS-based desktop computer. DVI content was usually distributed on CD-ROM discs, which in turn was decoded and displayed via specialized add-in card hardware installed in the computer. Audio and video files for DVI were among the first to use data compression, with audio content using ADPCM. DVI was the first technology of its kind for the desktop PC, and ushered in the multimedia revolution for PCs.

DVI was announced at the second annual Microsoft CD-ROM conference in Seattle to a standing ovation in 1987. The excitement at the time stemmed from the fact that a CD-ROM drive of the era had a maximum data playback rate of ~1.2 Mbit/s, thought to be insufficient for good quality motion video. However, the DSRC team was able to extract motion video, stereo audio and still images from this relatively low data rate with good quality.

The first implementation of DVI developed in the mid-80s relied on three 16-bit ISA cards installed inside the computer, one for audio processing,

another for video processing, and the last as an interface to a Sony CDU-100 CD-ROM drive. The DVI video card used a custom chipset (later known as the i80750 or i750 chipset) for decompression, one device was known as the pixel processor & the display device was called the VDP (video display processor).

Later DVI implementations used one, more highly integrated card, such as Intel's ActionMedia series (omitting the CD-ROM interface). The ActionMedia (and the later ActionMedia II) were available in both ISA and MCA-bus cards, the latter for use in MCA-bus PCs like IBM's PS/2 series.
Intel utilized the i750 technology in driving creation of the MMX instruction set. This instruction set was enabled in silicon on the original Pentium processors.

The DVI format specified two video compression schemes, **Presentation Level Video** or **Production Level Video (PLV)** and **Real-Time Video (RTV)** and two audio compression schemes, **ADPCM** and **PCM8**.

The original video compression scheme, called **Presentation Level Video (PLV)**, was asymmetric in that a Digital VAX-11/750 minicomputer was used to compress the video in non-real time to 30 frames per second with a resolution of 320x240. Encoding was performed by Intel at its facilities or at licensed encoding facilities set up by Intel.

Video compression involved coding both still frames and motion-compensated residuals using Vector Quantization (VQ) in dimensions 1, 2, and 4. The resulting file (in the .AVS format) was displayed in real-

time on an IBM PC-AT (i286) with the add-in boards providing decompression and display functions at NTSC (30 frame/s) resolutions.

The IBM PC-AT equipped with the DVI add-in boards hence had 2 monitors, the original monochrome control monitor, and a second Sony CDP1302 monitor for the colour video. Stereo audio at near FM quality was also available from the system.

The Real-Time Video (RTV) format was introduced in March 1988, then called Edit-Level Video (ELV).In Fall 1992, version 2.1 of the RTV format was introduced by Intel as Indeo2

6.9 SUMMARY

- Interpolative compression at the pixel level consists of transmitting a subset of the pixels and using interpolation to reconstruct the intervening pixels

- In decompression, the difference information is used to modify the previous pixel to get the new pixel

- A transform is a process that converts a bundle of data into an alternate form which is more convenient for some, particular purpose

- MPEG-4, formally the standard ISO/IEC 14496, was ratified by ISO/IEC in March 1999 as the standard for multimedia data representation and coding.

- The H.261 standard supports two picture formats: Common Intermediate Format (CIF), Quarter of Common Intermediate Format (QCIF)

- The H.261 standard supports two picture formats: Common Intermediate Format (CIF) and Quarter of Common Intermediate Format (QCIF)

- MPEG-4, formally the standard ISO/IEC 14496, was ratified by ISO/IEC in March 1999 as the standard for multimedia data representation and coding

- Digital Video Interactive (DVI) was the first multimedia desktop video standard for IBM-compatible personal computers

- The DVI format specified two video compression schemes, **Presentation Level Video** or **Production Level Video (PLV)** and **Real-Time Video (RTV)** and two audio compression schemes, **ADPCM** and **PCM8**.

6.10 UNIT END EXERCISES

1. Write a note on Interpolative Compression.

2. How does a JPEG compression work?

3. What happens in MPEG Compression?

4. Explain with the help of diagram Video Compression Techniques.

5. State the basic objective of H.261 standard.

6. Name the picture format supported by H.261.

7. Show the H.261 bitstream.

8. Show the Group of Block (GOB) arrangements in an H.261 picture.

9. Show the structure of block layer in H.261.

10. What is DVI Technology?

11. What is quantization?

12. Explain Discrete Cosine Transform.

6.11 ADDITIONAL REFERENCE

- Principles of Multimedia, Eighth reprint edition 2009, Ranjan Parekh, Tata McGraw-Hill Companies.

- https://nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Multimedia%20Processing/New_index1.html

- https://en.wikipedia.org/wiki/Digital_Video_Interactive

7

Operating System Support for Continuous Media Applications

Unit Structure

7.1 Objectives
7.2 Introduction
7.3 Limitations in Workstation Operating Systems
7.4 New OS Support
7.5 Experiments using Real-Time Mach
7.6 Summary
7.7 Unit End Exercises
7.8 Additional Reference

7.1 OBJECTIVES

In this Chapter you will understand:

- Limitations of Workstation Operating Systems
- Systems support for multimedia computing from three different points of view – architectural support, resource management support and programming support

7.2 INTRODUCTION

Manymodernworkstationsareequippedwithspecializedhardwarecapableofproducingdigitalaudioanddisplayinghigh-resolutiongraphics.Thecurrenttrendinmultimediacomputingistowardincorporatingfull-motiondigitalvideoandaudiointomanytypesofapplications.Suchapplicationsran

gefrommultimediamail,hypermedia,conferencingsystems,remoteteleprocessing,andvirtualreality.Wecallthese*continuousmediaapplications.*

Althoughcontinuousmediaapplicationsarebecomingverypopularinworkstationenvironments,currentworkstationoperatingsystemsfacemanyproblemsinsupportingmultipleinstancesofcontinuousmediaapplications.Forinstance,weoftenencounterajitterproblemwhilewearereadingvideomailwheneverabackgroundfiletransferprogramstarts.Amusicplayprogramoftenacceleratesitsspeedwhenthecontendingprogramsterminate.

Similarly,runningmanyvideophoneprogramsoftenplacesthesystemintoanoverloadsituation.However,thecurrentsystemsdonotprovideanintelligentoverloadcontrolormanagementscheme.Insteadofhaveveryprogramsufferfromtheoverload,ausermaywanttomaintainthequalityofservice(QOS)ofonlythecurrentimportantsessions.

Theseproblemsoriginatefromtwofactors:thelackofreal-timesupportfromtheoperatingsystem,whichcanprovidebettertime-drivencomputation,andthelackofQOS-basedresourcemanagementwherethesystemcanguaranteethequalityofserviceoftheactiveapplications.

InaQOS-basedresourcemanagementscheme,itisnotsufficienttosimplyspecifyaQOSlevelatsessioncreationtimewhichstaticallyremainsinforceforthelifeofthesession.Instead,wehaveproposedandimplementedadynamicQOScontrolschemewithCBSRP(Capacity-BasedSessionReservationProtocol)forreal-timecommunications.

InourQOSmodel,aQOSlevelforcontinuousmediaobjectscanbeexpressedwithtemporalaridspatialresolutions.Thetemporalresolutioncanbeexpressedbythenumberofframespersecond(fps)orsamplingrate.Thespatialresolutioncanbeexpressedbydatasize,thenumberofbitsperpixel(bpp),compressionscheme,compress

ionratio,andsoon.Forasimpledigitalvideosession,ausertaskmaychooseanacceptablerangeoffpsforitstemporalresolutionandarangefromtheimagesizeandthebpp,suchas8,16,or24bppforthespatialresolution.Byspecifyingtheseranges,thesystemoruserwillbeabletochange"QOSofexistingsessionsduringthecourseofthession.

7.3 LIMITATIONS IN WORSTATION OPERATING SYSTEMS

Sincemanyworkstationoperatingsystems(suchasUNIX)aredesignedtoofferfairuseofsystemresourcesamongcompetingprograms,itisnoteasytomaintaintimelysystemresponsetocontinuousmediaapplications.Thesystemdoesnothaveanymechanismtomaintainaspecified QOS levelforsuchapplications. Similarlythecurrentsystemsdonotprovideanyoverloadpreventionormanagementschemewhenthesystemencountersatransientoverloadsituation.Everyprogram,then,willslowdownandausermayend upwithunpredictabledelayandjitterproblems.

Wealsofaceaproblemwithlackofreal-timeservices.Forinstance,fromaprogrammingpointofview,UNIX's"process"abstractionisaveryusefulconcept;however,itscontextswitchingcostsareunacceptableformanycontinuousmediaapplications.Itoftenforcesustochooseauser-level"thread"packagethatprovidesmuchlightercontextswitching.Threadpackagesoften alsohaveashortcoming,namelythelackofthefirst-classtreatmentofuser-levelthreads,includingthelackofsupportforpreemp-tablethreads.

7.3.1 InterruptLatencyandThroughput

Thelargeinterruptlatencyisanotherreasonthat UNIX doesnotsupportreal-timeactivitieswell.Incontinuousmediaapplicationssuchasthehandlingofa MIDI datastream,theapplicationproducesafrequentnumberofinterruptstothekernel.Theresultingheavycontextswitchingcannotkeepupwiththehighlydemandingeventstream.

LiketheUNIXkernel,allnon-preemptablekernelsprovideverypoorinterruptlatency.Processingofareal-timeeventthatwakesupanewthreadmaybedeferreduntilthecurrentprocessingofthekernelprimitivecompletes.Itmaytakeaslongasseveralmillisecondsintheworstcase.

Therearethreeapproachestoreducingkernelinterruptlatency.Thefirstapproachistomakethekernelhighlypreemptablebychangingitsinternalstructure.Thesecondschemeistoaddasetofsafepreemptionpointstotheexistingkernel.Thethirdschemeistoconvertthecurrentkerneltobeoneoftheuserprogramsandrunitontopofamicrokernel.Thischapterdiscussesthethirdapproach.

7.3.2PriorityInversionManagement

Whenareal-timeprogramsharesthesameresourceinthesystemwithnon-real-timeprograms,therearecaseswherethereal-timeprogrammustwaitforthecompletionofthenon-real-timeprograms.Infact,ifmanyapplicationssharethesamesystemserver,suchasanetworkserver,thenahigh-prioritytask'svideostream'spacketmustwaitforthecompletionofallpreviouslyqueuedlow-prioritypackets.

Thistypeofpriorityinversioncausedbythenon-preemptiveserveroftencausesunpredictabledelayandjitterproblems.Thestructureoftheserver,whichisshownbelow,willcausetheproblem,sincethebodyoftheserver(i.e., do_service)issimplyexecutedonemessageatatimeandthemessagequeuingisdoneinaFIFOorder.

```
1.  server() {
2.  while (1) {
```

```
3.    receive_msg();
4.    do_service();
5.    reply();
6.    }
7.    }
```

Toavoidpriorityinversioninaclient/serverenvironment,ataminimum,betteroperatingsystemsupportisneeded.Priorityinheritanceandpriorityhand-offmechanismsaretwotechniquesthatareusedinreal-timeoperatingsystems.

7.3.3PeriodicActivity

Continuousmediarequireperiodicserviceactivitiesfortransmissionorpresentation.Thesimplestwayofmanipulatingaperiodicdatastreamistousealoopconstructforhandlingthesequenceofdataimagesasfollows.

```
1.    while(1) {
2.    get_cm_object (...);
3.    draw_cm_object(...) ;
4.    }
```

Inthisscheme,theuserdoesnothaveanycontroloverthedrawingrateofthecontinuousmediaobjects.Inamulti-programmedenvironment,thedrawingrate(i.e.,temporalresolution)mayvarydependingontheworkloadofthesystem.Whenthesystemencountersanoverloadsituation,thedrawingmaystopmomentarily.Whentheothertaskterminates,thedrawingmayevenspeedup.Werefertothistypeofcontrolschemeasan*implicitbinding*ofatimingconstraintonthecode.

Ontheotherhand,inatraditionaloperatingsystemlikeUNIX,wecanprovidemoreexplicittimingcontrolbyusingasleepfunctionforregulatingthedrawingrateasfollows.

```
1.  start time = get_current_time();
2.  while(l){
3.  get_cm_object(. . .) ;
4.  draw_cm_object(. . .) ;
5.  start_time = start_time + period;
6.  duration = start_time - get_current_time();
7.  sleep(duration);
8.  )
```

Inthisscheme,iftheexecutionoftheprogramissuspendedafterget_cm_object()is evaluatedandtheexecutionislaterresumed,durationiscalculatedwiththeincorrec tvalueofthecurrenttime.So,theprogrammightbedelayedtoolonginthesleepostat ement.Asshowninthisexample,thepreemptionoftheprogramofteninterfereswitht hetimingcontrollogic.Itiswellknownthatthistypeofrelativetemporalsynchronizati onisapoormechanisminamulti-programmedenvironment.Eitherabsolutetimespecificationoroperatingsystems upportforperiodicityisnecessarytoovercomethisproblem.

7.3.4DeadlineandRecoveryManagement

Unlikehardreal-timesystems,manycontinuousmediaapplicationprogramshaveinherentlysoftde adlines.Forinstance,wearetypicallyabletocontinueavideoconferenceevenifthe mostrecentvideoimagecouldnotbeprocessedintime.Missingasingledeadlinema ynotcausetotalchaos;however,thedeadline-missednotificationtotheapplicationi simportantinformation.Basedonthisinformation,theapplicationsmaywanttochan getheQOSlevelforvideooraudiosessions.

Thebasicissuesarerelatedtodeadlinecontrol,notificationandrecoveryschemest hesystemcansupport.Whentheprogrammissesadeadlineduetooverloadorhard ware/softwareerrors,theuserprogramshouldbeabletodecidethecounteractions.

Inordertodiscusstheissuesindeadlineandrecoverymanagement,letususeamodernreal-timeprogramminglanguage,suchasRTC++.InRTC++,asimpledeadlinehandler **q**canbeexpressedin**withintdosexceptq**asfollows.

```
1.  within (dead_line_duration) do {
2.  get_cm_object (. . .);
3.  draw_cm_object(. . .};
4.  } except {
5.  recovery_action;
6.  }
```

Inthisexample,therecoveryactionalwayswillberunningwiththesamepriorityofthemainactivity.However,itisoftennecessarytoexecutetheactioninahigherprioritythanthemainactivity.Sotherecoveryactionshouldtakeplaceafterbumpingupitspriorityfirst:

```
1.  within (dead_line_duration) do {
2.  get_cm_object(. . . );
3.  draw_cm_object(. . .);
4.  } except {
5.  bump_up_priority(high);
6.  recovery_action;
7.  }
```

However,again,duetoamultiprogrammingenvironment,beforeexecutingbump_up_priority (high),thistaskcouldbepreempted.Inotherwords,inordertoprovidesuchsemantics,weneedtotreattheexceptionnotificationandbump-upoperationasanatomicaction.

*7.3.5QOS*ManagementandAdmissionControl

QOSmanagementforcontinuousmediaapplicationscanbeclassifiedintotwotypesofQOScontrolschemes:***static***and***dynamic***.Withastaticcontrolsscheme,ausersimplyspecifiesaQOSlevelatsessioncreationtime.ThespecifiedQOSlevelwillbemaintainedduringthelifetimeofthesession.

Adynamiccontrolscheme,ontheotherhand,allowsthesystemorauserprogramtochangetheinitialQOSlevelduringthecourseofthesession.Itcanbeinitiatedintwoways:oneisfromaQOSmanagerwhentheavailabilityofsystemresourcesbecomesverylow,andtheotherisfromtheusertaskwhenitwantstodegradetheinitialQOSlevelgracefullyorimprovetheQOSlevel.Ingeneral,staticanddynamicQOScontrolschemesarenotwellsupportedinworkstationoperatingsystemsyet.

AsimpleskeletonofanadmissioncontrolfortheQOSmanagercanbeexpressedasfollows.

```
1.    qos_manager( )
2.    . . .
3.    accept_request( );
4.    switch(msg)
5.    case admission_test:
6.    estimate_resource_req( );
7.    buffer_check( );
8.    schedurability_check( );
9.    network_capacity_check( );
10.   if (is_request_acceptable)
11.   qos_level = determine_qos_init_level( );
12.   reply(requester, qos_level)
13.   ...
```

Inthisexample,theQOSmanagercanreturnaninitialvalueoftheQOSleveltotheclientifallresourcechecksscanbepassed.Currentsystems,however,donotprovideany

mechanismtoperformsuchadmissioncontrolforavoidingpotentialoverloadsituations.Aresourceenforcementmechanism,whichpreventsunexpectedexcessuseofprocessorcycles,isalsolackingforprocessorresources.

7.4 NEW OS SUPPORT

Manyworkstationsarenowofferingmultimediacomputingsupport;however,wedolackacomprehensivesoftwarestandardorcommonoperatingsystemsupportfunctions.Varioustypesofoperatingsystemsupporthavebeendiscussed,rangingfromaverysimplereal-timeschedulertosophisticatedmodificationtodevicedrivers.Althoughoperatingsystemsupportsalonecannotcreatedistributedcontinuousmediaapplications,fundamentalchangesareneededinthetraditionaloperatingsystems.

Newoperatingsystemsupportsforcontinuousmediaapplicationscanbeclassifiedintothreecategories:*architecturalsupport,resourcemanagementsupport,*and*programmingsupport.*Inthissection,we'willdescribethesecategoriesbasedonourexperienceonextendingReal-timeMachforcontinuousmediaapplications.

7.4.1ArchitecturalSupport

Thebasicstructureoftraditionaloperatingsystemshasbeendominatedbyamonolithickernelarchitecture.However,newtypesofapplicationssuchascontinuousmediaapplications,mobilecomputing,personaldigitalassistanceandwirelessnetworkingaredemandingnotonlyadvancedoperatingsystemservicesbutalsobetter,flexibleoperatingsystemsarchitectures.

7.4.2MicrokernelArchitecture

Amicrokernelarchitectureisbecomingverypopularamongthenextgenerationworkstationoperatingsystems.A*microkernel*isanoperatingsystemkernelwhichisonlyresponsibleformanipulatinglow-level(ormeta-level)systemresources,andisindependentfromanyspecificuser-

levelcomputationalparadigm.Examplesofsuchlow-levelsystemresourcesareaddressspaces,processorcycles,interrupt,andtrap-handlingmechanisms.

Inthemicrokernel-basedarchitecture,traditionalsystemcomponentssuchasfilesystemandnetwork protocolmodulesresideoutsideofthekernel.Asystemschedulercanalsobeoutsidethetraditionalkernel.

Therearesevеralwaysofplacingnewoperatingsystemfunctionsontopofthemicrokernel.Anewsystemservicecanberealizedasaservicebylibraryroutines(SL)byaserverorservers(SS),andbymicrokernelfunctions(SK).WhiletheSKschemecanprovideeasysharingofsystemresourcesbetweenanewsystemserviceandthemicrokernel,theSSandSLschemesprovidebetterextensibilityofthemicrokernel.However,trade-offamongtheseschemesmustbeexaminedcarefully.

7.4.3ResourceManagementsupport

Anewresourcemanagementschemeshouldbeadoptedtoprovidenecessarysystemresourcessuchasprocessorcycles,memoryandnetworkbandwidth,sothattheapplicationcanmaintaintherequestedQOSlevelwithoutencounteringunpredictabledelayandjitterwhilereproducingthevideodisplayandaudiosound.Wediscusseveralresourcemanagementtechniquesforcontinuousmediaapplications.

QOS-BasedResourceControl

Anewapproachistomanagecontinuousmediaapplicationsbasedontheir QOSrequirements.QOSincontinuousmediacanbeexpressedintermsoftemporalandspatialcharacteristics.Althoughthetemporalandspatialcharacteristicsofperiodicandaperiodicdatastreamsaremostlyapplication'dependent,thesystemmustmaintainusers'requestedQOSlevel.

Oneparadigmchangeisthatthesystemmayacceptauser'srequestif,andonlyif,there areenoughsystemresourcesavailabletomaintaintherequestedQOSlevels.ThesystemmaynegotiatewithapplicationprogramsorausertoreduceitsQOSlevelssothatitcanbeaccommodatedunderthecurrentresourceconstraint.Inotherwords,thesystemwillhaveanadmissioncontrolmechanismtoavoidunexpectedoverloadorinterferencetoongoingQOSguaranteedactivities.

Inadynamiccontrolscheme,auserprogramorthesystem'sQOSmanagermaychangetheinitialQOSlevelduringthecourseofthesessiohl[1]Forinstance,wecanshowthefollowingcaseasanexampleofdynamicQOSchangeintheprogram.

```
1.    main ( ) {
2.    main_thread_body;
3.    . . .
4.    session_create (qos_mgr, qos_req);
5.    . . .
6.    session_control(qosjmgr, qos_change);
7.    . . .
8.    }
9.    session_call_back_stub(session, qos level) {
10.   adjust qos(session, qos level);
11.   }
```

Aftercreatingasession(line4),theuserprogrammaybeabletosubmitarequestfordegradingtheinitialQOSlevelexplicitly(line6).Ontheotherhand,theQOSmanagermaybeabletoinvokeacall-backfunctionsuchassession_call_back_stub()(line9)forrestoringordegradingtheQOSoftheongoingsession.

Real-timeScheduler

Areal-timeschedulerofferbetterprocessorschedulingsupportamongtimeconstrained continuousmediaapplicationprograms.Thetraditionalreal-timeschedulingpolicy,suchasfixedpreemptive(FP),ratemonotonic(RM),andearliestdeadlinefirst(EDF)shouldbeincorporatedwiththeadmissioncontrolandenforcementmechanismfortheQOS-basedparadigm.Anenhancedversionofthereal-timeschedulershouldbeabletocontrolandenforceaminimumandmaximumexecutionrateofcontinuousmediaapplicationsinordertomaintaintherequestedQOSlevelduringthecourseofthesession.

Thenotionof*schedulabilityanalysis*isimportantinhard/softreal-timeisystems,whereastheQOS-basedsystemrequires*playabilityanalysis*underJtherangeofthegivenQOSlevels.

QOS-BasedMemoryManagement

Intraditionalreal-timesystems,managementavoidsusingvirtualmemorywithademandpagingschemesinceitmaynotboundtheworst-casepaging-in/outtime.However,thevolumeofcontinuousmediaobjects(suchasdigitalvideoobjects)isverylarge,andwithoutusingakindofsharedmemorymanagementscheme,thesystemwillbeveryslowduetoexcessamountofdatacopying.

InaQOS-basedapproach,thesystemshouldbeabletoprefetchthecontinuousmediaobjectsinatimelyfashion.Itiseffectivetousesuchtime-drivenprefetchingpolicy;however,itisverydifficulttoaccesstherightpagesinthecontinuousmediaobjectifauserstartsrequestingavideoframerandomlybypointingatitslocationinthevideostream.

TimedI/O,Management

Theprimaryconcernofreal-timesystemdesignersisimprovingtheprocessingspeedofincomingI/Oevents.However,forcontinuousmediaapplications,thetemporalcorrectnessoftheI/Orequestisveryimportantsincethesystemmust,supportthesynchronizationofmultipledatastreams,suchasaudioand.videosessions.

Oneapproachistoattachatimestampatwhichtheactualprocessingoftheı/Odatashouldtakeplace.Forinstance,inReal-timeMach,ouraudiodrivercanacceptarequestsuchasplayingaudiodataattime*t.* Inthisway,thedrivercanpreprocessthedatabeforetime*t,*thenwecanreducethedelayatthedriverlevel.

7.4.4ProgrammingSupport

Traditionalabstractionofconcurrentprocessesisveryusefulforcontinuousmediaapplications.However,additionalfeaturessuchaspropertimemanagement,real--timethreads,synchronization,andreal-timeIPCareoftenmissing.Inthissection,wediscussthesepprogrammingsupportsinReal--TimeMach.

Real-TimeThreads

Areal-timethreadinReal-TimeMachcanbecreatedandkilledusingthertandthreadsystemcalls.Unlikenonreal-timethreads,areal-timethreadisdefinedwithitstimingconstraint.AsshowninaC-likepseudo-languageinthefollowingexample,areal-timethread,f()iscreatedwithitsthreadattributes{f, Si, Ti, Di). findicatesitsthread'sfunctionf();Si, Ti,andDiindicatethreadf's starttime,period,anddeadline,respectively.

1. root ()

```
2.  {
3.  thread id f id;
4.  f_attr = {f,Si,Ti,Di};   /*set of thread atattribute of f */
5.  thread_create(f_id, f_attr); /* creating f () asa thread */
6.  }
7.  f(arg) {
8.  f's body
9.  }
```

Notethatifthreadfisperiodic,thenitwillautomaticallyrestart,orreincarnate,whenitreachestheendofitsfunctionbody.

DeadlineManagement

Whenathreadgeneratesabadmemoryreference,orcommitsafloatingpointerror,thisisseenasalogicalcorrectnesserrorandisflaggedbytheoperatingsystemasanexception.Areal-timethreadcanalsosufferfromerrorsintemporalcorrectness,or*timingfaults.*Atimingfaultisafailuremodewhicharisesfromafailuretomeetuser--specifiedtimingconstraints.

Toareal-timeprogramthiscanbeasdisastrousasabadmemoryreference.Forthisreasonweprovideaninterfacetocatchtimingfaultslikeotherexceptionsandtoallowuserstodisposeoftheminanapplication-specificfashion.

Forinstance,whenaperiodicthreadmissesadeadline,itsdeadlinehandlermustdecidewhetheritismeaningfultocontinueorwhetheritshouldsimplyabort.Furthermore,ifaperiodicthreadisdelayedmorethanoneadditionalperiod,itmustalsodecidewhetherthemainthreadshouldcatchupwithallskippedinstancesorsimplydiscardthem.

Theschedulingpriorityofthetimingfaulthandlerisanimportantissue.Ifweusealang uageconstruct,like**within(t)**dos**except**q,thentheexecutionof**q**wouldbetreateda sthesamepriorityass.However,itisoftennecessarytochange**q**'sprioritybasedont henatureofthetimingfault.

Wesupportbothmethodsbydecouplingthetimingfaultmechanismfromtheschedu lingcontextofthefaultingthread.Instead,atimingfaultcausesthefaultingthreadtob esuspendedandamessagetobesenttoauser-specifiedport.Aseparatethread,withuser-selectableschedulingprecedence,waitsonthisportandtakesactiononlywhenatim ingfaultoccurs.Belowisasimpleexampleofhowthisworks.

```
1.  root. { ) {
2.  f_attr = {f,Si,Ti,Di};          /*set thread attributeof f */;
3.  rf_attr = {rf, Si, Ti, Di)/* set threadattribute of rf */
4.  thread_create (rf_id, rf_attr); /* creating rfthread */
5.  thread_creat.e(f_id, f_at.tr); /*creating f
6.  thread */
7.  }
8.  f(arg) {
9.  f's body
10. }
11. rf(time, thread, message) {
12. wait_for_notification( );    /* waiting for adeadline-miss notice */
13. rf's body
14. }
```

Inthisexample,therf threadiscreatedbeforethemainthreadstarts,anditimmediatelywaitsforatimernoti ficationindicatingthatthreadfhasmisseditsdeadline.Afterreceivingthenotification ,itcanexecutetheproperrecoveryactionagainstthistimingfault.

Thismechanismeasilygeneralizestypicalhardandsoftreal-timeresponses.Asoftreal-timeapplicationmightignoretheerrorbyexecutingthread_resume (f)whileahardreal-timeapplicationmightterminatetheoffendingthreadwiththread_terminate (f).

Real-timeSynchronization

Forreal-timesynchronizationsupport,thesystemshouldatleastprovideafasteventnotificationmechanismandareal-timemutualexclusionmechanism.Forbothmechanisms,thequeuingpolicyforwaitingthreadsshouldbebasedontheirpriorityinsteadofaFIFOordering.Traditional(non-real-time)synchronizationprimitivesuseaFIFO-basedqueuingforavoidingthestarvationproblemamongwaitingthreads.However,forreal-timeprograms,FIFOorderingoftencausesthepriorityinversionproblem.Onewayofavoidingthepriorityinversionproblemistousepriority--basedqueuingwiththepriorityinheritanceprotocol.

InReal-timeMach,forexample,acriticalsectioncanbeimplementedbyusingthefollowingrt_mutex_lockandrt_mutex_unlockprimitives[13].

```
1.  mutex_attr.mutex_policy = PRI_BPI  /* setting a mutex policy */
2.  ret = rt_mutex_allocate( mutex, mutex_attr)                    /*
    allocating a mutex variable */
3.  ret = rt_mutex_lock(mutex) ;/* body of critical section */
4.  ret = rt_mutex_unlock(mutex);
```

Asshowninline1,ausercanchooseasynchronizationpolicyforacriticalsectionbys ettinganattributeofthemutexvariable.InReal-timeMach,kernelizedmonitor(KM),basicpolicy(BP),basicpriorityinheritanceprot ocol(BPI),priorityceilingprotocol(PCP),andrestartablecriticalregion(RCS)aresu pported.

IntheKMprotocol,ifathreadentersthekernelizedmonitorregion,allpreemptionispr evented.Thus,thedurationofthecriticalsectionmustbeshorterthananyreal-timeth read'sdeadline.TheBPpolicy,ontheotherhand,simplyenqueueswaitingthreadsi nthelockvariablebasedonthethread'spriority.BPIprovidestheinheritingfunctiont hatalower-prioritythreadexecutingthecriticalsectioninheritsthepriorityofthehigher-priorit ythreadwhenthelockisconflicted.InthePCPprotocol,theceilingpriorityofthe lockisdefinedastheprirorityofthehighestprioritythreadthatmaylockthelockvariabl e.

TheunderlyingideaofPCPistoensurethatwhenathread*T*preemptsthecriticalsecti onofanotherthread*S*andexecutesitsowncriticalsection*CS,*thepriorityatwhich*CS* willbeexecutedisguaranteedtobehigherthantheinheritedprioritiesofallthepre-emptedcriticalsections.

FortheRCSpolicy,ahigher-prioritythreadisabletoabortthelower-priorityth readinthecriticalsectionandputitbacktothewaitingqueuewhilerecoverin gthestateofthesharedvariable.Afterthisrecoveryaction,thehigher-priorityth readcanenterthecriticalsectionwithoutanywaitinginthequeue.Auserpro grammustberesponsibletorecoverthestateofsharedvariable.

7.4.5Real-timeIPC

Similartotheoriginalsynchronizationprimitives,thequeuingpolicyofthemessagei sinaFIFOordering.Thus,ausercannotavoidtheprirorityinversionproblemwithinan on-pre-emptiveserver.InReal-

timeMach,wehaveextendedtheoriginalIPCbyprovidingpriority-basedqueuing,priorityhand-off,andpriorityinheritancemechanisms.Theprogramminginterfaceisalmostidenticaltotheoriginal IPCexceptthatproperportattributesmustbesetwhenitisallocated.Thefollowingattributesforthecommunicationportareprovided:

- **MessageQueuing:** Itspecifiesthemessagequeueordering.FIFOandpriority--basedorderingpoliciescanbeused.

- **PriorityHand-off:** Priorityhand-offmanipulatesthereceiver'spriority whenamessageistransferred.Ifthisattributeisset,thepriorityofthereceiverispropagatedfromthatofthesenderorgivenpriorityaccordingtotheselectedpolicy.Ifitisdisabled,thepriorityofthereceiverisnotchanged.

- **PriorityInheritance:** Itexecutestheprorityinheritanceprotocol.Ifitisactivated,theserverinheritsthepriorityofthesenderthreadwhichsentthehighestprioritymessage.

- **MessageDistribution:** Itselectsaproperreceiverthreadwhentwoormorereceiversarerunning.Arbitraryorpriority-based(WORK)selectioncanbespecifiedasapolicy.Whenthearbitrarypolicyisspecified,areceiverthreadischoseninFIFOorder.Thepriority-basedpolicyselectsareceiverthreadaccordingtoagivenpriority.

Sincetheserver'sservicetimeseemsmuchlongerthanacriticalsection,ausermustbeconcernedwiththesepriorityinversionproblemsindistributedmultimediaapplications.Inparticular,wehaveimplementedanintegratedpriorityinheritancemechanismbetweenthesynchronizationandIPCdomains.

7.5EXPERIMENTS USING REAL-TIME MACHS

Processor Reservations - Creating and Destroying Reservations

This program demonstrates how to create and terminate reserves, bind them to threads, and determine what reserve is bound to a particular thread. When run, the program creates a reserve, sets its name to “Example,” and requests that it have a reserve of 10 milliseconds out of every 50 milliseconds. Next it shows the reserve that it is currently running against, binds itself to run against this new reserve, again shows the name of the reserve that it’s running against, destroys the reserve, and one last time, shows the name of the reserve that it’s running against (demonstrating that a thread will be re-bound to the default reserve automatically when its reserve is terminated).

Program Components

- lines 14-42 show the name of the reserve the given thread is bound to
- lines 24-29 get the reserve the given thread is bound to
- lines 34-39 retrieve the name of this reserve
- lines 53-59 set the scheduling policy to SCHED POLICY RESERVES
- lines 62-67 retrieve the default processor set
- lines 72-73 a start time of zero represents an immediate start
- lines 78-79 we will request 10 milliseconds of compute time
- lines 84-85 we will request a 50 millisecond period
- lines 87-92 create a processor reserve. Note that when a reserve is first created, it does not actually have a reservation until we call reserve request()
- lines 97-103 associate a logical name with our new reserve
- lines 108-113 actually request a reservation of 10 milliseconds out of every 50 milliseconds starting immediately

- line 115 get our thread
- line 118 show the name of the reserve our thread is running against
- lines 123-128 bind our thread to the newly created reserve
- line 131 show the name of the reserve our thread is running against
- lines 136-141 terminate (destroy) this new reserve

```
11 #include <rt/mach_reserves.h>
12 #include <rt/sched_policy.h>
13
14 void show_my_reserve(thread)
15         thread_t thread;
16 {
17         mach_reserve_name_t myreserve_name;
18         mach_reserve_t myreserve;
19         kern_return_t ret;
20
21         /*
22          * What reserve are we running against?
23          */
24         ret = thread_get_reserve(thread,  &myreserve);
25         if (ret != KERN_SUCCESS) {
26                 printf("thread_get_reserve failed with %s\n
27                                 mach_error_string(ret));
28                 exit(1);
29         }
30
31         /*
32          * Get the name of this reserve
33          */
34         ret = reserve_name(myreserve, myreserve_name);
35         if (ret != KERN_SUCCESS) {
36                 printf("reserve_name failed with %s\n",
37                                 mach_error_string(ret));
38                 exit(1);
39         }
40
41         printf("Now running against reserve %s\n", myreserv
42 }
43
44 main()
45 {
46         mach_reserve_name_t reserve_name;
```

```
66                      exit(1);
67              }
68
69              /*
70               * set start time to zero (immediate)
71               */
72              start.seconds = 0;
73              start.nanoseconds = 0;
74
75              /*
76               * computation time of reserve
77               */
78              compute.seconds = 0;
79              compute.nanoseconds = 10000000; /* 10 ms */
80
81              /*
82               * period of reserve
83               */
84              period.seconds = 0;
85              period.nanoseconds = 50000000; /* 50 ms */
86
87              ret = reserve_create(psetname, &reserve);
88              if (ret != KERN_SUCCESS) {
89                      printf("reserve_create failed with %s\n",
90                                      mach_error_string(ret));
91                      exit(1);
92              }
93
94              /*
95               * Set a name for our new reserve
96               */
97              strcpy(reserve_name, "Example");
98              ret = reserve_set_name(reserve, reserve_name);
99              if (ret != KERN_SUCCESS) {
100                     printf("reserve_set_name failed with %s\n",
101                                     mach_error_string(ret));
102                     exit(1);
103             }
```

```
114
115         thread = mach_thread_self();
116
117         /* show our reserve */
118         show_my_reserve(thread);
119
120         /*
121          * Bind our thread to this new reserve
122          */
123         ret = thread_set_reserve(thread, reserve);
124         if (ret != KERN_SUCCESS) {
125                 printf("thread_set_reserve failed with %s\n",
126                                 mach_error_string(ret));
127                 exit(1);
128         }
129
130         /* show our reserve */
131         show_my_reserve(thread);
132
133         /*
134          * Destroy the reserve
135          */
136         ret = reserve_terminate(reserve);
137         if (ret != KERN_SUCCESS) {
138                 printf("reserve_terminate failed with %s\n",
139                                 mach_error_string(ret));
140                 exit(1);
141         }
142
143         /* show our reserve */
144         show_my_reserve(thread);
145 }
```

7.6 SUMMARY

- There are three approaches to reducing kernel interrupt latency:
    - make the kernel highly preemptable by changing its internal structure
    - add a set of safe preemption points to the existing kernel
    - convert the current kernel to be one of the user programs and run it on top of a microkernel

- When the system encounters an overload situation, the drawing may stop momentarily. When the other task terminates, the drawing may even

speed up. We refer to this type of control scheme as an *implicit binding* of a timing constraint on the code

- QOS management for continuous media applications can be classified into two types of QOS control schemes: ***static*** and ***dynamic***

- With a static controls scheme, the user specifies a QOS level at session creation time. The specified QOS level will be maintained during the lifetime of the session.

- A dynamic control schemeallows the system or a user program to change the initial QOS level during the course of the session

- Thereareseveralwaysofplacingnewoperatingsystemfunctionsontopofthe microkernel.A new system service can be realized as a service by library routines (SL) byaserverorservers(SS),andbymicrokernelfunctions(SK)

- Intraditional real-time systems,managementavoidsusingvirtualmemorywitha demand paging scheme since it may not bound the worst-case paging-in/out time

- **Message Queuing**specifies the message queue ordering where FIFO and priority--based ordering policies can be used.

- **Priority Hand-off** manipulates the receiver's priority when a message is transferred.

- **PriorityInheritance:**Itexecutestheprorityinheritanceprotocol.

- **Message Distribution:** It selects a proper receiver thread when two or more receiversarerunning.

7.7Unit End Exercises

1. What are the limitations in the workstation operating system?
2. Explain QOS Management and Admission Control.
3. How are new operating system supporting continuous media applications?
4. What are the attributes for communication port?

7.7 Additional Reference

- Principles of Multimedia, Eighth reprint edition 2009, Ranjan Parekh, Tata McGraw-Hill Companies

- Real-Time Mach 3.0 User Reference Manual - CiteSeerXhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.5147&rep=rep1&type=pdf

8

Middleware System Services Architecture

Unit Structure

Objectives
Introduction
Goals of Multimedia System Services
Multimedia System Services Architecture
Media Stream Pool
Summary
Unit End Exercises
Additional Reference

OBJECTIVES

In this Chapter you will understand:

- Multimedia System Services architecture proposed jointly by Hewlett-Packard (H-P), International Business Machines (IBM), and SunSoft
- Concepts key to the Multimedia System Services

INTRODUCTION

Since the emergence of multimedia applications more than a decade ago, applications such as streaming media players, distributed games, and online virtual worlds have become commonplace today. Multimedia applications access a combination of audio, video, images and textual data and have timeliness constraints. Until recently, the demanding computing and storage requirements of these applications as well as their soft real-time nature necessitated the use of specialized hardware and software. For instance, continuous media servers were used to stream audio and video instead of

general-purpose file servers, while audio-video playback required the use of specialized hardware decoders.

GOALS OF MULTIMEDIA SYSTEM SERVICES

The primary goal of the Multimedia System Services is to provide an infrastructure for building multimedia computing platforms that support interactive multimedia applications dealing with synchronized, time-based media in a heterogeneous distributed environment. One can truthfully say that there are several products today that provide multimedia services that support interactive applications which deal with synchronized, time-based media. Most existing systems, however, operate only in a stand-alone environment. Thus, the major distinction of the Multimedia System Services is the ability to support such applications in a heterogeneous, distributed environment.

The Multimedia System Services is intended to address a broad range of application needs. It extends the multimedia capabilities of today's standalone computers to capabilities that are usable both locally and remotely. The Multimedia System Services gives applications the ability to:

- handle live data remotely
- handle stored data remotely
- handle both live and stored data simultaneously
- handle multiple kinds of data simultaneously
- handle new kinds of devices and media types

To provide support for remote media device control and remote media access that derive from the above application scenarios, the Multimedia System Services uses two distinct mechanisms. To support interaction with remote objects, the Multimedia System Services depends upon the Object Management Group's (OMG) Common Object Request Broker Architecture

(CORBA). To support the media-independent streaming of time-critical data, the Multimedia System Services defines a Media Stream Protocol.

Summary of Multimedia System Services Functions

The Multimedia System Services is designed to satisfy the requirements put forth in the IMA Multimedia System Services Request for Technology and is broadly constrained by that document. As such, the Multimedia System Services constitutes a framework of "middleware" - system software components lying in the region between the generic operating system and specific applications. As middleware, the Multimedia System Services marshals lower-level system resources to the task of supporting multimedia processing, providing a set of common services which can be used by multimedia application developers on an industry-wide basis.

The Multimedia System Services encompasses the following characteristics:

- provision of an abstract interface for a media processing node, extensible through sub-classing to support abstractions of real media processing hardware or software;
- provision of an abstract interface for the data flow path or the connection between media processing nodes, encapsulating low-level connection and transport semantics; grouping of multiple processing nodes and connections into a single unit for purposes of resource reservation and stream control;
- provision of a media data flow abstraction, with support for a variety of position, time, and/or synchronization capabilities;
- separation of the media format abstractions from the data flow abstraction;
- synchronous exceptions and asynchronous events;
- application visible characterization of object capabilities;
- registration of objects in a distributed environment by location and capabilities;

- retrieval of objects in a distributed environment by location and constraints;
- definition of a Media Stream Protocol to support media-independent transport and synchronization;
- use of industry standard CORBA technology as the basis for supporting distributed objects;
- provision of a local library to simplify the task of writing Multimedia System Services-based applications.

MULTIMEDIA SYSTEM SERVICES ARCHITECTURE

The next few pages present several comprehensive views of the Multimedia System Services, which, taken together, represent a broad, architectural summary. These views include:

- an Object interaction diagram, to characterize the dynamic relation. ships among instantiated objects and to illustrate client-visible inter. faces;
- an interface inheritance diagram, to describe the inheritance hierarchy among IDL interfaces; and
- a discussion of a typical case

Object Framework

Figure 8.1 summarizes the interactions between Multimedia System Services framework objects and the client; Figure 8.2 summarizes the interaction among framework objects. As seen in Figure 8.1, only a subset of the objects and interfaces are actually visible to a Multimedia System Services client. In particular, much of the interaction between the virtual connection and other objects in the framework is not client visible. This specification is concerned primarily with client-visible interfaces.

Figure 8.1 is suggestive, rather than realistic: The objects shown are instances of abstract classes, rather than concrete classes which would normally be instantiated. Also, object creation and destruction are not shown in this diagram.

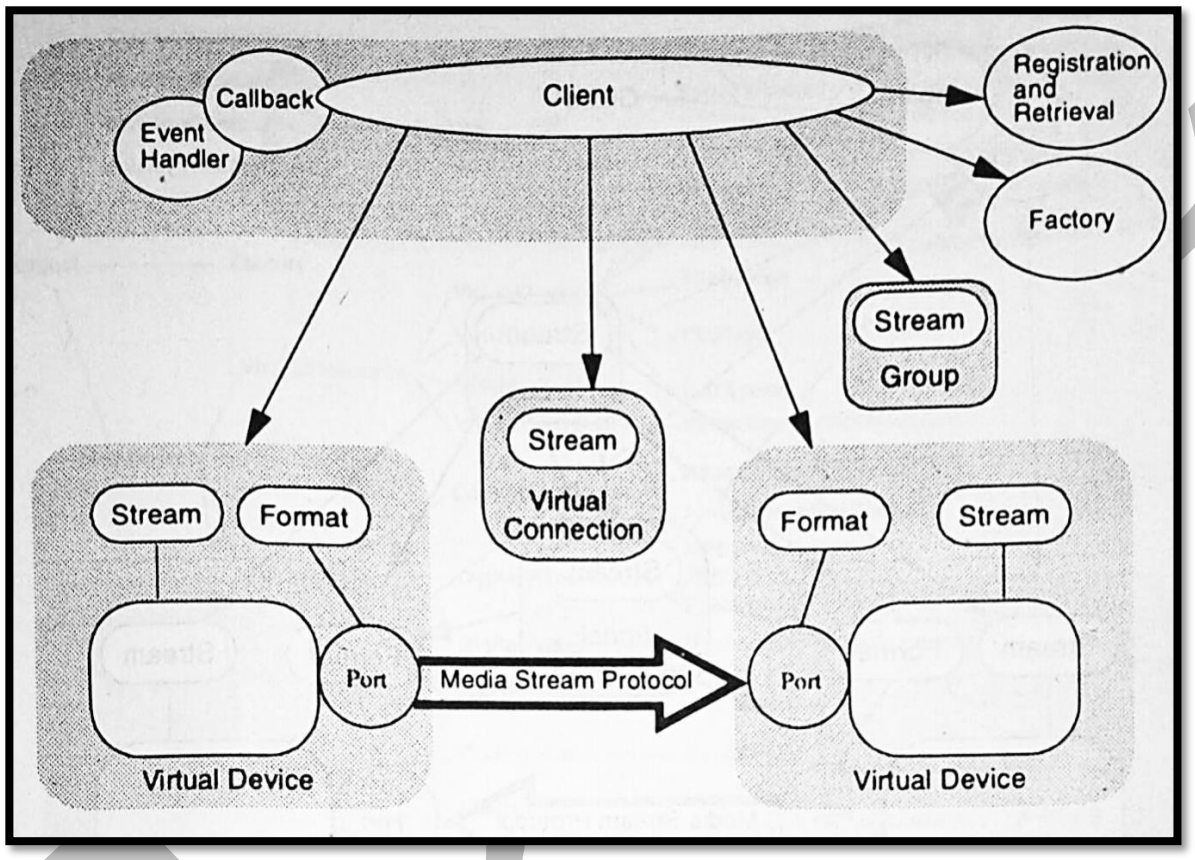


Figure 8.1: Multimedia system services client interaction

In Figure 8.1, the client is communicating with a small data flow graph, comprised of two virtual devices and a virtual connection. A group object, which assists the client, is also shown. The client interacts with the objects indicated by the arrows. Each of these interfaces may be local or remote.

Each virtual device is a processing node in the data flow graph. The nature of the processing (capture, encoding, filtering, etc.) varies according to the specific object (and is implemented by subclassing). Associated with each virtual device is a stream object and one or more format objects shown by the boxes in the shaded areas. Virtual connections and groups also have an

associated stream object and this association is represented similarly. These associations are referred to as inclusion. Although explicitly shown in the diagram, the client interacts directly with the included stream and format interfaces.

A stream object provides the client with an interface to observe media stream position in various terms (as a function of media transport, media samples, or logical time). Some stream objects also provide an interface for controlling the flow of media data in a media stream and some stream objects provide synchronization interfaces.

In addition to a stream, a virtual device also contains one or more ports, describing an input or output mechanism for the virtual device. Ports are framework objects that do not have a client-visible interface; in the diagram, they are shaded to indicate this. Virtual devices do provide an interface to select a specific port, using an index as opaque handle.

Just as the stream object allows a media stream control abstraction, which is separable from media processing, the format object provides an abstraction of the details of media formatting, which is separate from both processing and flow control. For example, the details of a frame-dependent video encoding like MPEG, would be represented by a subclass of format.

The virtual connection provides an interface to create a connection between an output port of one virtual device and an input port of another, fully encapsulating low-level transport semantics. Virtual connections also provide support for multicast connections. An included stream object provides an interface for controlling the data flow on the virtual connection.

The group object, shown in Figure 9.1, provides assistance to the client to manage the data flow graph of the two virtual devices and the virtual connection.

A group object provides a convenient mechanism for atomic resource allocation and specification of end-to-end Quality of Service (QOS) values for the whole graph. The group interface includes a stream object, through which the client can control data flow for the encapsulated graph.

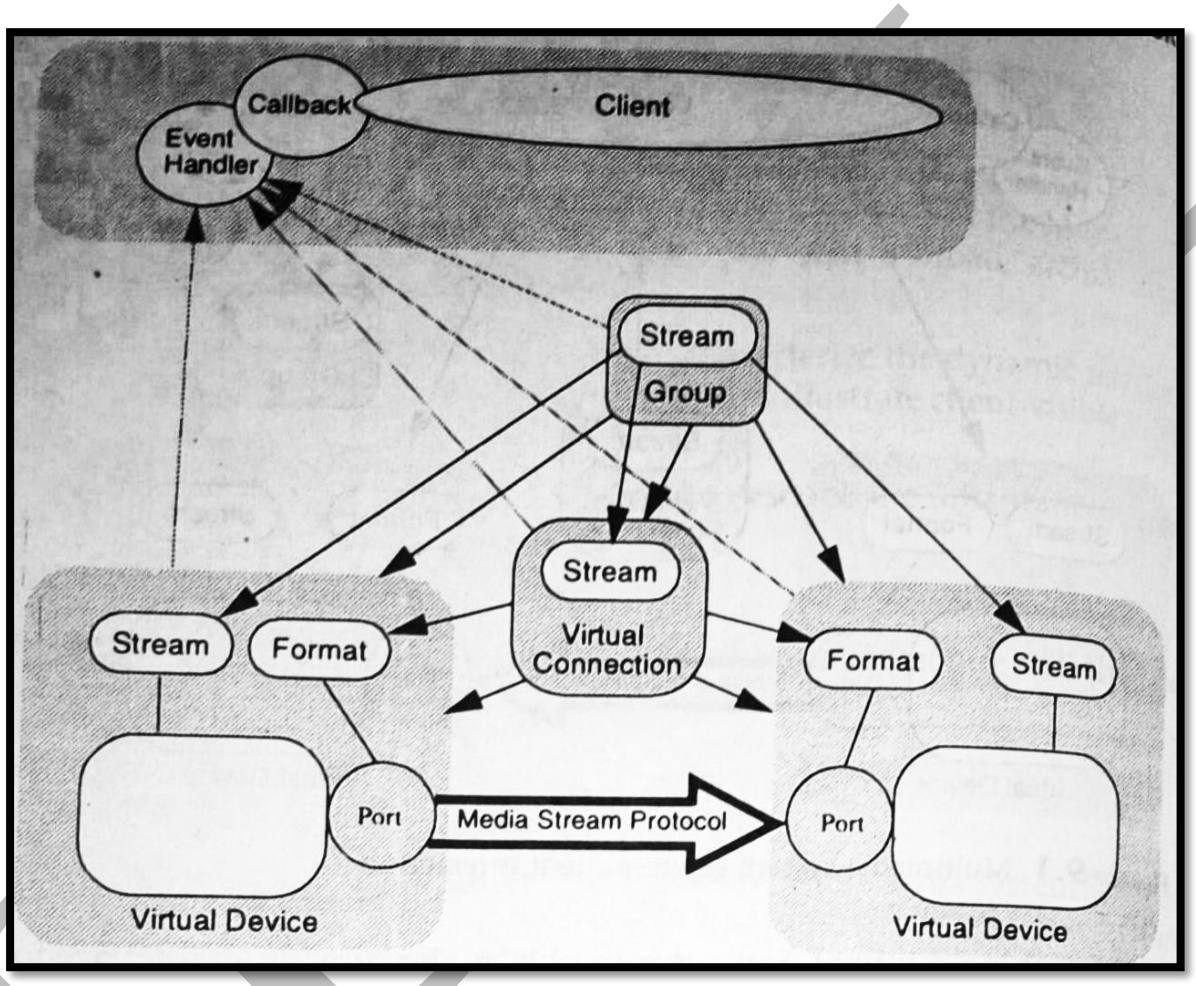


Figure 8.2: Multimedia system services internal interfaces

Multimedia System Services objects are instantiated by factories. A factory provides the client an interface to select among the various objects that the factory is capable of creating. A client can also use the Registration & Retrieval Service to find a reference to a factory capable of instantiating an object whose capabilities satisfy a list of constraints.

A client can register interest in receiving specific events produced by the various objects. The client can specify callback functions to handle these events received by an event handler.

Figure 8.2 shows the internal interfaces between Multimedia System Services objects. For the most part, the client is unaware that these interfaces exist, and this document will not focus on such interfaces. They are shown here to help explain the Multimedia System Services architecture.

The primary purpose of most of the internal interfaces is to off-load work from the client. Note, for example, that the virtual connection interacts with the formats of both the source and target virtual devices. This allows the virtual connection to match those formats without client intervention.

The group and the stream associated with the group provide similar assistance to the client; the group can assist in resource allocation, while the associated stream can assist in stream control. The shaded arrows show that the objects send events to the client via the event handler.

Interface Inheritance

Another view of the Multimedia System Services architecture is given in Figure 9.3. This is an inheritance diagram for the IDL interfaces in the Multimedia System Services. It can be read like a class diagram, but specifies only interface inheritance; a specific implementation may mimic this with a parallel class hierarchy, or it may use no inheritance at all.

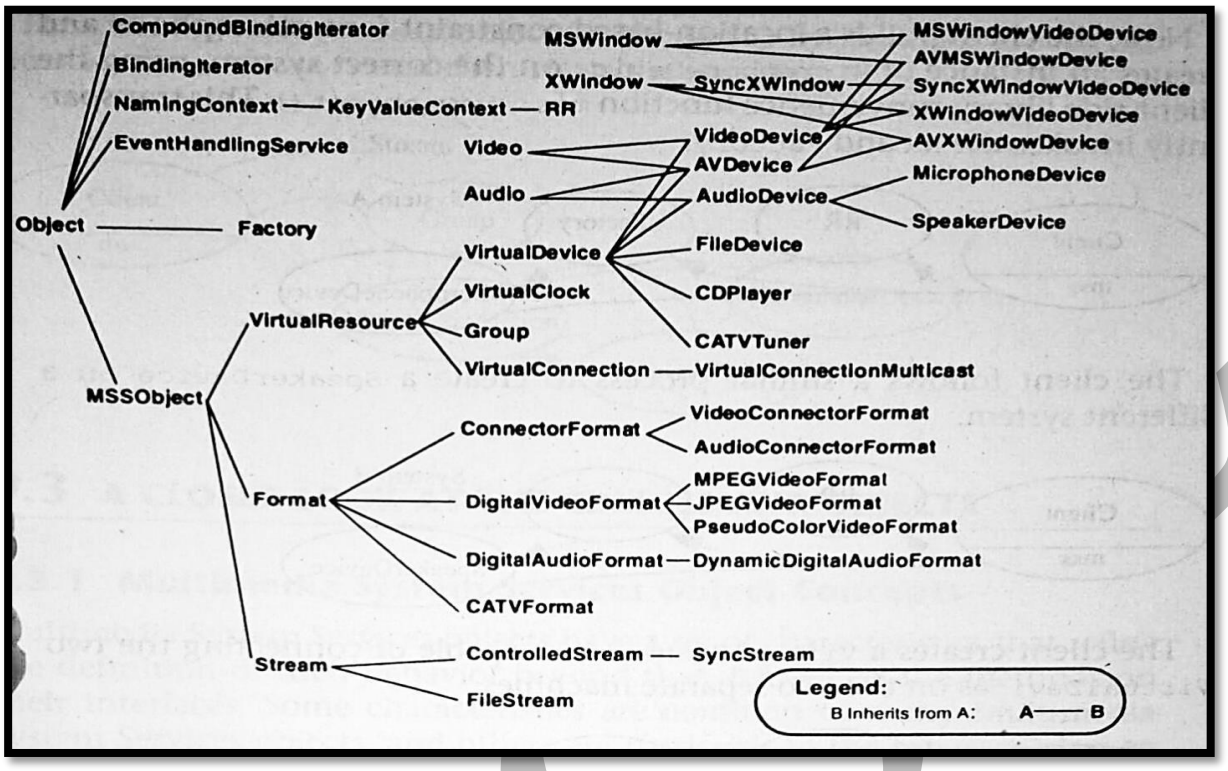


Figure 9.3: Interface Inheritance Diagram

MEDIA STREAM PROTOCOL

When a virtual connection determines that two virtual devices cannot be directly connected (often because they are on different machines), it creates a virtual connection adapter to transport media data between them. The virtual devices may reside in different implementations of the Multimedia System Services, so the virtual connection adapter must share a common protocol in order to interoperate.

The Multimedia System Services envisions one such protocol, the Multimedia System Services Media Stream Protocol (MSP), which runs over a number of network transports including NetBIOS, SPX/IPX, TCP(UDP)/IP, and RTP/ST-II. An interoperable implementation of the Multimedia System Services should provide a virtual connection adapter that implements the MSP over at least one network transport. Implementations are free to provide both additional transports for the MSP and complete alternatives to the MSP.

The purpose of the MSP is to convey media data along with the necessary information to do regulation, synchronization, and time-critical delivery of that data. To this end, the MSP defines a "media packet" that consists of media data and a media packet header (note that the media packet may be transmitted as any number of network packets).

The virtual connection determines the format of the media data, including its bit-level representation, by negotiation with the relevant Format objects and perhaps with client intervention. The MSP treats the media data as an opaque entity whose only visible attribute is its size. Neither the virtual connection adapter nor the underlying network transports know how to extract information from the media data; any information that is needed for data transport and regulation must be conveyed by the media packet header.

The MSP can convey the following information along with the opaque media data:

- timestamp (monotonically increasing value from the start of the stream),
- duration (for aperiodic media),
- priority (importance of this media packet relative to others in the same stream),
- dependency information. This allows virtual devices, virtual connections, or other non-framework objects to determine the structure of a stream, without knowledge of the media in the stream. An example use of this would be MPEG where the codes can be "I," "P," and "B." Given the dependency information one can decide that if for some reason (e.g., failures or lack of processing time) a "P" sample frame cannot be delivered, then the "B" frames are not useful and should not be delivered,
- in-stream events, which can include markers and errors related to the media data,
- sequence number,
- checksum (for transports that require it),

- length check-summed (for transports that require it
- length of the media (for transports that require it).

The checksum length is given as a separate field to allow checksumming of all the data, just the header, or no checksumming.

SUMMARY

- The primary goal of the Multimedia System Services is to provide an infrastructure for building multimedia computing platforms that support interactive multimedia applications dealing with synchronized, time-based media in a heterogeneous distributed environment

- To support interaction with remote objects, the Multimedia System Services depends upon the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA).

- To support the media-independent streaming of time-critical data, the Multimedia System Services defines a Media Stream Protocol

- As middleware, the Multimedia System Services marshals lower-level system resources to the task of supporting multimedia processing, providing a set of common services which can be used by multimedia application developers on an industry-wide basis

- The Multimedia System Services Media Stream Protocol (MSP), which runs over a number of network transports including NetBIOS, SPX/IPX, TCP(UDP)/IP, and RTP/ST-II

- The purpose of the MSP is to convey media data along with the necessary information to do regulation, synchronization, and time-critical delivery of that data

Unit End Exercises

1. What are the characteristics of Multimedia System Services?
2. What are the goals of Multimedia System Services?
3. Write a note on views of the Multimedia System Services.
4. Explain Multimedia system services internal interfaces.
5. What is Interface Inheritance?
6. What is media stream protocol?

Additional Reference

- Principles of Multimedia, Eighth reprint edition 2009, Ranjan Parekh, Tata McGraw-Hill Companies

- Middleware versus Native OS Support: Architectural Considerations for Supporting Multimedia Applications by Prashant Shenoy, Saif Hasan, Purushottam Kulkarni, Krithi Ramamritham available at http://lass.cs.umass.edu/papers/pdf/RTAS02.pdf

9

Multimedia Devices, Presentation Services, and The User Interface

Unit Structure

9.1 Objectives
9.2 Introduction
9.3 Client Control of Continuous Multimedia
9.4 Device Control
9.5Temporal Condition and Composition
9.6 Toolkits
9.7 Hyperapplications
9.8 Summary
9.9 Unit End Exercises
9.10 Additional Reference

9.1 OBJECTIVES

In this Chapter you will understand:

- Multimedia Services and the window system
- Stream Processing and synchronization of multiple streams
- Hyperapplications

9.2 INTRODUCTION

Multimedia applications differ from conventional applications in the use of new media types, complex temporal composition, Object hyperlinking and annotation, and external multimedia devices. The challenge for developers of services and toolkits for multimedia user interfaces is to provide an extensible and efficient architecture as well as the appropriate programming

abstractions. Concurrently, the application framework is evolving to include distributed object access, scripting languages, and multimedia interchange services. Integration with these additional facilities will require careful planning by application services designers. The state of knowledge regarding the use of multimedia in the user interface will grow dramatically as the technologies for constructing such interfaces become more pervasive.

A number of toolkits exist for constructing user interfaces composed of buttons, text entry, scrollable areas, and other conventional interactors. As the technology for adding multimedia data types to applications progresses, the following presentation-related services will be needed by application developers:

- Control of image presentation and continuous media streams such as digital audio and video
- Temporal composition and synchronization so that parallel and serial timing relations between different presentation steps can be easily expressed
- High-level control of multimedia peripherals and continuous media to hide device dependencies and low-level synchronization issues from the application
- Hyperlinking between two or more content objects to facilitate hypertext and hypermedia applications
- Support for new input technologies such as pen input and voice recognition
- Multimedia content interchange so that applications can exchange and share complex compositions in a heterogeneous environment
- Standardized multimedia-related interactors, such as a VCR-style panel, so that universal interaction paradigms will have a consistent look and feel

9.3 CLIENT CONTROL OF CONTINUOUS MULTIMEDIA

Presentation of individual media objects is a basic service that is typically provided as an extension to an existing graphical user interface toolkit. For image, animation, and video media, the application needs to be able to specify a viewport by which the object is presented and which may permit interactive zooming and panning. For continuous media, stream parameters such as position, direction, and rate are usually available to the application. The stream or track view of continuous media can be controlled using a player abstraction in the API.

For example, the following segment, based on the Fluency digital Object API, creates a combined audio and video stream and processesrequests to play, stop, and start recording.

```
/ • Create an audio-video stream * /
hwnd = DvoCreateAudioVideo (stream_name, x, y, w, h, …);

/ * Handle client request * /
switch (type) {
case IDM_PLAY:
      DvoP1ayForward (hwnd); break;
case IDM_STOP:
      DvoStop (hwnd); break;
case IDM_RECORD_START:
      DvoRecord (hwnd); break;
```

The player abstraction is a natural way to control video and audio media. Notice that the stream media is distinct from the player and can be considered an opaque type. This is consistent with the view of continuous media applications programming in which data and control are separate. This the stream not only data, frees but the it is application also more efficient program since from the overall details movement of buffering of data is reduced. Data copying of continuous media data transfers from one device to another is a significant performance problem for conventional operating

systems. Unless the application needs to process the continuous media data, client mediation of the data stream is unwarranted.

9.3.1 Stream Processing

For many cases, processing of the stream data is either unnecessary or can be handled by generic hardware or system software. For example, video compression levels, frame size, and frame rate are common hardware CODEC parameters. In other cases, in the absence of server support for processing operations, client-side processing is necessary.

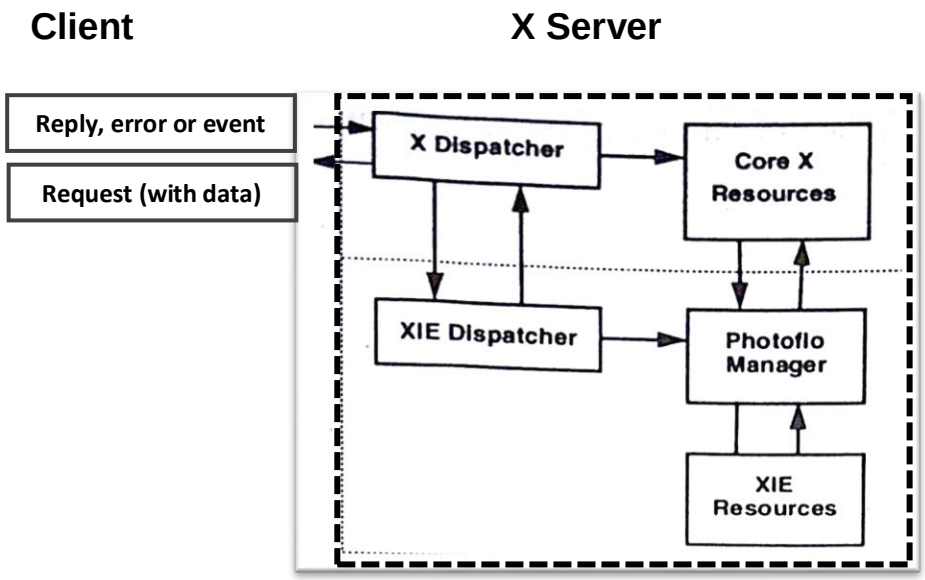


Figure 9.1 High-level view of X Server with X Imaging Extension

The X Imaging Extension (XIE) is an example of a server integrated processing system for image data. Figure 9.1 shows a high-level view of XIE. The Photoflo Manager is the computational engine which manages the image processing pipeline. The client controls the Photoflo Manager by specifying a directed acyclic graph (DAG), which represents the operations to be performed on the image. These operations include traditional image processing functions such as scaling, image arithmetic, convolution, filters,

and histograms. The client passes a compressed image stream to the server which carries out the operations specified in the Photoflo DAG. Since the client still handles the compressed image data in XIE, the benefits of separation of data and control are only partially achieved; however, only the server has to work with the uncompressed data, reducing client buffering requirements.

XIE is a server-based processing architecture for a specific media type. Such architectures for video and audio processing will likely result as research in these areas progresses.

9.3.2 Synchronization of Multiple Streams

In the Fluency API, an audio stream and a video stream are treated as a unit for synchronization purposes. A more general approach is needed to allow arbitrary numbers of streams to be controlled in synchrony. For example, ACME allows an arbitrary collection of streams to be grouped for synchronization. It implements such synchronization using a logical time system (LTS).

The ACME LTS is an independent clock with which various stream devices are associated. When the LTS is running, the clock value increases at the same rate as real time. Each stream is treated as a sequence of timestamped units. When the clock reaches time t, data units with timestamp t are processed by the associated devices.

An application creates an LTS and associates it with a device as follows:

```
LTS create_LTS (mode, param);
bind (LTS, logical_device, start_time, max_skew, start_count);
```

The mode of the LTS is used to specify the reference for the LTS clock; it can be device-driven, connection-driven, or n external timer. The bind operation associates the LIS with the given device. The start time is the LTS time when device operation begins. The LTS will stop if a device falls behind by

max_skew due to starvation. The start count is used to prime output devices so that a given amount of jitter can be tolerated.

Once the LIS is created and bound to the devices, the application controls progress using `start( )` and `stop( )` functions.

9.4 DEVICE CONTROL

Although the multimedia revolution is digital, the existing analog equipment base is still important for many applications. There are a large number of devices available today, which, under computer control, can be integrated with other desktop multimedia services. Available functions range from play and record to special effects and device-to-device connections. The technical problems of providing a device and media control service include:

- Identifying the correct device abstractions that cover many different device types
- Providing a model for device and media objects which permits applications to easily manipulate collections of such objects
- Controlling shared access to external devices

An example of the concepts involved in an architecture for device and media control, we present a system developed at the University of Massachusetts Lowell called DMCS (Distributed Media Control System). DMCS was developed to provide an interim base for applications development; it has also been used in a commercial video-on-demand test. Distinctive features of DMCS include:

- Support for hybrid media (analog and digital)
- A routing layer for circuit-switch audio-video networks
- A connection management layer for conferencing applications
- Virtual objects which can be aggregated and provide an application level view of media objects

A number of other device control environments have been developed like Galatea and the Touring Machine, the IMA Virtual Device model.

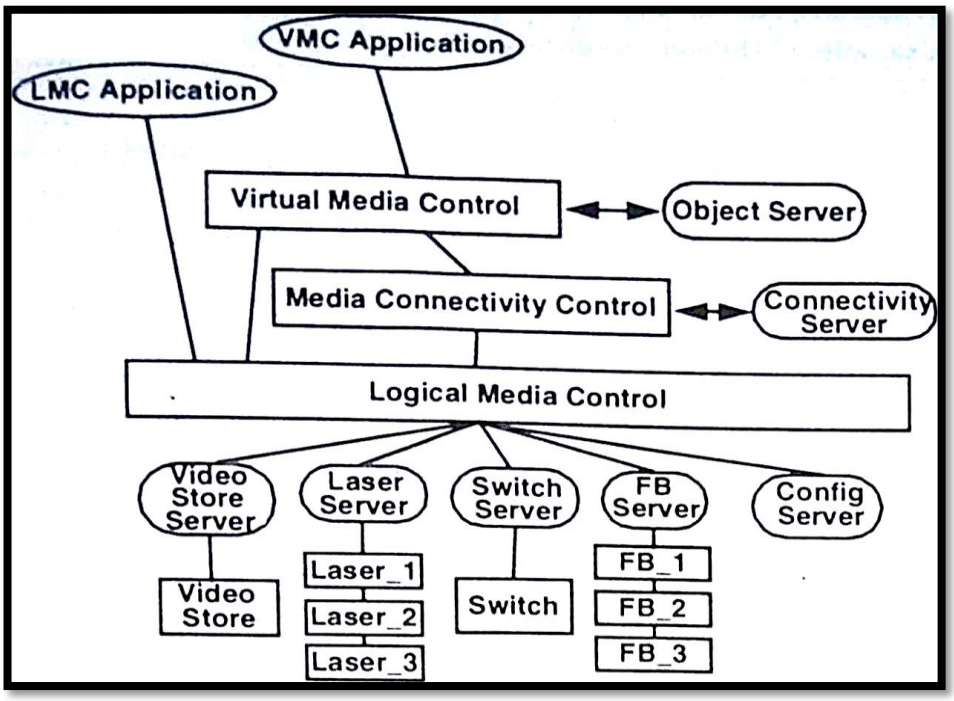


Figure 9.2: Distributed Media Control System layered architecture

DMCS consists of three layers (Figure 9.2) from lowest to highest. The Logical Media Control (LMC) layer provides a device-independent view of device functions. There are eight classes of devices, which include players, special effects, switches, speech generators, speech recognizers, and CODECs. The Media Connectivity Control (MCC) layer provides deviceconnectivity for circuit switched interconnections as might be available through CATV and interconnected video switches. The Virtual Media Control (VMC) layer has an API for creating and managing persistent objects which represent one or more LMC-level media types. VMC objects can be manipulated using play, record, switch, and other functions. Additionally, the VMC layer has facilities for supporting audio-video conferencing using the session abstraction.

There is no single abstract device model that captures the range of equipment that can be interfaced to computers and interconnected today. Representative categories of devices are shown in Table 9.1. Some devices have

combinations of functions and are typically decomposed into the corresponding logical classes.

The device control abstraction in DMCS allows the application to first (LMC layer) ignore any manufacturer-specific details within a device class and second (VMC layer) use an object API to create and manipulate virtual composite devices. The first goal is achieved by defining a logical view of eight categories of devices. These categories cover the current devices supported by DMCS, and each has a corresponding set of operations defined for it.

The second goal is implemented by providing a persistent object store in which virtual object definitions are created and stored. These virtual objects are referenced by the application by using a handle or name and are accessed by the VMC layer when a specific operation, such as playing an object, is performed.

9.5 TEMPORAL COORDINATION AND COMPOSITION

9.5.1 Temporal Coordination

Temporal Coordination is an activity with the objective to ensure that the distributed actions realising a collaborative activity takes place at an appropriate time, both in relation to the activity's other actions and in relation to other relevant sets of neighbour activities. Temporal coordination is mediated by temporal coordination artefacts and is shaped according to the temporal conditions of the collaborative activity and its surrounding socio-cultural context. This definition consists of three parts.

First, temporal coordination is an activity. The object of an activity can be another activity and temporal coordination is thus in itself an activity, which seeks to integrate distributed collaborative actions. The dynamic nature of any

activity implies that temporal coordination can be achieved both as an action within the overall collaborative activity and as an activity in itself directed towards another collaborative activity. In this sense coordination can be achieved both intrinsically within a group of collaborating actors sharing the same object of work – i.e. the actors organise and coordinate the actions themselves – and extrinsically to the group – i.e. the actions are organised and coordinated by someone outside the group. McGrath identifies three "macro-temporal levels" of collaborative work: *(i) synchronisation, (ii) scheduling, and (iii) time allocation:*

i. **Synchronisation** is an ad hoc effort aimed at ensuring that action "a", by person "i", occurs in a certain relation to the time when action "b" is done by person "j" according to the conditions of collaborative activity. Because synchronisation is tied to the conditions of the activity, synchronisation corresponds to the operational level of temporal coordination.
ii. **Scheduling** is to create a temporal plan by setting up temporal goals (i.e. deadlines) for when some event will occur or some product will be available, and is thus the anticipatory (action) level of temporal coordination.
iii. **Allocation** is to decide how much time is devoted to various activities. The essence of allocation is to assign resources according to the overall motives of the collaborative work setting and hence reflects a temporal priority according to different motives. Thus, allocation is the intentional (activity) level of temporal coordination.

Second, temporal coordination is mediated by artefacts. Coordinating activities in time is essentially to determine exactly when some event will occur or some results will be available in relation to other activities and actions. A particular effective way to do this is to establish starting times and deadlines according to some external and socially shared time measurement. Hence, a temporal artefact, such as the clock or the calendar, can be turned

into a temporal coordination artefact, mediating the temporal coordination, when shared within a collaborating community of practice.

Actually, Hutchins (1996) argues that “the only way humans have found to get such [socially distributed] tasks done well is to introduce machines that can provide a temporal meter and then coordinate the behaviour of the system with that meter”. Within hospitals the clock found in every hospital unit is “one of the major ‘collective representations’ of the sociotemporal order of the hospital”because it represents the official time according to which all activities are recorded and synchronised.

However, psychological temporal artefacts are important mediators of temporal coordination as well. The notion of time as a psychological faculty of the mind goes back to Kant, who viewed time as a category which is logically prior to the individual’s construction of reality, and without which reality cannot even be meaningfully experienced.

Following this line of thought, Durkheim argued that the notion of time originated not in the individual but in the group, arguing for a social origin and nature of temporality. Taken together, these two ideas give us a dialectical understanding of temporality as a cognitive structure that is shaped, developed and defined within a cultural-historical context. Therefore, the temporal reference frames in which we perceive, measure, conceptualise, and talk about temporality are cultural-historical developed and defined artefacts.

Such temporal reference frames play a crucial role in enabling and mediating temporal coordination. Even though we might take for granted the use of seconds, minutes, hours, days, and weeks to denote time, the horological system of today and the Gregorian calendar has historically speaking just been one of many competing temporal frameworks emerged within different socio-cultural contexts.

The prevailing international use of the Gregorian calendar should be seen in the light of the need for temporal coordination across nations in a time of globalisation of trade, production, and cultural interaction. It provides an "international temporal reference framework", used to mediate the synchronisation of social interaction on a global scale.

Third, as any other activity, the practical process of realising temporal coordination cannot be detached from the conditions of the concrete situation. Temporalcoordination is shaped according to the conditions of its object (i.e. the collaborative activity it tries to coordinate in time) and the conditions of the sociocultural environment in which it takes place. In collaborative work activities this environment is the organisational setting.

9.5.2 Temporal Composition

Temporal composition requires evaluating the relationships among component elements and scheduling their retrieval and presentation accordingly. The relationships can occur naturally, as for live audio and video, and they can also be synthetic, consisting of arbitrary temporal constraints on any multimedia data type. Figure 9.3 shows various data elements retrieved from storage and presented serially at the times indicated.

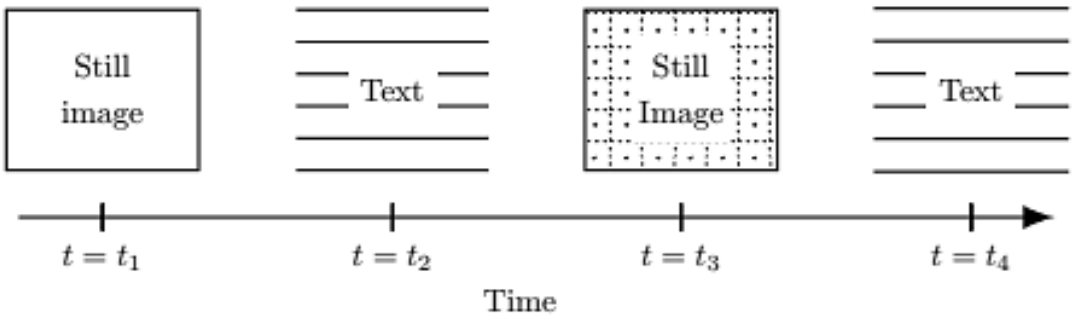


Figure 9.3 Temporal Composition

Multimedia data objects are time-dependent and must be synchronized accordingly. This synchronization requirement encompasses static objects, such as still images and text, and continuous streams of audio and video.Object composition requires consideration of both the temporal and spatial characteristics of multimedia elements. There are trade-offs to partitioning the object-composition process and its mapping onto the resources of the network based on the communication and computational requirements of an object. The mapping of this process, which combines spatial and temporal composition, results in a value-added service provided by the network.

9.6 TOOLKITS

9.6.1 Athena Muse

Athena Muse is an experiment kit for the construction of multimedia learning environments. Learning environments developed with Muse offer a diverse set of complementary interaction techniques, styles, and devices.

An interface developer can choose from four representation approaches: directed graphs, multidimensional spatial frameworks, declarative constraints, and procedural languages.At ProjectAthena,therequiredspatialdimensionsfor the Navigationdiskweregeneralizedto includetemporaland otherdimensions.Inparticular,severalforeign language videodiscsfundedbythe AnnenburgFoundationwerebeingproducedunderthe directionof Janet Murray.

Animportantgoal was to provideculturalcontextinadditionto practicewithgrammarand vocabulary.Studentswere presented withinteractivescenarios featuringnativespeakers. Inorder to respond correctly,studentsneeded to understandthe speakers. Thus,theyneededsubtitlessynchronizedto thevideo and the abilityto

controlthemtogether. The conceptof user-controllabledimensionswas created as a generalsolutionto the controlof spatiallyorganizedmaterial(as intheNavigationproject)and temporallyorganized material(as inthe foreignlanguage projects).

AthenaMuse packages text,graphics, and videoinformation togetherand allowsthemto be linkedina directedgraphformator operatedindependently. Differentmediacan be linkedto any numberof dimensionswhichcan thenbe controlledby the studentor end user. WhenreimplementedinAthenaMuse,the NavigationLearningEnvironmentused seven dimensionsto simulate themovementof a boat. Twodimensionsrepresent the boat's positionon thewaterand twomore represent the boat's headingand speed. A fifthtracks the user's viewingangle and the sixthand seventh manage a simulated compass thatcan be positioned anywhereon thescreen. Theuser can move freely withintheenvironmentand use the tools availableto a sailorto check location (charts, compass, lookinginall directionsaroundtheboat) to set a course.

Otheraspects of a simulationcan be added, suchas otherboats, weather conditionsunchartedrocks, etc. Here, the user does not change theunderlyingstructureof the information,since itis based on constraintsinthereal world,buthe or she can move freely,ask questions, and save informationintheform of notes and annotationsfor futureuse.

9.6.2 Quicktime

QuickTime consists of two major subsystems:The Movie Toolbox and the Image Compression Manager. The Movie Toolbox consists of a general API for handling time-based data, while the Image Compression Manager provides services for dealing with compressed raster data as produced by video and photo codecs.

Developers can use the QuickTime software development kit (SDK) to develop multimedia applications for Mac or Windows with the C programming language or with the Java programming language or under Windows, using COM/ActiveX from a language supporting this.
The COM/ActiveX option was introduced as part of QuickTime 7 for Windows and is intended for programmers who want to build standalone Windows applications using high-level QuickTime movie playback and control with some import, export, and editing capabilities. This is considerably easier than mastering the original QuickTime C API.

QuickTime 7 for Mac introduced the QuickTime Kit (aka QTKit), a developer framework that is intended to replace previous APIs for Cocoa developers. This framework is for Mac only, and exists as Objective-C abstractions around a subset of the C interface. Mac OS X v10.5 extends QTKit to full 64-bit support. The QTKit allows multiplexing between QuickTime X and QuickTime 7 behind the scenes so that the user need not worry about which version of QuickTime they need to use.

Quicktime provides a basic set of software compression/decompression schemes for still images, animations, and video. Included are compressors of JPEG, a run-length encoded animation format, and a digital video format. The Video Compressor allows digitized video sequences to be played for a hard disk or CD-ROM in real-time without additional hardware. Compression ratios ranging from 5:1 to 25:1 are possible. The video playback size is typically less than one-fourth of the computer screen size.

9.7 HYPERAPPLICATIONS

Hyperapplication features allow you to make links between information that is stored and accessed by DECwindows applications. A **hyperapplication** is an

application that, in addition to its regular features (for example, reading online books), allows you to link information and to follow links. You can tell that an application is a hyperapplication if it has the Link menu in its menu bar, as illustrated in the following figure.

This section presents a simple example to introduce some basic hyperapplication concepts and techniques. It discusses linkable objects (the pieces of information that you can link). It then discusses the features available from the Link menu, but not in the order that they appear on the menu. Rather, they are presented in an order to make it easy for you to learn and use hyperapplication features:

- Creating links (Start Link, Complete Link...)
- Following links (Visit, Go To, Go Back)
- Adjusting highlighting (Turn Highlight On/Off, Highlight...)
- Showing and deleting links (Show Links...)
- Other features (Show History..., Step Forward)

The section concludes with a section on advanced LinkWorks concepts.

Linkable Objects

Linkable objects are pieces of information in hyperapplications that you can link together.Information from any hyperapplication can be linked to information from the same hyperapplication or from any other hyperapplication. For example, a Calendar timeslot for a seminar could be linked to a Bookreader topic for a chapter that covers material to be discussed at the seminar. You can also link the Bookreader topic to a related topic in the same manual or in a different manual.

The linking possibilities are rich, and you can make them as complex as you want: for example, links between a Calendar timeslot and Bookreader topics and Cardfiler cards, with the topics and cards being linked to still other pieces of information. You can allow others to see and follow your links, and you can see and follow links made by others. We'll see later how you can progress from the simple uses of hyperapplication features to more complex and powerful uses.

Each hyperapplication determines which of its objects are linkable objects, based on the importance and granularity of objects. For example, a Cardfiler card is a linkable object, but a part of a word within a card is not. For a listing of the linkable objects for a specific hyperapplication, see its documentation in the *DECwindows Motif for OpenVMS Applications Guide*.

9.5 SUMMARY

- The primary goal of the Multimedia System Services is to provide an infrastructure for building multimedia computing platforms that support interactive multimedia applications dealing with synchronized, time-based media in a heterogeneous distributed environment

- For continuous media, stream parameters such as position, direction, and rate are usually available to the application

- The player abstraction is a natural way to control video and audio media.

- The X Imaging Extension (XIE) is an example of a server integrated processing system for image data.

- In the Fluency API, an audio stream and a video stream are treated as a unit for synchronization purposes.

- Temporal Coordination is an activity with the objective to ensure that the distributed actions realising a collaborative activity takes place at an appropriate time, both in relation to the activity's other actions and in relation to other relevant sets of neighbour activities
- Temporal composition requires evaluating the relationships among component elements and scheduling their retrieval and presentation accordingly

- Athena Muse packages text, graphics, and video in formation together and allows them to be linked in a directed graph format or operated independently.

- QuickTime consists of two major subsystems:The MovieToolbox and the Image Compression Manager.

- Hyperapplication features allow you to make links between information that is stored and accessed by DECwindows applications.

9.6Unit End Exercises

1. Which presentation-related services are needed by application developers?
2. Explain High-level view of X Server with X Imaging Extension.
3. How are multiple streams synchronised?
4. Write a note on device controls.
5. What is temporal coordination and composition?
6. Explain QuickTime in brief.

7. What are hyperapplications?

9.7 Additional Reference

- Multimedia Systems, John F. Koegel Buford

- Middleware versus Native OS Support: Architectural Considerations for Supporting Multimedia Applications by Prashant Shenoy, Saif Hasan, Purushottam Kulkarni, KrithiRamamritham available at http://lass.cs.umass.edu/papers/pdf/RTAS02.pdf

- Temporal Coordination:*On Time and Coordination of Collaborative Activities at a Surgical Department*by JAKOB E. BARDRAM available online at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.194.6777&rep=rep1&type=pdf

- *Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks* by Thomas DC Little and Arif Ghafoor available online athttp://hulk.bu.edu/pubs/papers/1991/TR-1991-10-01.pdf

- Research Article: *Virtual Video Editing in Interactive Multimedia Applications* by Wendy E. Mackay and Glorianna Davenport, available at https://www.researchgate.net/publication/220425653_Virtual_Video_Editing_in_Interactive_Multimedia_Applications

- Document: *Using DECwindows Motif for OpenVMS* available online at http://www.itec.suny.edu/scsys/vms/ovmsdoc073/v73/5633/5633pro_009.html#ch_conn_menu

10

Multimedia File Systems and Information Models

Unit Structure

10.1 Objectives
10.2 Introduction
10.3 The case for Multimedia Information Systems
10.4 File System Support for Continuous Media
10.5Data Models for Multimedia and Hypermedia Information
10.6 Content-based Retrieval of Unstructured Data
10.7 Summary
10.8 Unit End Exercises
10.9 Additional Reference

10.1 OBJECTIVES

In this Chapter you will understand:

- Corporate Information System
- Issues facing the design of Multimedia Information
- Document and Hypermedia Models

10.2 INTRODUCTION

Continuous media is data where there is a timing relationship between source and destination. The most common examples of continuous media are audio and motion video. Continuous media can be real-time (interactive), where there is a "tight" timing relationship between source and sink, or streaming (playback), where the relationship is less strict. The value of unstructured media forms such as images, audio, and video isthey represent certain types of information more effectively than other available

means.Continuous media have traditionally been absent from information systems because of issues including storage capacity, bandwidth limitations, scheduling issues, lack of appropriate data models, and inadequate query support.

10.3 THE CASE FOR MULTIMEDIA INFORMATION SYSTEMS

Multimedia user interfaces, databases, authoring and presentation tools, interchange, and many others have become standard parts of the application framework. Information that previously had to be stored and processed outside of the computer will now be accessible in the same context as the records and reports that make up day-to-day information flow of the organization.

All information, whether text, picture, audio, or video, will be integrated, easily accessed, and shared with others over the computer network. Such Corporate Information System supports rich information retrieval and organization communication. For example,

- CAD drawings annotated with voice or video
- Product simulations using animation and video
- Marketing materials including videos and other documentation on competitive products
- Requirements statements including video recording of users, customers and prior products
- Online manuals using videos of equipment operation
- Product development teams sharing design information through application sharing over a wide area network
- Multimedia mail containing video clips and/or pictures of proposed design changes and justifications

The ability to store and access all representations of information online solves the problem of disjoint information.

The main purpose of the corporate information system deployment is to use all resources efficiently and to enhance management solutions in order to get high profits. At present corporate information systems are used successfully not only in big companies but also in medium-sized companies. The achievement of multimedia information systems faces a variety of technical problems, which are due in part to two characteristics of multimedia information: large amounts of data and stringent temporal constraints for both delivery and recording.

Current research results in multimedia information systems in the areas listed in table 10.1 below. Since the area is very broad, the discussion is restricted to these areas, which are covered in the following sections:

- File system support for continuous media
- Data models for multimedia and hypermedia information
- Content-based retrieval of unstructured data

Component	Issues	Examples
Physical Media	High-capacity and performance storage architecture	Disk arrays
File System	Layout Admissibility	Contiguous versus scattered Deterministic guarantees or non-deterministic guarantees Round-robin, quality proportional Real-time with minimum rate and buffering
Data Model	Document model Temporal relationships	HyTime (Hypermedia/Time-based Structuring Language) OCPN (Object Composition Petri Net)
Query Language	Search mechanisms	Content-based retrieval
Application	Single or multiuser	VOD (Video-on-demand) Server

	Continuous media only or non-real-time data	

Table 10.1 Summary of Issues facing the Design of Multimedia Information

10.4 FILE SYSTEM SUPPORT FOR CONTINUOUS MEDIA

Historically, continuous media has not been cost-effective in the digital domain because of its large bandwidth requirements for transmission, and large storage space requirements. However, advances in both transmission and storage technology have made digital continuous media feasible. In particular, the advent of optical disk storage technology and fibre optic transmission technology has encouraged attention on digital continuous media. This is not to say that these technologies are required for continuous media. In fact, magnetic disk storage and wire transmission are advanced enough to support it.

The traditional file server was designed to provide access to text and numeric information. The first wave of multimedia research widened the scope of this service to include support for documents containing images. Examples include the Diamond System and the Muse System. The next wave of multimedia research involved the addition of audio and video in analog form. It included the Etherphone project's support for video using analog transmission and storage, and Mackay and Davenport's work on video filing using consumer electronic devices. As a final step, work on storage, retrieval and transmission of multimedia data all within the digital domain has taken place. In particular, the concept of an "on-demand" digital video server, which provides services similar to a neighbourhood video tape rental store over a metropolitan area network, has attracted attention.

In a multimedia server supporting conventional data, it would make sense to store conventional data in a log structure, even if log structures are not used for the CM data (e.g. partitioning the disk with a log-structured conventional partition, and a separate CM partition). Additionally, a multimedia server must also support interactive control functions such as fast forward (FF) and rewind. These operations can be implemented either by playing back media at a rate higher than normal, or by continuing playback at the normal rate while skipping some data. Since the former approach may yield significant increase in the data rate requirement, its direct implementation may be impractical. The latter approach may also be complicated by the presence of inter-data dependencies (e.g. in compression schemes that store only differences from previous data).

Continuous media presents new challenges to system designers because of its real time, high bandwidth nature. Supporting continuous media in a file server requires rethinking conventional server strategies including disk scheduling, admission control, data placement, and file mapping.

Handling digital audio and video data ('continuous media") in a general-purpose file system can leadto performance problems. File systems typically optimize overall average performance, while many audio/videoapplicationsneed guaranteed worst-case performance. These guarantees cannot be provided by fast hardware alone; we must also consider the interrelated software issues of file layout on disk, disk scheduling, buffer space management, and admission control, the Continuous Media File System addresses these issues.

10.4.1 Continuous Media File System

CMFS (Continuous Media File System) is a disk storage system for integrated Continuous Media. CMFS has the following properties:

- Clients of CMFS can reserve capacity in the form of sessions, each of which sequentially reads from or writes to a file - with a guaranteed data rate
- Multiple sessions, perhaps with different data rates, can exist concurrently, sharing a single disk drive
- Non-real-time traffic is handled concurrently. Thus, CMFS can be used asa general-purpose file system that handles CM data as well

The operations (RPCs or system calls) to create and start a CMFS session have the following form:

```
ID request_session (
        int direction,
        FILE_ID name,
        int offset,
        FIFO* buffer,
        TIME cushion,
int rate);
start_clock(ID id);
```

If direction is READ, request_session( ) requests a session in which the given file is read sequentially starting from the given offset. If the session cannot be accepted, an error code is returned. Otherwise, a session is established and its ID is returned. Start_clock() starts the session's logical clock. The client is notified (via an RPC or exception) when the end of the file has been reached. CMFS also provides a seek() operation that flushes data currently in the FIFO and repositions the read or write point. A real-time file is created using

```
create_realtime_file (
BOOLEAN expandable,
int size,
int max_rate);
```

expandable indicates whether the file can be dynamically expanded. If not, size gives its (fixed) size. max_rate is the maximum data rate (bytes per

second) at which the file is to be read or written. CMFS rejects the creation request if it lacks disk space or if max_rate is too high.

CMFS also supports non-real-time file access. There are two service classes: *interactive* and *background*. Interactive access is optimized for fast response, background for high throughput. The interface for non-real-time access is like that of a UNIX file system, except that the open() call specifies the service class. There are no performance guarantees for non-real-time operations

CMFS shows that it is possible for a file system to simultaneously handle multiple sessions with different data rate guarantees, together with non-real-time workload. This is an important step in the integration of audio/video in general-purpose computer systems. CMFS contributes new ideas in its acceptance test and scheduling policies, and also in the flexible but rigorous semantics of sessions.

10.5 DATA MODELS FOR MULTIMEDIA AND HYPERMEDIA INFORMATION

Multimedia refers to the integration of text, images, audio, and video in a variety of application environments. These data can be heavily time-dependent, such as audio and video in a motion picture, and can require time-ordered presentation during use. The task of coordinating such sequences is called multimedia *synchronization* or *orchestration.* Synchronization can be applied to the playout of concurrent or sequential streams of data and also to the external events generated by a human user.

Temporal relationships between the media may be implied, as in the simultaneous acquisition of voice and video, or may be explicitly formulated, as in the case of a multimedia document which possesses voice annotated text. In either situation, the characteristics of each medium

and the relationships among them must be established in order to provide coordination in the presence of vastly different presentation requirements.

In addition to simple linear playout of time-dependent data sequences, other modes of data presentation are also viable and should be supported by a multimedia information system (MMIS). These include reverse, fast-forward, fast-backward, and random access. Although these operations are quite ordinary in existing technologies (e.g., VCRs), when nonsequential storage, data compression, data distribution, and random communication delays are introduced, the provision for these capabilities can be very difficult.

10.5.1 Models of Time

A significant requirement for the support of time-dependent data playout in a multimedia system is the identification and specification of temporal relations among multimedia data objects.

As a multimedia author, we seek to specify the relationships between the components of this application. The service provider (the multimedia system) must interpret these specifications and provide an accurate rendition. The author's view is abstract, consisting of complex objects and events that occur at certain times. The system view must deal with each data item, providing abstract timing satisfaction as well as fine-grained synchronization (lip sync) as expected by the user. Furthermore, the system must support various temporal access control (TAC) operations such as reverse or fast playout.

Time-dependent data are unique in that both their values and times of delivery are important. The time dependency of multimedia data is difficult to characterize since data can be both static and time-dependent as required by the application.

For example, a set of medical cross-sectional images can represent a three-dimensional mapping of a body part, yet the spatial coordinates can be mapped to a time axis to provide an animation allowing the images to be described with or without time dependencies. Therefore, a characterization of multimedia data is required based on the time dependency both at data capture and at the time of presentation.

Time dependencies present at the time of data capture are called *natural* or *implied* (e.g., audio and video recorded simultaneously).

Table 10.2 Definitions of Time Dependencies

Static	**no time dependency**
Discrete	single element
Transient	ephemeral
Natural or Implied	real-world time dependencies
Synthetic	artificially created time dependencies
Continuous	playout is contiguous in time
Persistent	maintained in a database
Live	data originate in real time
Stored Data	data originate from pre-recorded storage

Time and Multimedia Requirements

The problem of synchronizing data presentation, user interaction, and physical devices reduces to satisfying temporal precedence relationships under real timing constraints. In this section, we introduce conceptual models that describe temporal information necessary to represent multimedia synchronization. We also describe language- and graph-based approaches to specification and survey existing methodologies applying these approaches.

The goal of temporal specification is to provide a means of expressing temporal relationships among data objects requiring synchronization at the time of their creation, in the process of orchestration. This temporal specification ultimately can be used to facilitate database storage and playback of the orchestrated multimedia objects from storage.

To describe temporal synchronization, an abstract model is necessary for characterizing the processes and events associated with presentation of elements with varying display requirements. The presentation problem requires simultaneous, sequential, and independent display of heterogeneous data. This problem closely resembles that of the execution of sequential and parallel threads in a concurrent computational system, for which numerous approaches exist. Therefore, there exists a bound on the speed of delivery beyond which a user cannot assimilate the information content of the presentation. For computational systems it is always desired to produce a solution in minimum time. An abstract multimedia timing specification concerns presentation rather than computation.

To store control information, a computer language is not ideal; however, formal language features are useful for the specification of various properties for subsequent analysis and validation. These systems allow specification and analysis of real-time specifications but not guaranteed execution under limited resource constraints. Providing guaranteed real-time service requires the ability to either formally prove program correctness, demonstrate a feasible schedule, or both. In summary, distinctions between presentation and computation processing are found in the time dependencies of processing versus display and the nature of the storage of control flow information.

Timing in computer systems is conventionally sequential. Concurrency provides simultaneous event execution through both physical and virtual mechanisms. Most modeling techniques for concurrent activities are

specifically interested in ordering of events that can occur in parallel and are independent of the rate of execution. However, for time-dependent multimedia data, presentation timing requires meeting both precedence and timing constraints. Furthermore, multimedia data do not have absolute timing requirements.

A representation scheme should capture component precedence, real-time constraints, and provide the capability for indicating laxity in meeting deadlines. The primary requirements for such a specification methodology include the representation of real-time semantics and concurrency and a hierarchical modeling ability. The nature of multimedia data presentation also implies further requirements including the ability to reverse presentation, to allow random access, to incompletely specify timing, to allow sharing of synchronized components among applications, and to provide data storage of control information. Therefore, a specification methodology must also be well suited for unusual temporal semantics as well as be amenable to the development of a database for timing information.

The important requirements include the ability to specify time in a suitable manner for authoring, the support of temporal access control (TAC) operations, and suitability for integration with other models (e.g., spatial organization, document layout models). In Section 7.4 we describe related requirements with respect to enforcing temporal specifications and delivery.

Relative versus Absolute Timing Specification

Temporal Instants

An instant-based temporal reference scheme has been extensively applied in the motion picture industry, as standardized by the Society of Motion Picture and Television Engineers (SMPTE). This scheme associates a virtually unique sequential code to each frame in a motion picture. By assigning these codes to both an audio track and a motion picture track, intermedia

synchronization between streams is achieved. This absolute, instant-based scheme presents two difficulties when applied to a computer-based multi-media application. First, since unique, absolute time references are assumed, when segments are edited or produced in duplicate, the relative timing between the edited segments become lost in terms of playout. Furthermore, if one medium, while synchronized to another, becomes decoupled from the other, then the timing information of the dependent medium becomes lost. This scenario occurs when audio and image sequences are synchronized to a video sequence with time codes. If the video sequence is removed, the remaining sequences do not have sufficient timing information to provide intermedia synchronization. Instant-based schemes have also been applied using MIDI (Musical Instrument Digital Interface) time instant specification as well as via coupling each time code to a common time reference. Other work using instant-based representation includes for editing multimedia presentations using timelines.

Temporal Intervals

Temporal intervals can be used to model multimedia presentation by letting each interval represent the presentation of some multimedia data element, such as a still image or an audio segment using TIB modeling. The notion of temporal intervals can also support reverse and partial playout activities. For example, a recorded stream of audio or video can be presented in reversed order. For this purpose, reverse temporal relations can be defined. These relations, derived from the forward relations, define the ordering and scheduling required for reverse playout. Furthermore, partial interval playout is defined as the playout of a subset of a TIB sequence.

Parallel and Sequential Relations

A common representation for time-dependent media relies on a subset of the thirteen temporal relations by using only the parallel *(equals)* and sequential *(meets)* relations.

By restricting temporal composition operations to these relations, most temporal interactions can be specified. This definition also requires uniform representation of data elements to eliminate overlap in time. Conversion from aperiodic representations can be achieved by decomposing larger intervals into a uniform size, as shown in Figure.

Temporal Access Control

A significant requirement for a media representation is the support of TAC operations. These operations provide the base functionality on which time-based multimedia applications can be built, including the system support (delivery). Clearly there are common characteristics required by the media authoring system, the user TAC functionality, and the system support primitives. Here we introduce and identify the following TAC operations:

- reverse
- fast-forward
- fast-backward
- midpoint suspension
- midpoint resumption
- random access
- looping
- pseudo-sequential access (browsing)

These operations can be implemented in various ways. For example, fast-forward can be provided either by skipping video frames or by doubling the rate of playout. Therefore, these operations can imply vastly different data structures and system delivery functionality.

Incomplete Timing

Under some conditions, it may be desirable to introduce incomplete timing specifications, as can often arise when time-dependent data are to be played

out in parallel with static ones. For example, if an audio segment is presented in synchrony with a single still picture, the time duration for image presentation could be unspecified and set to the duration of the audio segment. Incomplete specification can allow the static medium to assume the playout duration of the continuous medium. It is always possible to incompletely specify the timing for the parallel *equals* relation when only one medium is not static. For other types of relations, more information is required to describe the desired temporal result.

If both media have pre-assigned but unequal time durations, synchronous playout requires forcing one medium to alter its timing characteristics by time compression/expansion or data dropping/duplication.

Temporal Transformations

Temporal transformations change one frame of time referenceto another. These transformations can meet the Conceptual TAC operation requirements, although they can also be restricted by limitations of system delivery mechanisms.

Temporal transformations include

- scaling
- cueing
- inverting
- translation (shifting)

Temporal transformations can be applied to many time-based representations. If a time-based representation expresses precedence and ordering, or relative timing, temporal transformations can provide a mapping from the representational domain to a playout time coordinate system. For example, the relative timing of video frames as described by sequence numbers (i=1. 2, 3,...) can be mapped to real-time units of 15, 30, or *n*

frames/s. In time is described as a dimension that can be manipulated apart from real time and is treated as a virtual dimension. The system must provide support for any transformation or manipulation of this dimension (a virtual time coordinate system).

10.5.2 Hypermedia Document Models

The issues involved with hypermedia documentation and presentation with the development of the CWI Multimedia Interchange Format (CMIF), and the CMIFed environment for the creation and playback of CMIF hypermedia documents have been investigated. CMIF encodes crucial hypermedia functionalities such as synchronization, hyperlinking, screen display layout, and presentation style. Our main goal for CMIF is that it be adaptable to individual presentation situations. The model for CMIF documents is applicable to a wide variety of hypermedia presentation circumstances, including different document formats, variations in platforms and software, available resources, user characteristics, and desired style of presentation.

The CMIF document format has been developed in terms of the Amsterdam Hypermedia Model (AHM). The AHM extends the Dexter Hypertext Reference Model into multimedia by adding to it concepts such as the synchronization of multimedia object display and more complex specifications of how link activation affects document presentation. The Dexter model specifies significant abstractions shared by typical hypertext systems to provide a reference for discussing and comparing these systems. It is designed to aid in the general discussion of different hypertext documents and applications. It defines a layered model onto which individual hypertext systems can be mapped. It also defines common terms to be uniformly applied to the hypertext systems when they are described by Dexter.

This sectionalso presents the HyTime encoding for CMIF. It also describes the extension to the CMIF environment for processing this encoding and the

presentation instructions that accompany it. This translation provides an empirical test of HyTime's ability to represent the hypermedia structure of existing environments. It also provides an opportunity to explore the issues behind processing this generic structure for actual presentation on interactive multimedia environments. The HyTime encoding of CMIF uses Berlage architecture, an SGML-defined extension of HyTime that encodes how a document's HyTime-defined structure is mapped to certain aspects of presentation processing.

The Amsterdam Hypermedia Model

The Amsterdam Hypermedia Model has been designed to capture the elements of structure, timing, layout and interaction that are needed to specify an interactive, time-based on-line presentation. The AHM is not a perfect model of hypermedia, but instead seeks to achieve a balance of expressibility and implementability.

The main components of the AHM are the channel, atomic component, temporal and atemporal composite component and the link component. Presentation specifications that can be associated with these components are selected from temporal, spatial, style and activation information.

Although AHM has its roots in the Dexter and CMIF models, it incorporates the following novel extensions:

- The presentation specifications within the model have been explicitly stated as temporal, spatial, style and activation information. Each aspect occurs throughout the model and we have shown how the occurrences relate to one another.
- Anchors have been extended to include semantic attributes and presentation specifications, including start time and duration for an atomic anchor of a non-continuous media type.

- Content is specified explicitly as a media item reference along with a corresponding data-dependent specification.
- Anchor reference and channel reference in addition to a component reference are used throughout the model.
- Composition of anchors has been introduced.
- Composition of components is of two types: temporal and atemporal. Dexter expressed only atemporal and CMIF expressed temporal. Including both types of composition within one model requires the inclusion of activation state information
- Activation state information has been incorporated throughout the model. This includes: initial activation state, play/pause state and change in activation state on following a link.
- Link components have been extended to include context in the link specifier.
- Transition information, including transition duration and special effect, has been incorporated in the model.

The model could be extended in the following directions:

- Separate out the spatial layout hierarchy from the channel element and make a reference to it from a channel.
- Introduce spatial layout with respect to content.
- Include a way of selecting between synchronized streams of information, e.g. by introducing a visible/invisible state.
- Include a timeline more explicitly
- Separate out link semantic direction from link traversal direction.
- Allow the specification of anchors in terms of time and space for atomic and temporal composite components
- Allow the inclusion of absolute time within the model. This could be associated with children of atemporal composites but would require an extension to the current link activation mechanism

- Include auto-firing of links

Media item

A media item is an amount of data that can be retrieved as a single object from a store of data objects or is generated as the output from an external process. The form of a media item is outside the scope of the AHM. For a media item to be included in a document the requirements laid upon it are that the temporal and spatial dimensionality are known and that the corresponding duration and extent are also calculable. The duration may be specified as indefinite.

Channel

A channel defines a spatial position and extent and collects together a number of presentation and semantic attributes that are applicable to a particular media type. A channel consists of an identifier, a presentation specification, attributes and a media type.

The identifier is a globally unique identifier. The presentation specification stores a channel reference, spatial information for visual media types, and style information.

- The channel reference is a reference to another channel or a system-defined window.
- The spatial information specifies the position and extent of the channel. The position and extent are specified with respect to another channel, given by a channel reference, or a window.
- Style information includes media item style, anchor style and transition special effect.

The channel presentation specification may include default properties for the spatial and style information applicable to the media type associated with the channel, such as a scale factor, Z-order, orientation, background colour etc. The attributes allow semantic information to be associated with the channel,

for example the natural language used (for text or audio channels). A description of the attributes themselves falls outside the scope of the AHM. The media type is the specification of one or more data formats that can be played on the channel.

Document requirements

The identifier and media type are required for all channels. The channel reference, position and extent are required for channels with a visual media type. The style and attributes are optional. The channel reference, position and extent are meaningless for non-visual media types.

Requirements \ AHM elements		Channel			Atomic component						Composite components						Link component					
							Anchors						Anchors						Anchors			
		Pres. spec.	Attribtues	Media type	Pres. spec.	Attribtues	Pres. spec.	Attributes	Value	Content	Pres. spec.	Atributes	Pres. spec	Attributes	Value	Children	Pres. spec.	Attrbutes	Pres. spec	Attributes	Value	Specifi ers
Within com-ponent layer	Media item			*						*												
Anchoring	Part of media item								*	*												
Storage layer	Properties of instance				*	*	*	*														
	Composition: temporal atemporal spatial	*									* *				*	* *						*
	Linking																					*
	Semantic attr.		*			*		*				*		*				*		*		
Presentation Specifi cations	Temporal layout				*		*				*						*					
	Spatial layout	*			*												*					
	Style: media item anchor transition link	* * *			* * *		*				* * *		*				* *					* * *
	Activation state				*						*											*

Table 10.3 Hypermedia document model requirements and the elements of AHM

Atomic Component

An atomic component specifies information pertinent to a media item, including the data needed for displaying it. It consists of an identifier, a presentation specification, attributes, anchors and content. The identifier is a globally unique identifier. Semantic attributes can be associated with the component. They enable the creation of knowledge structures and information retrieval of the components. A description of the attributes themselves falls outside the scope of the AHM.

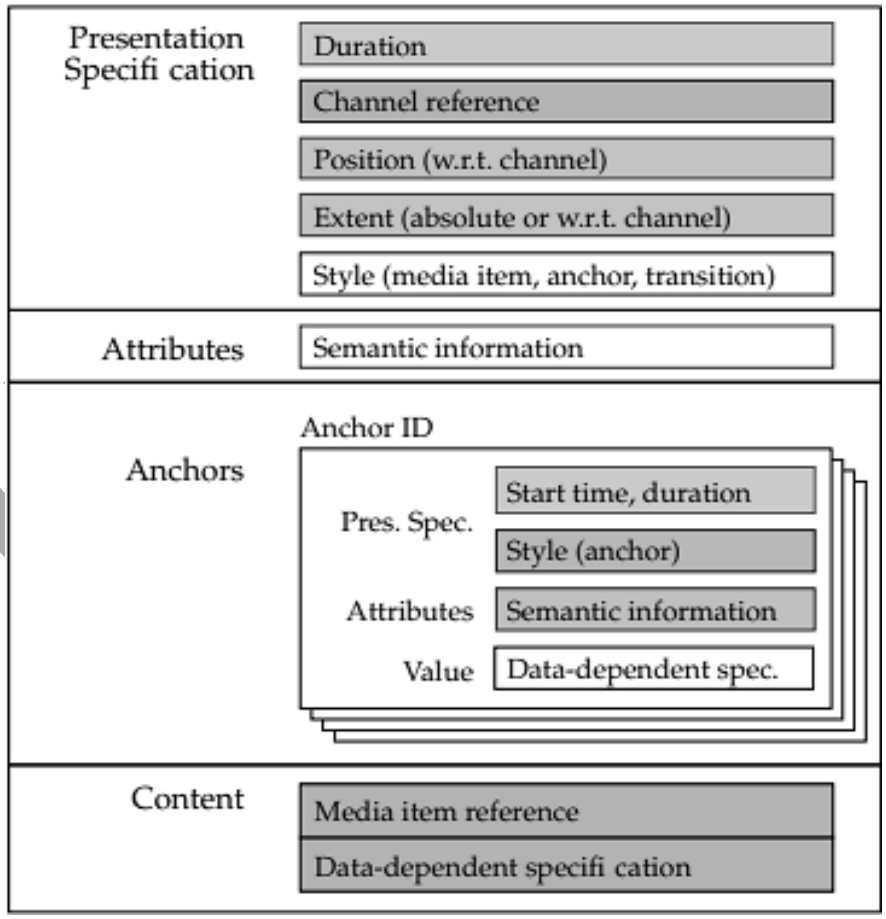


Figure 10.1 AHM Atomic components

Changes from Dexter are shaded

Content

The content specifies the data for the atomic component and consists of a media item reference, which is operating system dependent, and a data-dependent specification of part of the media item, which is data type dependent.

Document requirements

The parts of an atomic component that are required are the channel reference, the duration of the presentation specification and the content. The duration may be derived from the content. The other presentation specifications, attributes and anchors are optional. The content requires a media item reference and the data dependent specification is optional. The content need not be specified if a knowledge structure only is being created, but in this case the attributes are required. An anchor specified within an atomic component requires an anchor identifier and a value. Anchor attributes and presentation specifications are optional. The anchor value need not be specified if a knowledge structure only is being created, but the attributes would be required.

Composite components

An AHM composite component is a single element referring to a collection of atomic, composite and/or link components. The composition types in AHM are temporal and atemporal. Temporal composition is a grouping of components which are temporally related to one another. Atemporal composition is a grouping of components with no associated temporal relations.

As with the atomic component, an AHM composite component has an identifier, a presentation specification, attributes and anchors, but instead of content a list of children, Fig. 10.2. We specify the presentation specification of temporal and atemporal composites separately. The identifier is a globally unique identifier.

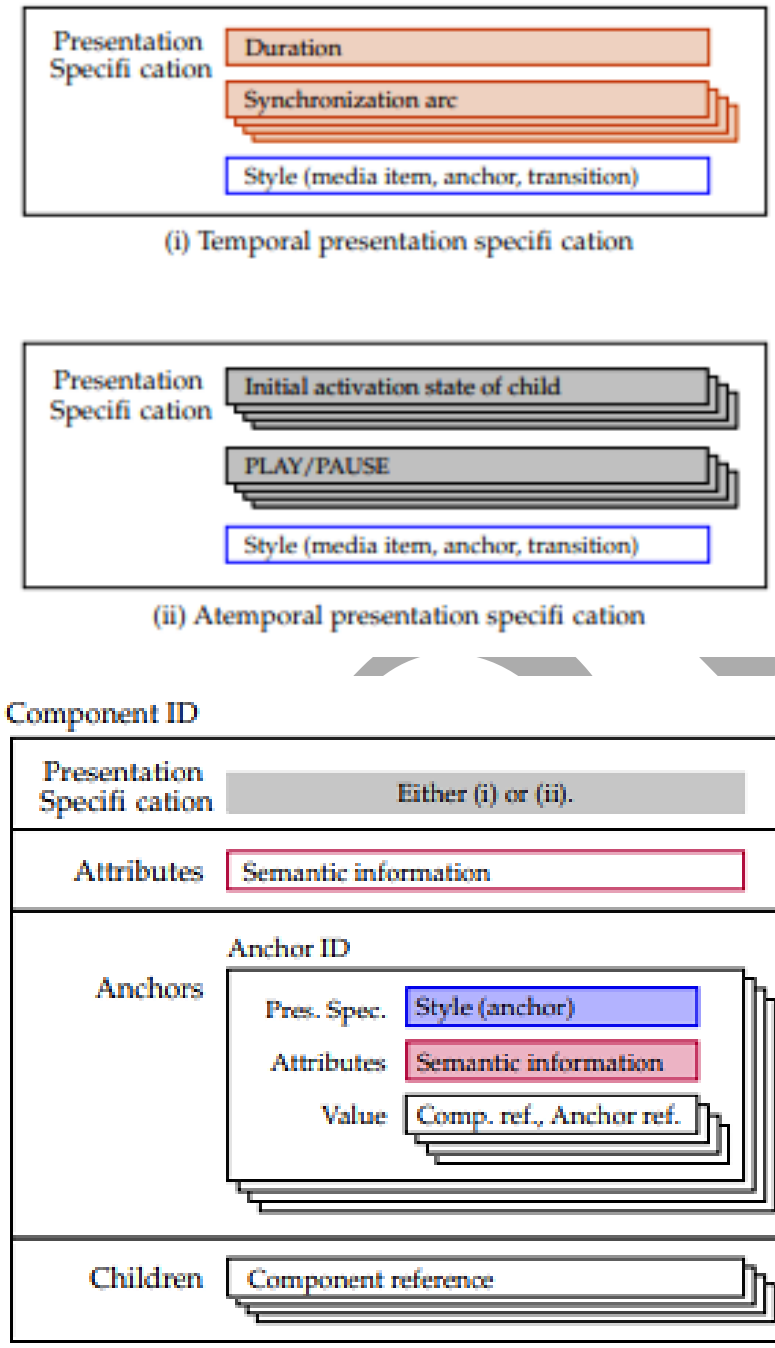


Figure 10.2 AHM Composite Components

Changes from Dexter are Greyed

The attributes allow the attachment of semantic information to the composite as a whole, for the creation of knowledge structures and for retrieval purposes. A description of the attributes themselves falls outside the scope of the AHM

Temporal composite presentation specification

The presentation specification for a temporal composite consists of temporal and style information, as in the Figure 10.2(i). The temporal information consists of synchronization arcs and a duration.

- A synchronization arc specifies the timing constraint between two parts of a presentation and in doing so establishes a single time axis shared by the ends of the arc. It consists of a source and destination, a scheduling interval and a synchronization type. The source and destination specify anchors which are the end-points of the temporal relation. Each is a component reference/anchor reference pair plus a START or END attribute. The anchor may be a point in time, but will more likely be an interval, although not necessarily contiguous. The START/END attribute specifies whether the scheduling interval starts from the beginning or the end of the source anchor reference and extends to the beginning or the end of the destination anchor reference. The scheduling interval specifies the temporal relation between the source and destination of the arc. The start time of the destination is relative to the source. The synchronization type specifies tolerance and precision properties. The source and destination component references are restricted to referring to a component which is a descendant of the temporal composite or the temporal composite itself. The children of a temporal composite with associated content require to be located along the same time-axis. This is achieved by the association of synchronization arcs. The intrinsic duration of the composite is the result of combining the durations of the children along with all the specified synchronization arcs. The duration may be indefinite. The synchronization arc does not have an identifier, since it is meaningful only within the temporal composite in which it is specified.
- The duration allows a scaling factor to be applied to the duration calculated from the duration of the children of the composite and the synchronization arcs.

- Style information can be specified to apply to all the descendant atomic components of the composite. It can contain media item style, anchor style and transition special effect. It may also include link style if the composite includes links.

Atemporal composite presentation specification

The presentation specification of an atemporal composite consists of an initial activation state and style information.

- The initial activation state specifies whether each child is played or not when the composite is activated at runtime. The play/pause state is the initial state of the child when it is made active. Each child of the composite requires an initial activation and play/pause state.
- Style information can be specified to apply to all the descendant atomic components of the composite. It can contain media item style, anchor style, transition special effect and link style.

Anchors

The anchor for a composite component has the same structure as the anchor for an atomic component: identifier, presentation specification, attributes and value.

- The identifier and attributes are the same as for an atomic component anchor.
- The presentation specification specifies an anchor style for the descendants of the anchor.
- The anchor value is a list of component reference/anchor reference pairs where the component reference refers to a component which is a descendant of the composite component. The component reference can refer to an atomic or composite component. The anchor reference may be omitted, with the interpretation that the complete component plays the role of the anchor. This removes the requirement for introducing a special anchor value for referring to a complete

component. The structure of the composite anchor is a hierarchy, since the composition of components is a directed acyclic graph and because of the descendant restriction on the component reference.

Children

The children are the components grouped together to form the composite and are given by a list of component references.

Document requirements

A temporal composite requires at least one child with associated content (directly or indirectly). The structure composed of the children and the synchronization arcs must specify a single temporal extent, in other words the components and synchronization arcs must form a connected graph. An explicit duration, extra synchronization arcs, styles, attributes and anchors are optional.Each synchronization arc requires a source and destination component reference/anchor reference and START or END attribute, where both components are descendants of the temporal component from which they are referred to, and a preferred scheduling interval. The source and destination cannot use the same component reference. The synchronization type is optional.

An atemporal composite requires at least one child. Each child is required to have an associated initial activation state and play/pause state. Synchronization arcs cannot be specified among descendants of an atemporal composite since the descendants are temporally independent. Styles, attributes and anchors are optional. The initial activation and play/pause states may be omitted if a knowledge structure only is being created, but in this case the attributes are required.

An anchor specified within a composite component requires an identifier and at least one component reference/anchor reference pair. The anchor

reference may be omitted from the component reference/anchor reference pair. Style and attributes are optional. The anchor value may be omitted if a knowledge structure only is being created, but in this case the anchor attributes are required.

Link component

A link component specifies a relationship among components (atomic, composite or link. It consists of an identifier, a presentation specification, attributes, anchors and a list of specifiers. The identifier is a globally unique identifier. Semantic attributes can be associated with the component, in particular for describing the relationship the link represents.

Presentation specification

The presentation specification consists of a duration, a relative position and style information.

- The duration is the duration of the transition of the source context to the destination context of the link (the definition of source and destination context is given below). It is specified using a synchronization arc, where the source of the arc is the END of the source context and the destination of the arc is the START of the destination context.
- The relative position is the position of the destination context with respect to the source context, where their layout is not prespecified via the channels, e.g. by being in different windows.
- The style information consists of a link style which can be used for creating displays of links.

Anchors

The anchor for a link component has the same structure as the anchor for the atomic and composite components.

- The identifier and attributes are the same as for atomic or composite component anchors. The presentation specification is a link anchor

style that can be applied to a visual representation of link to link graph structures.

- The anchor value is outside the scope of the model.

Specifiers

A specifier stores the information for the (possibly multiple) ends of the link and is itself composed of a number of parts: a presentation specification, an anchor, a context and a direction.

- The presentation specification consists of a source context activation state, a destination context play/pause state and style information. The source context activation state determines the activation state for a specifier with direction FROM or BIDIRECT when it is part of the source context. The specifier context can remain active or DEACTIVATE when the link is followed. When it remains active the specifier context canCONTINUE to play or PAUSE. The destination context play/pause state determines the activation state for a specifier with direction TO or BIDIRECT when it is part of the destination context. The specifier context becomes active and can eitherPLAY or PAUSE. The possible styles are anchor, media item and transition special effect which can be applied when the link is followed. For example, for highlighting a source anchor to show that it has been selected, or for specifying the media item style of the destination context.
- The anchor specifies an end-point of the relationship represented by the link. It is given as a component reference/anchor reference pair. The component reference can refer to an atomic, composite or link component. The anchor reference may be omitted, with the interpretation that the complete component plays the role of the anchor.
- The context specifies the scope of the relationship at a link end and is given by a component reference. The context component reference is restricted to being an ancestor of the anchor component reference and may be equal to it. The context is thus guaranteed to contain the

anchor. The context is further restricted to being an immediate child of an atemporal component, otherwise following the link may violate temporal relationships specified within a temporal composite.

- The direction specifies the direction of the relationship represented by the link and can be interpreted as a traversal direction. The values of direction are FROM, TO and BIDIRECT.

The transition information for the link consists of the link duration, the relative position and the special effect stored in the link specifier.

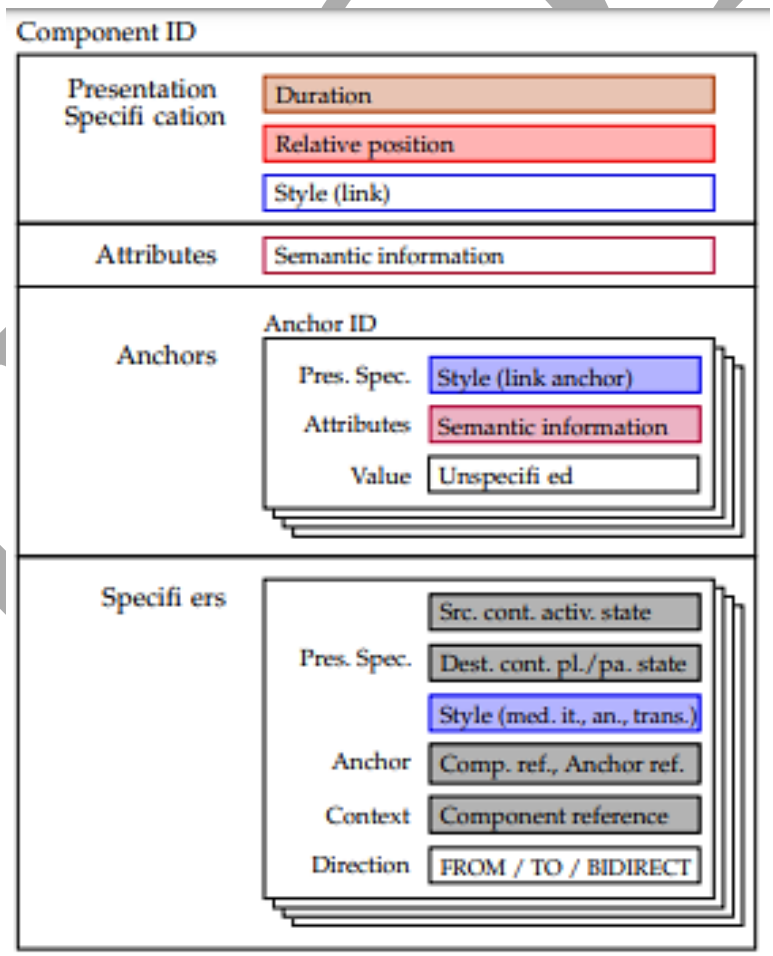


Figure 10.3 AHM Link Component

Changes from Dexter are greyed

Source and destination context

The source context of a link is the collection of specifier contexts that are active at the time the reader selects the link. The destination context is the collection of specifier contexts that are not part of the source context and that have direction TO or BIDIRECT, i.e. the collection of specifier contexts that will be made active because of the selection of the link.

Note that the source and destination contexts are runtime definitions.Also,that a specifier context may be in neither the source nor destination context, if it is inactive when the link is selected and has direction FROM.

Document requirements

The parts of the structure that require to be specified for a link component are at least one specifier. We do not go into the issues of dangling links. The presentation specification (including transition information), attributes and anchors are optional. For each specifier the source context activation state, the context and direction are required. For each specifier with direction TO or BIDIRECT the destination context play/pause state is required. For each specifier with direction FROM or BIDIRECT the anchor is required. Specifier style is optional. The specifier context component reference is restricted to being the immediate child of an atemporal composite component when the direction is FROM or BIDIRECT and the source context activation is DEACTIVATE.

HyTime

Hypermedia/Time-Based Structuring Language is an international standard for the definition of the structure of documents containing hypermedia andmultimedia information. HyTime is based on the international standard SGML (Standard Generalized Markup Language).

HyTime is formally defined by a set of constructors called Architectural Forms (AFs) used in the specification of the structural aspects of a document - hypermedia structural aspects include multi-ended links; multimedia structural aspects include powerful addressing and locating mechanisms used in the scheduling and synchronization of events in space and time.

Measurement module

The AFs of this module formalize the specification of measurement units using elements that allow the marking in space and, as a result, the definition of distance, positioning and length units. The measures are expressed as maximum and minimum integer values in a finite space. The architectural forms allow the definition of elements ranging from a point in space to complex areas in several dimensions.

Scheduling module

In order to allow the scheduling of the presentation of objects, HyTime adopts a model for space and time that is based on finite axes - where each axis defines one addressable space. A set of axes is defined as a finite coordinate space using the AF fcs. The scheduling module allows the definition of regions in a finite coordinate space where objects are to be placed - therefore an event is defined as the association of a data object to a particular interval in that space. The semantics associated with an event, as well as its presentation form are, as always in HyTime, defined by the application.

10.6 CONTENT-BASED RETRIEVAL OF UNSTRUCTURED DATA

The process of retrieval of relevant images from an image database(or distributed databases) on the basis of primitive (e.g. color, texture, shape etc.) or semantic image features extracted automatically is known as Content Based Image Retrieval.

It differs generically from conventional information retrieval/Data Mining (DM) due to the following reasons:

- Unstructured nature of image databases
- Contains pixel intensities with no inherent meaning
- Any kind of reasoning about image content is possible only after extraction of some useful image information(e.g, presence of primitive or semantic feature)

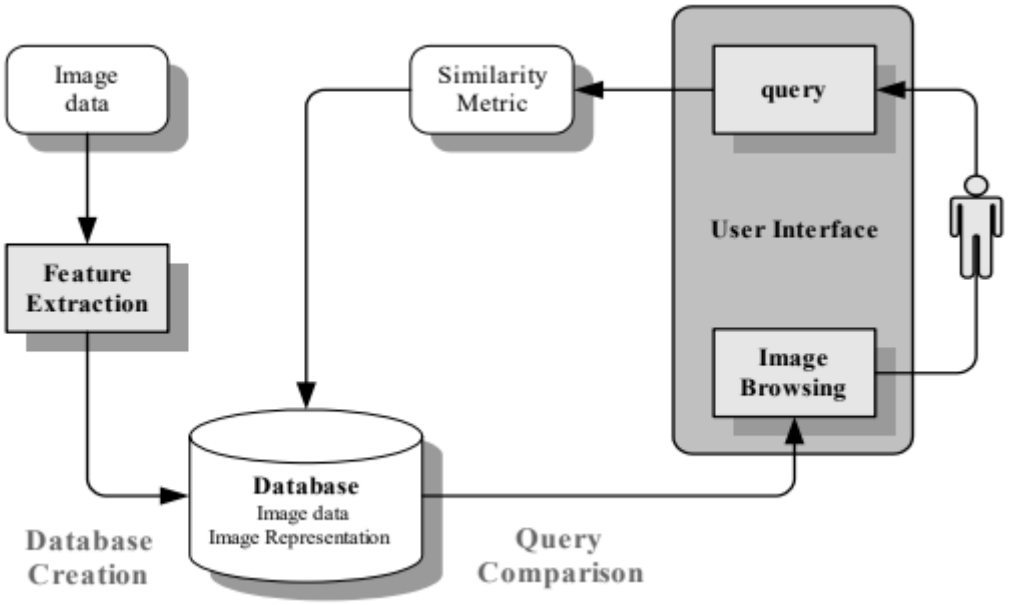


Figure 10.4 CBIR Architecture

What is Unstructured Data?

Any data without a well -defined model for information access. For example – Image, Video, Sound, Word documents, E -mails. Examples of structured data would be – Database tables, Objects, XML tags.Unstructured data contains significant scientific and commercial information. Technologies and tools are being developed for efficient extraction of information in unstructured data.

At the beginning of image retrieval, a user expresses his or her imaginary intention into some concrete visual query. The quality of the query has a significant impact on the retrieval results. A good and specific query may sufficiently reduce the retrieval difficulty and lead to satisfactory retrieval

results. Generally, there are several kinds of query formation, such as query by example image, query by sketch map, query by color map, query by context map, etc.. Different query schemes lead to significantly distinguishing results.The most intuitive query formation is query by example image. That is, a user has an example image at hand and would like to retrieve more or better images about the same or similar semantics. For instance, a picture holder may want to check whether his picture is used in some web pages without his permission; a cybercop may want to check a terrorist logo appearing in the Web images or videos for anti-terrorism. To eliminate the effect of the background, a bounding box may be specified in the example image to constrain the region of interest for query. Since the example images are objective without little human involvement, it is convenient to make quantitative analysis based on it so as to guide the design of the corresponding algorithms. Therefore, query by example is the most widely explored query formation style in the research on content-based image retrieval. Besides query by example, a user may also express his intention with a sketch map. In this way, the query is a contour image. Since sketch is closer to the semanticrepresentation, it tends to help retrieve target results in users' mind from the semantic perspective. Initial works on sketch-based retrieval are limited to search for special artworks, such as clip artsand simple patterns.

In content-based visual retrieval, there are two fundamental challenges, i.e., intention gap and semantic gap. The intention gap refers to the difficulty that a user suffers to precisely express the expected visual content by a query at hand, such as an example image or a sketch map. The semantic gap originates from the difficulty in describing high-level semantic concept with low-level visual feature.

Deep Learning in CBIR

Despite the advance in content-based visual retrieval, there is still significant gap towards semantic-aware retrieval from visual content. This is essentially due to the fact that current image representation schemes are hand-crafted and insufficient to capture the semantics. Due to the tremendous diversity and quantity in multimedia visual data, most existing methods are un-supervised. To proceed towards semantic-aware retrieval, scalable supervised or semi-supervised learning are promising to learn semantic-aware representation so as to boost the content-based retrieval quality. The success of deep learning in large-scale visual recognitionhas already demonstrated such potential.

To adapt those existing deep learning techniques to CBIR, there are several non-trivial issues that deserve research efforts. Firstly, the learned image representation with deep learning shall be flexible and robust to various common changes and transformations, such as rotation and scaling. Since the existing deep learning relies on the convolutional operation with anisotropic filters to convolve images, the resulted feature maps are sensitive to large translation, rotation, and scaling changes. It is still an open problem as whether that can be solved by simply including more training samples with diverse transformations. Secondly, since computational efficiency and memory overhead are emphasized in particular in CBIR, it would be beneficial to consider those constraints in the structure design of deep learning networks. For instance, both compact binary semantic hashing codes and sparse semantic vector representations are desired to represent images, since the latter are efficient in both distance computing and memory storing while the latter is well adapted to the inverted index structure.

10.7 SUMMARY

- In a multimedia server supporting conventional data, it would make sense to store conventional data in a log structure, even if log structures are not used for the CM data

- Supporting continuous media in a file server requires rethinking conventional server strategies including disk scheduling, admission control, data placement, and file mapping

- The time dependency of multimedia data is difficult to characterize since data can be both static and time-dependent as required by the application

- The problem of synchronizing data presentation, user interaction, and physical devices reduces to satisfying temporal precedence relationships under real timing constraints

- To describe temporal synchronization, an abstract model is necessary for characterizing the processes and events associated with presentation of elements with varying display requirements

- The children are the components grouped together to form the composite and are given by a list of component references

- The presentation specification consists of a duration, a relative position and style information

- The anchor for a link component has the same structure as the anchor for the atomic and composite components

- The source context of a link is the collection of specifier contexts that are active at the time the reader selects the link.

- The destination context is the collection of specifier contexts that are not part of the source context and that have direction TO or BIDIRECT, i.e. the

collection of specifier contexts that will be made active because of the selection of the link.

- HyTime is based on the international standard SGML (Standard Generalized Markup Language).

- The process of retrieval of relevant images from an image database (or distributed databases) on the basis of primitive (e.g. color, texture, shape etc.) or semantic image features extracted automatically is known as Content Based Image Retrieval

10.8Unit End Exercises

1. What are the Issues facing the design of Multimedia Information?
2. What is a Corporate Information System?
3. Write a note on Relative versus Absolute Timing Specification
4. Write a note on the Amsterdam Hypermedia Model
5. Explain the concept of HyTime.
6. What is Content Based Information Retrieval? Explain with the help of diagram.

10.9Additional Reference

- Principles of Multimedia, Eighth reprint edition 2009, Ranjan Parekh, Tata McGraw-Hill Companies

- Thesis: Support for Continuous Media in File Servers by D. James Gemmell available online at https://core.ac.uk/download/pdf/56370861.pdf

- The Continuous Media File System, research paper by David P. AndersonUniversity of California, Berkeley, available online at https://www.usenix.org/legacy/publications/library/proceedings/sa92/anderson.pdf

- Archive Copy: *A Framework for Generating Adaptable Hypermedia Documents* by Lloyd Rutledge,Jacco van Ossenbruggen, Lynda Hardman,Dick C.A. Bulterman available online at http://xml.coverpages.org/rutledgeFramework9711.html

- Thesis: Chapter 3: *The Amsterdam Hypermedia Model*, available online at https://homepages.cwi.nl/~lynda/thesis/Chap3.pdf

- Thesis: *Temporal Relations in Multimedia Objects: WWW Presentation from HyTime Specification* by Maria da Graca and Cesar A.C. Teixeira available online athttps://pdfs.semanticscholar.org/c94b/137e9a79e12adf57cef226c5aa0e59c48e4e.pdf

- Content Based Image Retrieval : An Introduction by Malay Kundu, Machine Intelligence Unit, Indian Statistical Institute, Kolkata available online athttps://www.isical.ac.in/~miune/LECTURES/Sikkim_CBIR_2014F_MKK.pdf

- *Recent Advance in Content-based Image Retrieval*: A Literature Survey by Wengang Zhou, Houqiang Li, and Qi Tian Fellow, available online athttps://arxiv.org/pdf/1706.06064.pdf

11

Multimedia Presentation and Authoring

Unit Structure

11.1 Objectives
11.2 Introduction
11.3 Design Paradigms and User Interface
11.4 Barriers to widespread use
11.5Research Trends
11.6 Summary
11.7 Unit End Exercises
11.8 Additional Reference

11.1 OBJECTIVES

In this Chapter you will understand:

- Easy-to-use tools for creating, manipulating and presenting multimedia contents
- Research Trends

11.2 INTRODUCTION

Multimedia user interfaces combine different media such as text, graphics, sound, and video to present information. Due to improvements in technology and decreases in costs, many human factors engineers will soon find themselves designing user interfaces that include multimedia. Since many educators, parents, and students believe that multimedia helps people to learn, one popular application of this technology will be the field of education.

The user interface also embodies the data and functions of computer-based products and provides a basis for the product's usability and commercial success. One of the important challenges to user interface design is how to help the novice user become quickly proficient and eventually become an expert user without the encumbrance of the training aids that were useful for the novice.

Systematic, information-oriented visual communication, or the graphic design of the user interface, is an important part of technologically sophisticated computer-based products as they spread internationally to consumer and business markets.

11.3 DESIGN PARADIGMS AND USER INTERFACE

In traditional programming paradigm, the user begins the program and then the program takes control and prompts the user for inputs as needed. Slowly, as graphical interfaces evolve, “event-driven” programs replaced this paradigm. Events are messages that the user gives to the program, for example, moving the mouse, clicking, double-clicking the mouse at a button on the screen, or typing a letter on the keyboard.

The term user interface in the context of the computer is usually understood to include such things as windows, menus, buttons, the keyboard, the mouse, the sounds that a computer makes, and, in general, all the information channels that allow the user and the computer to communicate. Most successful multimedia systems are used by people who do not have any knowledge of programming. A graphical user interface should be designed with this in mind and should allow ordinary people to use computers daily.

A good graphical user interface uses pictures rather than text to provide users an understanding of how to work on a system. A common understanding of

symbols allow us to make systems that can be used with little instruction. Screen objects are also known as Windowing System Components. Some examples of common screen objects are: Windows, menus, controls, dialog boxes, control panels, query boxes, etc.. The design of these screen objects have become standard within the Windows programming environment. Using the same kind of dialog boxes across various programs allows the user to transfer the knowledge gained from using one program to another.

Usability testing, as an emerging and expanding research area of human-computer interface, can provide a means for improving the usability of multimedia software design and development through quality control processes. In the process of usability testing, evaluation experts should consider the nature of users and the tasks they will perform, trade-offs supported by the iterative design paradigm, and real-world constraints in order to effectively evaluate and improve multimedia software.

Usability experts need to be in the field where they can see how real users work with real multimedia software. It is the responsibility of performance technologists, especially multimedia developers', to make multimedia software simple to use, simple to understand, yet still powerful enough for the task. The issue is no longer whether to conduct usability testing, but how to conduct useful usability testing.

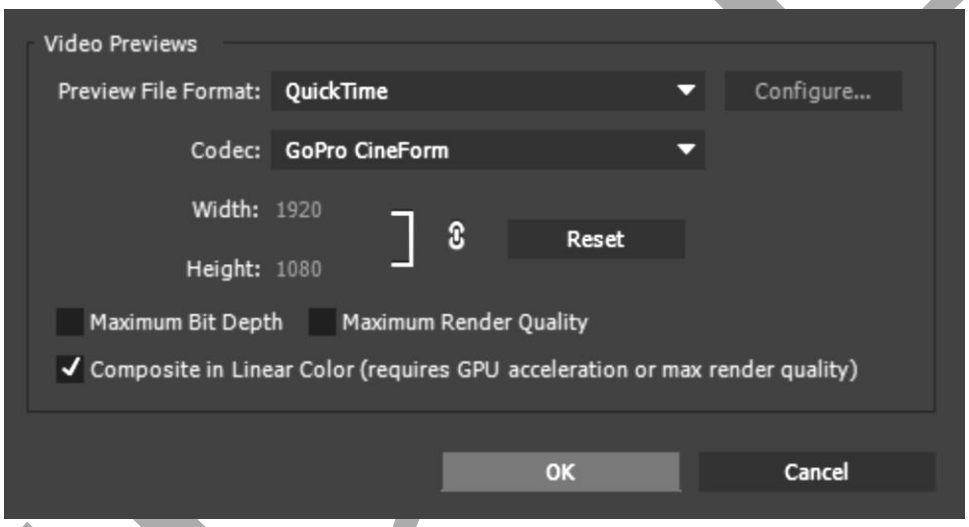
11.3.1 Overview of Tool

ADOBE PREMIERE

Adobe Premiere is a very simple video editing program that allows you to quickly create a simple digital video by assembling and merging multimedia components. It effectively uses the score authoring metaphor, in that components are placed in "tracks" horizontally, in a Timeline window.

Premiere Pro provides full smart rendering support for GoPro CineForm files on Windows. You can preview CineForm files in QuickTime format.

1. In the New Sequence or Sequence Settings dialog, select Editing Mode as Custom.
2. Select Preview File Format as QuickTime.
3. Select Codec as GoPro CineForm.
4. (Optional) Save a preset for each combination of height-and-width and frame rate that you commonly use.



The user interface provides HiDPI support for Apple's Retina displays and Windows 8.1 displays.

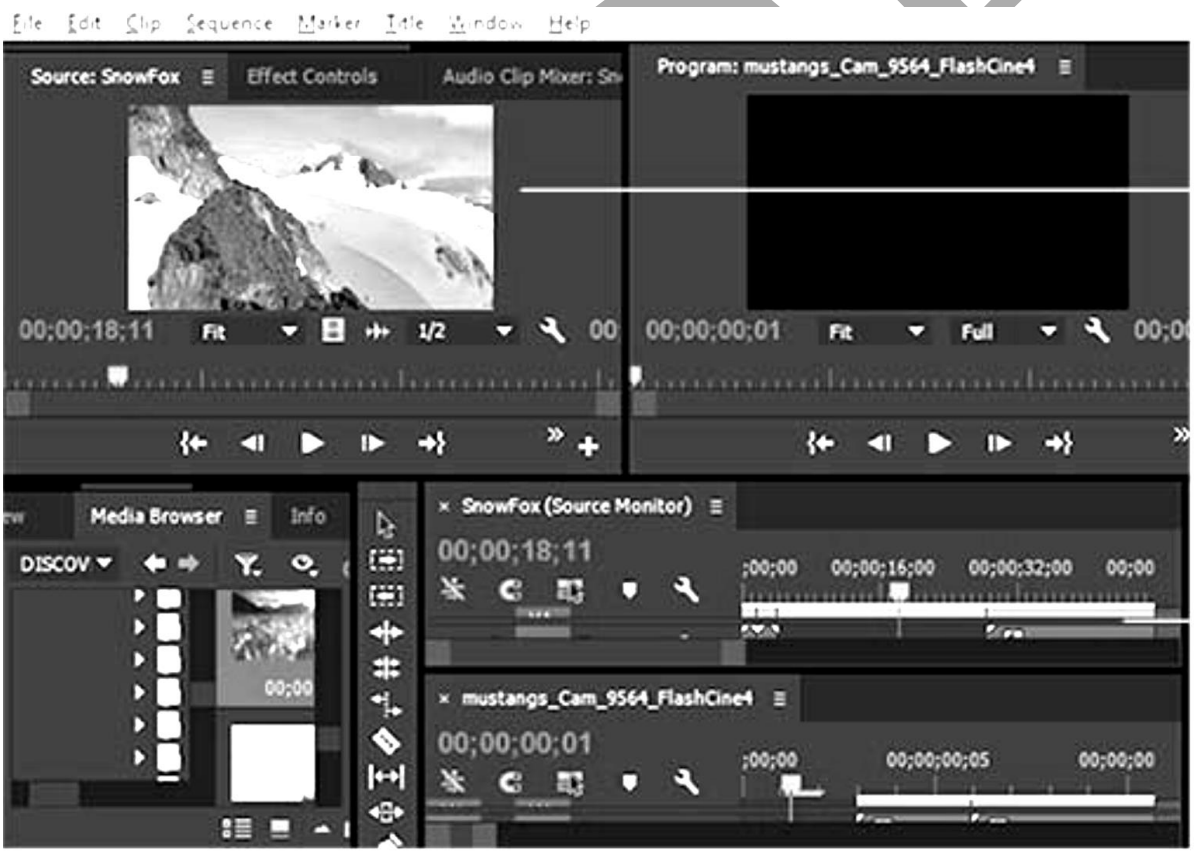
The modernized user interface provides cleaner visuals that let you focus more on the content. Apart from the visual enhancements, there are subtle but effective enhancements to the overall user experience. When you select a user interface element, the selected element appears with a blue outline indicating its active state. When deselected, it appears in gray. This high contrast helps you easily distinguish between selected and deselected elements.

For a comfortable viewing experience, you can vary the brightness of the user interface from a darker to a lighter tone by using the Appearance preference option.

With HiDPI support, Premiere Pro provides a higher resolution user interface that displays text, icons, and other user interface elements in greater clarity. You can notice an optimal display clarity under various scaling factors. At a 100% scaling, the application displays more real estate for viewing, which means many more panels can be viewed at once. When you change the

scaling, the user interface elements scale optimally and continue to appear sharp and clear.

Premiere Pro lets you open and view sequences from an unopened project without importing the sequence into your current project. Using the Media Browser, navigate to the project containing the sequence, and double-click the sequence to open. The new Source Monitor Timeline view opens a second Timeline that displays the contents of the sequence in read-only mode. This second Timeline makes it easy to edit or reuse existing clips, cuts, and transitions from different projects.



A Source Monitor Timeline View

Premiere Pro now provides an option in the Project Settings dialog that lets you keep all instances of your project items in sync automatically. Select File > Project Settings General to open the Project Settings dialog. In the Project

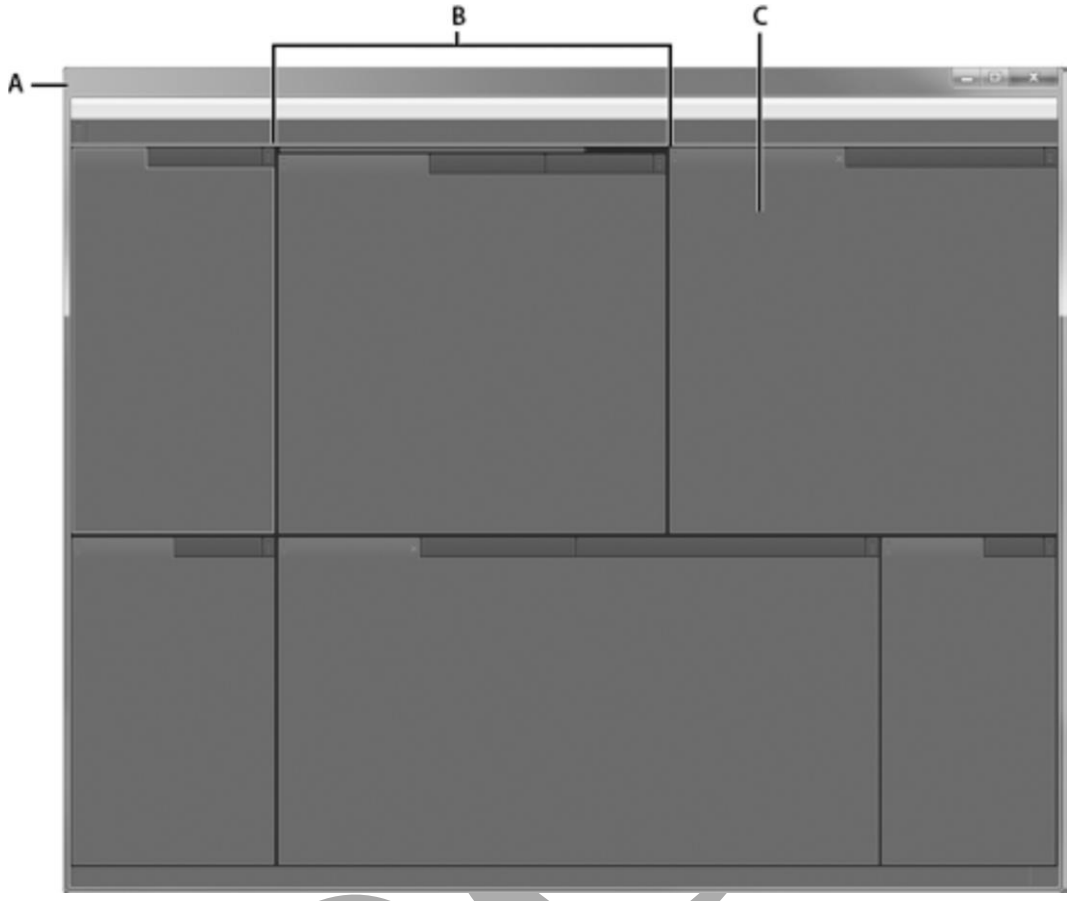
Settings dialog, select *Display The Project Items Name And Label Color For All Instances*. When you select this option, any changes made to a clip in the Project panel ripple to all instances used in sequences. For example, when you change the name of a sequence clip, it ripples up to the master clip and then down to all other sequence clips.

When you send a clip for audio editing in Audition by selecting *Edit > Edit in Adobe Audition > Clip*, the rendered copy of the clip is automatically saved alongside the original media file on disk. Storing the rendered media files along with the original files makes media management easier.

Adobe video and audio applications provide a consistent, customizable workspace. Although each application has its own set of panels (such as Project, Metadata, and Timeline), you move and group panels in the same way across products.

The main window of a program is the application window. Panels are organized in this window in an arrangement called a workspace. The default workspace contains groups of panels as well as panels that stand alone. You customize a workspace by arranging panels in the layout that best suits your working style. As you rearrange panels, the other panels resize automatically to fit the window.

You can create and save several custom workspaces for different tasks—for example, one for editing and one for previewing.



A: Application window B: Grouped panels C: Individual panel

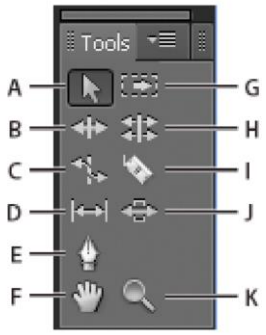
When you open the Options panel, it opens by default in the horizontal docking area running just under the menu bar, forming the Options bar. You can undock, move, and redock the Options panel like any other panel. By default, the Options panel contains a menu of workspaces and a link to CS Services. You can also dock the Tools panel to the Options panel.

TOOLS

When you select a tool, the pointer changes shape according to the selection. For example, when you select the Razor tool and position the pointer over a clip in a Timeline panel, the icon changes to a razor. However, the Selection tool icon can change to reflect the task currently being performed. In some cases, pressing a modifier key (such as Shift) as you use a tool changes its function, and its icon changes accordingly. Select tools from the Tools panel, or use a keyboard shortcut. You can resize the Tools panel and orient it

vertically or horizontally.

Note: *The Selection tool is the default tool. It's used for everything other than specialized functions. If the program isn't responding as you expect, make sure that the Selection tool is selected.*



A: SELECTION TOOL B: RIPPLE EDIT TOOL
C: RATE STRETCH TOOL D: SLIP TOOL E: PEN TOOL
F: HAND TOOL G: ROLLING TOOL H: ROLLING EDIT TOOL I: RAZOR TOOL
J: SLIDE TOOL K: ZOOM TOOL

Select any tool to activate it for use in a Timeline panel by clicking it or pressing its keyboard shortcut. Let the cursor hover over a tool to see its name and keyboard shortcut.

Selection Tool

The standard tool for selecting clips, menu items, and other objects in the user interface. It's generally a good practice to select the Selection Tool as soon as you are done using any of the other, more specialized, tools.

Track Selection Tool

Select this tool to select all the clips to the right of the cursor in a sequence. To select a clip and all clips to the right in its own track, click the clip. To select a clip and all clips to its right in all tracks, Shift-click the clip. Pressing Shift changes the Track Selection Tool into the Multi-track Selection Tool.

Ripple Edit Tool

Select this tool to trim the In or Out point of a clip in a Timeline. The Ripple Edit Tool closes gaps caused by the edit and preserves all edits to the left or right of the trimmed clip.

Rolling Edit Tool

Select this tool to roll the edit point between two clips in a Timeline. The Rolling Edit Tool trims the In point of one and the Out point of the other, while leaving the combined duration of the two clips unchanged.

Rate Stretch Tool

Select this tool to shorten a clip in a Timeline by speeding up its playback, or to lengthen it by slowing it down. The Rate Stretch Tool changes speed and duration, but leaves the In and Out points of the clip unchanged.

Razor Tool

Select this tool to make one or more incisions in clips in a Timeline. Click a point in a clip to split it at that precise location. To split clips in all tracks at that location, Shift-click the spot in any of the clips.

Slip Tool

Select this tool to simultaneously change the In and Out points of a clip in a Timeline, while keeping the time span between them constant. For example, if you have trimmed a 10-second clip to 5 seconds in a Timeline, you can use the Slip Tool to determine which 5 seconds of the clip appear in the Timeline.

Slide Tool

Select this tool to move a clip to the left or right in a Timeline while simultaneously trimming the two clips that surround it. The combined duration of the three clips, and the location of the group in the Timeline, remain

unchanged.

Pen Tool

Select this tool to set or select keyframes, or to adjust connector lines in a Timeline. Drag a connector line vertically to adjust it. Ctrl-click (Windows) or Command-click (Mac OS) on a connector line to set a keyframe. Shiftclicknoncontiguous keyframes to select them. Drag a marquee over contiguous keyframes to select them. For more information about using the Pen Tool, see Select keyframes.

Hand Tool

Select this tool to move the viewing area of a Timeline to the right or left. Drag left or right anywhere in the viewing area. Zoom Tool Select this tool to zoom in or out in a Timeline viewing area. Click in the viewing area to zoom in by one increment. Alt-click (Windows) or Option-click (Mac OS) to zoom out by one increment.

Navigate clips in the Source menu in the Source Monitor

You can set keyboard shortcuts for navigating multiple clips loaded into the Source Monitor. Keyboard shortcuts can speed toggling of clips, skipping to the first or last clip, or closing one or all the clips in the Source Monitor popup menu.

1. Select Edit > Keyboard Shortcuts (Windows) or Premiere Pro > Keyboard Shortcuts (Mac OS). The Keyboard Shortcuts dialog box opens.
2. In the dialog box, click the triangle next to Panels, and then click the triangle next to Source MonitorPanel to reveal the keyboard shortcuts for that panel.
3. Set keyboard shortcuts for any of the following commands:
   - Source Clip: Close • Source Clip: Close All • Source Clip: First
   - Source Clip: Last • Source Clip: Next • Source Clip: Previous
4. Click OK.

Using the Source Monitor and Program Monitor time controls

The Source Monitor has several controls for moving through time (or frames) in a clip. The Program Monitor contains similar controls for moving through a sequence.

A: Current time display B: Playhead C: Zoom scroll bar D: Time Ruler E: Duration display

Time rulers

Display the duration of a clip in the Source Monitor and sequence in the Program Monitor. Tick marks measure time using the video display format specified in the Project Settings dialog box. You can toggle the time rulers to display timecode in other formats. Each ruler also displays icons for its corresponding monitor's markers and In and Out points. You can adjust the playhead, markers, and the In and Out points by dragging their icons in a time ruler. Time ruler numbers are off by default. You can turn the time ruler numbers on by selecting Time Ruler Numbers in the panel menu of the Source or Program Monitors.

Playhead

Shows the location of the current frame in each monitor's time ruler. Note: *The playhead was formerly called the "current-time indicator" (CTI).*

Current time displays

Show the timecode for the current frame. The current time displays are at the lower left of each monitor's video. The Source Monitor shows the current time for the open clip. The Program Monitor shows the sequence's current time. To move to a different time. Alternatively, click in the display and enter a new time, or place the pointer over the time display and drag left or right. To toggle display between full timecode and a frame count, Ctrlclick (Windows) or Command-click (Mac OS) the current time in either monitor or a Timeline panel.

Duration display

Shows the duration of the open clip or sequence. The duration is the time difference between the In point and the Out point for the clip or sequence. When no In point is set, the starting time of the clip or of the sequence is substituted. When no Out point is set, the Source Monitor uses the ending time of the clip to calculate duration. The Program Monitor uses the ending time of the last clip in the sequence to calculate duration.

Zoom scroll bars

Zoom scroll bars correspond with the visible area of the time ruler in each monitor. You can drag the handles to change the width of the bar and change the scale of the time ruler below. Expanding the bar to its maximum width reveals the entire duration of the time ruler. Contracting the bar zooms in for a more detailed view of the ruler. Expanding and contracting the bar is centered on the playhead. By positioning the mouse over the bar, you can use the mouse wheel to contract and expand the bar. You can also scroll the mouse wheel in the areas outside of the bars for the same expanding and contracting behavior. By dragging the center of the bar, you can scroll the visible part of a time ruler without changing its scale. When you drag bar, you are not moving the playhead, however, you can move the bar and then click in the time ruler to move the playhead to the same area as the bar. A zoom scroll bar is also available in the Timeline.

Note: *Changing the Program Monitor's time ruler or zoom scroll bar does not affect the time ruler or viewing area in a Timeline panel.*

Before you begin editing

Before you begin editing in Premiere Pro, you will need footage to work with. You can either shoot your own footage, or work with footage that other people have shot. You can also work with graphics, audio files, and more. Many projects you work on do not need a script. However, sometimes you work from or write a script, especially for dramatic projects. You can write your script and organize your production details with *Adobe Story*.

While you shoot, organize your shots and take log notes. You can also adjust and monitor footage as you shoot, capturing directly to a drive. It is important to note that using Adobe Story is not necessary for editing with Adobe Premiere Pro. Writing a script, and making notes on the set are optional steps to help organize a project before you get started.

Get started editing

After you have acquired footage, follow the steps to get started editing with Premiere Pro.

1. **Start or open a project**

Open an existing project, or start a new one from the Premiere Pro Welcome screen. If you are starting a new project, the New Project dialog launches. From the New Project dialog, you can specify the name and location of the project file, the video capture format, and other settings for your project. After you have chosen settings in the New Project dialog, click OK. After you have exited the New Project dialog, the New Sequence dialog will appear. Choose

the sequence preset in the dialog that matches the settings of your footage. First, open the camera type folder, then the frame rate folder (if necessary), and then clicking a preset. Name the sequence at the bottom of the dialog, and then click OK. To open an existing project, click on a link under *Open A Recent Item* in the Premiere Pro Welcome screen. After clicking a link, the project will launch.

2. Capture and import video and audio

For file-based assets, using the Media Browser you can import files from computer sources in any of the leading media formats. Each file you capture or import automatically becomes a clip in the Project panel. Alternatively, using the Capture panel, capture footage directly from a camcorder or VTR. With the proper hardware, you can digitize and capture other formats, from VHS to HDTV. You can also import various digital media, including video, audio, and still images. Premiere Pro also imports Adobe® Illustrator® artwork or Photoshop® layered files, and it translates After Effects® projects for a seamless, integrated workflow. You can create synthetic media, such as standard color bars, color backgrounds, and a countdown.

You can also use *Adobe® Bridge* to organize and find your media files. Then use the Place command in Adobe Bridge to place the files directly into Premiere Pro. In the Project panel, you can label, categorize, and group footage into bins to keep a complex project organized. You can open multiple bins simultaneously, each in its own panel, or you can nest bins, one inside another. Using the Project panel Icon view, you can arrange clips in storyboard fashion to visualize or quickly assemble a sequence.

Note: *Before capturing or importing audio, ensure that Preferences>Audio>DefaultTrack Format* is set to match the desired channel format.

3. Assemble and refine a sequence

Using the Source Monitor, you can view clips, set edit points, and mark other important frames before adding clips to a sequence. For convenience, you can break a master clip into any number of subclips, each with its own In and Out points. You can view audio as a detailed waveform and edit it with sample-based precision.

You add clips to a sequence in a Timeline panel by dragging them there or by using the Insert or Overwrite buttons in the Source Monitor. You can automatically assemble clips into a sequence that reflects their order in the Project panel. You can view the edited sequence in the Program Monitor or watch the full-screen, full-quality video on an attached television monitor. Refine the sequence by manipulating clips in a Timeline panel, with either context-sensitive tools or tools in the Tools panel. Use the specialized Trim Monitor to fine-tune the cut point between clips. By nesting sequences—using

a sequence as a clip within another sequence—you can create effects you couldn't achieve otherwise.

4. Add titles

Using the Premiere Pro full-featured Title, create stylish still titles, title rolls, or title crawls that you can easily superimpose over video. If you prefer, you can modify any of a wide range of provided title templates. As with any clip, you can edit, fade, animate, or add effects to the titles in a sequence.

5. **Add transitions and effects**

The Effects panel includes an extensive list of transitions and effects you can apply to clips in a sequence. You can adjust these effects, as well as a clip's motion, opacity, and Variable Rate Stretch using the Effect Controls panel. The Effect Controls panel also lets you animate a clip's properties using

traditional keyframing techniques. As you adjust transitions, the Effect Controls panel displays controls designed especially for that task. Alternatively, you can view and adjust transitions and a clip's effect keyframes in a Timeline panel.

6. **Mix audio**

   For track-based audio adjustments, the Audio Track Mixer faithfully emulates a full-featured audio mixing board, complete with fade and pan sliders, sends, and effects. Premiere Pro saves your adjustments in real time. With a supported sound card, you can record audio through the sound mixer, or mix audio for 5.1 surround sound.

7. **Export**

   Deliver your edited sequence in the medium of your choice: tape, DVD, Blu-ray Disc, or movie file. Using Adobe Media Encoder, you can customize the settings for MPEG-2, MPEG-4, FLV, and other codecs and formats, to the needs of your viewing audience.

Cross-platform workflow

You can work on a project across computer platforms. For example, you can start on Windows and continue on Mac OS. A few functions change, however, as the project moves from one platform to the other.

1. **Sequence settings**

   You can create a project on one platform and then move it to another. Premiere Pro sets the equivalent sequence settings for the second platform, if there is an equivalent. For example, you can create a DV project containing DV capture and device control settings on Windows. When you open the project on Mac OS, Premiere Pro sets the appropriate Mac DV capture and device control settings. Saving the project saves these Mac OS settings. Premiere Pro translates these settings to Windows settings if the project is

later opened on Windows.

2. **Effects**

All video effects available on Mac OS are available in Windows. Windows effects not available on the Mac appear as offline effects if the project is opened on the Mac. These effects are designated “Windows only” in Premiere Pro Help. All audio effects are available on both platforms. Effect presets work on both platforms (unless the preset applies to an effect not available on a given platform).

3. **Adobe Media Encoder presets**

Presets created on one platform are not available on the other.

4. **Preview files**

Preview files made on one platform are not available on the other. When a project is opened on a different platform, Premiere Pro re-renders the preview files. When that project is then opened on its original platform, Premiere Pro renders the preview files yet again.

5. **High-bit-depth files**

Windows AVI files containing either 10-bit 4:2:2 uncompressed video (v210), or 8-bit 4:2:2 uncompressed video (UYVU) are not supported on Mac OS.

6. **Preview rendering**

The playback quality of un-rendered non-native files is not as high as playback quality of these files on their native platforms. For example, AVI files do not play back as well on Mac OS as they do on Windows. Premiere Pro renders preview files for non-native files on the current platform. Premiere Pro always renders preview files in a native format. A red bar in the timeline indicates which sections contain files needing rendering.

7. **Exporting for the Web and mobile devices**

Adobe Premiere Pro lets you create videos that can be exported to the Web or to mobile devices. To export your project, click on the sequence and select *File* > *Export* > *Media*. In the Export Settings dialog box, you can choose the most optimal file format, frame size, bit rate or ready-made presets for faster upload time and better playback quality.

11.4BARRIERS TO WIDESPREAD USE

In spite of recent advances in hardware and software technology, there are some substantial barriers to the widespread use and success of authoring and presentation systems. These issues are discussed in the following subsections:

Cost of acquisition, development and delivery of multimedia

Software applications are now available to help a developer plan, track and integrate the multiple information types involved. These tools can boost productivity, and help reduce the number of people and overall number of hours required for development.

As certain products become industry standards, it is likely that developers can be found who are already familiar with the authoring tools selected for the project, reducing the time required for staff training. Similarly, platform costs have dropped while functionality and storage have increased.

Difficulties with production quality

A major issue related to cost of development has been that of production quality for multimedia materials. High-quality video or audio can be expensive because of the need for professional equipment and facilities (such as sound stages and editing suites) and contract employees to create the audio or

video. If developers create the materials on their own without professional equipment or contract producers, the quality may not meet the users' expectation.

This situation is changing because of the technology. Equipment for recording high-quality video such as SVHS and Hi-8 mm camcorders is now much more accessible to developers. Digital Audio Tape (DAT) recorders can handle better than CD audio quality. Capture and edit tools for these media are available as off-the-shelf PC products today. Current development limitations include lighting and recording facilities.

Enforcement of Intellectual Property Rights

Due to the significant investment needed to obtain multimedia materials and the large potential value of such information, policies and mechanisms for enforcement of intellectual property rights are needed, otherwise, owners of such materials will be reluctant to make them available for use in multimedia applications.

The growing power of tools for manipulating digital media makes it easier for new content to be derived from existing content.

Cost, availability and ease of use of tools

More software tools have become available for designing and delivering multimedia productions, but problems still exist. Some applications cost several thousand dollars to purchase, and many require substantial learning time for a developer to become a proficient and productive user.

As noted earlier, the more powerful a tool is, the more difficult it is to become an expert, Ease of use will continue to be an important research issue.

Lack of standards for delivery and interchange

The same process that settled the standards issue for video disc developers may solve the notable fragmentation of delivery platforms that exist today. There are many activities underway, from formal standards organizations to industry consortia to vendor-driven. Without such standards, the fragmented market will make it harder to attract publishers of multimedia materials.

Lack of clear vision for multimedia applications

Today there seems to be tremendous interest in all aspects of multimedia, and early adopters are creating prototypes and innovative products. However, there is still a need for better understanding of the potential of new interactive media in areas such as business communications and mass market entertainment, particularly with regard to the forms these new media need to tale.

The new products and services being developed today are causing the most important transition in the industry: the raised awareness of both developers and users of the power and importance of the new media.

11.5 RESEARCH TRENDS

Web 2.0

Web 2.0 is a new term that describes a new form of technological communication that gives educators, business professionals or students "the ability to collaborate and share information online." (WebMediaBrands Inc., 2009).Web 2.0 consists of the following tools: – Podcasts – Blogs – RSS feeders – Social Networking

Benefits of Web 2.0

- Encourages participation and collaboration amongst educators, students and business professionals.
- Available anytime on line, anywhere from any location, inexpensive and affordable for anyone.

- Instant feedback on projects, reports or research papers.
- Sites are easy to design, manage and update.
- Opportunities for real time conversations, videoconferencing or messaging without having to download Instant Messaging (IM) software or constantly use "traditional" email.

Microsoft Surface

It is a commercial computing platform that enables people to use touch and real-world objects to share digital content at the same time. The Surface platform consists of software and hardware products that combine vision based multitouch PC hardware, 360-degree multiuser application design, and Windows software to create a natural user interface (NUI) for users.

Digital Spherical Displays

Provides a 360-degree multimedia display. It can be incorporated in many markets or territories and can be used in educational installations, exhibitions, displays, events, parties or set designs.

Popular Company: Pufferfish

Digital Spokesperson (live actor)

- Aids in bringing a website to life
- Adds a personal touch to a website
- Script can be tailored to fit the needs of your site
- Spokesperson can be used to market various sites or even conduct on line tutorials.

Personalized Access

Several important research directions may be derived from this vision:

- Representation of multimedia content descriptions
- Representation of user preferences and context

- Automated content annotation
- Automated presentation authoring

Multicast Backbone

Equivalent of conventional TV and Radio on the Internet enabling Technologies developing at a rapid rate to support ever increasing need for Multimedia. Carrier, Switching, Protocol, Application, Coding/Compression, Database, Processing, and System Integration Technologies at the forefront of this.

Synchronized Multimedia Integration Language (SMIL)

This is synchronized multimedia what HTML is to hyperlinked text. Pronounced smile, SMIL is a simple, vendor-neutral mark-up language designed to let Web builders of all skill levels schedule audio, video, text, and graphics files across a timeline without having to master development tools or complex programming languages.

Animation

Animation is the rapid display of a sequence of images of 2-D or 3-D artwork or model positions to create an illusion of movement. The effect is an optical illusion of motion due to the phenomenon of persistence of vision, and can be created and demonstrated in several ways.

Android Applications

Applications are usually developed in the Java language using the Android Software Development Kit, but other development tools are available, including a Native Development Kit for applications or extensions in C or C++, Google App Inventor, a visual environment for novice programmers and various cross platform mobile web applications frameworks.

Musical Instrument Digital Interface

MIDI is an electronic musical instrument, industry specification that enables a wide variety of digital musical instruments, computers and other related devices to connect and communicate seamlessly with one another.

11.6 SUMMARY

- Usability testingcan provide a means for improving the usability of multimedia software design and development through quality control processes

- Web 2.0 idescribes a new form of technological communication that gives educators, business professionals or students “the ability to collaborate and share information online.”

- SMIL is a simple, vendor-neutral mark-up language designed to let Web builders of all skill levels schedule audio, video, text, and graphics files across a timeline without having to master development tools or complex programming languages

- MIDI is an electronic musical instrument, industry specification that enables a wide variety of digital musical instruments, computers and other related devices to connect and communicate seamlessly with one another

11.7Unit End Exercises

1. Write a note on design paradigms and user interface
2. Explain the barriers to widespread use of multimedia
3. What are the latest research trends in multimedia?

11.8Additional Reference

- Principles of Multimedia, Eighth reprint edition 2009, Ranjan Parekh, Tata McGraw-Hill Companies

- Principles of Multimedia, Eighth reprint edition 2009, Ranjan Parekh, Tata McGraw-Hill Companies.

- Introduction to Multimedia Systems, Chapter 16, Gaurav Bhatnager, Shikha Mehta, Sugata Mitra, © 2002 by ACADEMIC PRESS.

- Designing Interactive Multimedia, paper by Lori L. Scarlatos, http://www.uni-mannheim.de/acm97/papers/Scarlatos/

- Premiere_pro_reference.pdf available at Adobe Website.

- Najjar, L. J. (1998). Principles of educational multimedia user interface design. Human Factors, 40(2), 311-323

- Designing Interactive Multimedia, paper by Lori L. Scarlatos, http://www.uni-mannheim.de/acm97/papers/Scarlatos/

- Flash_reference.pdf available at Adobe's Website

- Using_authorware_7.pdf available at Adobe's Website.

- Director_11.5_help.pdf available at Adobe's Website

- Latest trends in Multimedia available online at https://www.slideshare.net/TheSahilPunni/latest-trends-in-multimedia-by-sahil-punni

12

Multimedia Services over the Public Networks: Requirements, Architectures, and Protocols

Unit Structure

12.1 Objectives
12.2 Introduction
12.3 Network Services
12.4 Applications
12.5 Summary
12.6 Unit End Exercises
12.7 Additional Reference

12.1 OBJECTIVES

In this Chapter you will understand:

- Network services architecture
- Private Networks, Switched Networks, Metropolitan Area Networks

12.2 INTRODUCTION

Network-based solutions seem to be most appropriate for a moderate size community of endpoints (users, data sources, as well as data sinks), where the connection requirements are dynamic. If we accept this, the next question that arises is whether the services required should be provided on private or public networks. Again, the decision depends on the circumstances, and the specific economics have to be considered. Where the set of user sites to be connected is dynamic and relatively large, and where organizational and institutional boundaries need to be crossed, public network-based solutions seem to be the most viable.

In reality, if experience with past and present communication is to be any guide, all the above arrangements have to be provided for. Recall the LAN/digital PBX/Centrex debate and many others like it. The public network has to provide a range of services and functionality to meet this goal. The development of network-based services in this environment is quite a challenge, since traditional networks have not been designed to address these issues. New architectures, protocols, and services are required. Some of these are emerging, though not all the issues have been worked out.

Application Requirements

The class of end-user applications we are considering include video-conferencing, multimedia information access, shared multimedia workspace, and collaborative design. The industries where such applications are being considered are as varied as healthcare, education, manufacturing, training, financial, and entertainment. The benefits of multimedia and broadband communications have been amply covered in the literature. For a discussion of the strategic benefits of broadband communications to some of these industries.

We will focus on health-care applications to review the requirements that are imposed on the network services. Some typical applications in this field are: review of medical reports, medical report generation, teleradiology, remote pathology, surgical planning, and remote consultation. These applications readily lead to the unique characteristics mentioned in the introduction.

Review of Medical Reports

A typical medical report folder has text, images, charts, graphs, etc. If this is to be replaced by an electronic medicalrecord, there is a need for simultaneous access to text, databases, images various formats, and stored video (in case of cardiology and obstetric plications).

- **Medical Report Generation:** This has much the same needs as the view application. The additional needs are access to audio (for report cording) and live video (sonogram applications).

- **Teleradiology:** The main requirements here are that of high-speed large-image transfer Mb/s) as well as access to audio.

- **Remote Pathology:** Control channels are required for remote manipulation of image views.

- **Surgical Planning:** This combines the needs of teleradiology and emote pathology.

- **Remote Consultation:** Additional requirements in this case are the ability to share information among users, access to collaboration aids, and audio and/or video conferencing.

Case Study: Report/Review/Consult

We shall use the Report/Review/Consult application system developed jointly by NYNEX Science and Technology and the Children's Hospital inBoston (as part of the trials mentioned above) to be the primary focus for detailed discussion of the requirements below. We shall just touch upon the nature of the requirements. For details on the system that was implemented. This system has to address the requirements of three main work processes It the hospital, namely:

- **Reporting**: Creating a medical report based on the interpretation of an image-based examination;
- **Reviewing**: Review of the medical report by other imaging specialists, clinicians, surgeons, etc.;
- **Consultation**: Simultaneous review of the medical report by two or more physicians.

The following sections discuss the requirements for each of these processes in some detail.

Case Study: The Medical Reporting Process Requirements

The reporting process induces the following requirements.

Information Access:

The reporting physician needs access to patient information such as demographics, identification, and referral notes. This information is stored in hospital or departmental databases. Access is required to images acquired during the examination being interpreted as well as images acquired in previous exams. (In the particular case under consideration these were nuclear medicine images.) Retrieval of images from examinations performed with other modalities (in the particular case these would be computerized tomography or magnetic resonance studies) is also highly desirable for diagnostic purposes. Such images are stored in a variety of file systems and formats, depending on the equipment used to acquire them. Thus, to summarize, access to databases, file systems, and imaging devices is required along with support for handling images in a variety of formats. In addition, there is a need to store and manage relationships between pieces of information, which are stored and managed in a distributed manner.

Information Presentation:

Various image display modes are required. Examples are single and multiple image views, video view of image sequences, 3-D views, and support for display of varying resolution. The textual/database information described above has to be presented to the reporting physician. In addition, audio annotation and/or reports have to be available. There is clearly a need for mechanisms to transfer and coordinate the presentation of combinations of information in different media, with different granularities. The ability to

handle large data chunks in short periods of time is also apparent—the smallest requirement being the retrieval of a set of thirty 128-by-128 pixel images with 8 bits/pixel, in one second. The ensuing bandwidth requirements vary from 4 to 126 Mb/s, depending on the image modalities to be supported. See, for example, the tables in [Il for details. Image compression may reduce these requirements a bit, but not significantly, since lossy compression is unacceptable for most medical scenarios and because of the low latency response time required. In most other application areas, however, compression is acceptable for both image and video transmission and will lead to significant reductions in the bandwidth requirements. This is especially the case in video conferencing.

Information Manipulation and Processing:

Information manipulation and enhancement tools are required. Contrast enhancement and thresholding, image fusion between image sets, 3-D renderingfrom planar image studies, and the ability to support visual image directories, are some of the requirements on image manipulation. Text editing facilities, as well as audio record and playback facilities, are required for report creation. The audio needs to be stored in a manner that transcriptionists can access from the transcription system (via telephones). They can then transcribe the reports using word processing tools. The reports are then to be stored as part of the text report. (Future systems could involve voice recognitio[n] technologies to attempt this voice-to-text transfer.) State information of the reporting process as applied to the individual components (text, image, audio) of the report is required. Access to input and output facilities such as scanners and printers is also required. Apart from the need for end-station presentation tools, there is clearly a need for facilities to coordinate storage of the various media components and maintain automated transfer mechanisms between these. In certain cases, there is a need for access to network-based processing facilities such as

audio processing and computationally intensive processing (e.g., 3-D rendering).

12.3 NETWORK SERVICES

12.3.1 Architecture

The end-user applications communicate with each other and with the application services from the Application Service Provider on a peer-to-peer basis. This layer provides the network applications such as multimedia conferencing, mail, file transfer, etc., to the end user. The Application Service Layer calls upon services from the teleservice andnetwork access layers to perform its functions. The Advanced Call Services (Teleservices) Layer provides the control, data exchange, data access, and collaboration functions mentioned at the end of the previous section. There is peer communication at this layer across the transport network, as well aswith entities within the network to provide these services. The Network Access layer provides access to the appropriate data transport facilities. Again, there is peer communication at this layer across the transport network as well as with entities within the network to provide these Services. The transport and switching network carry out data transfer, switching and appropriate operations and management functions.

Note that the layering presented is a service hierarchy, not a protocol hierarchy. The complexity of the communication problem for multimedia forces us to an object-oriented approach. This is in contrast to the traditional protocol layer approach and procedure-based model for data communications as prescribed by the OSI model. Each of the layers in Figure 12.1 could be considered as a set of objects, which provides a well-defined set of services and has an appropriately defined interface. The layers provide increasingly complex services, as we proceed from bottom up. The entities at the

appropriate layer obtain services from any of the lower layers and are not restricted to service from the next layer below. As an example, data transfer with low delay requirements can take place between applications via the Access Layer directly, rather than going through the Teleservices Layer. The Teleservices layer itself may have been used to provide control and call setup services for this same transfer.

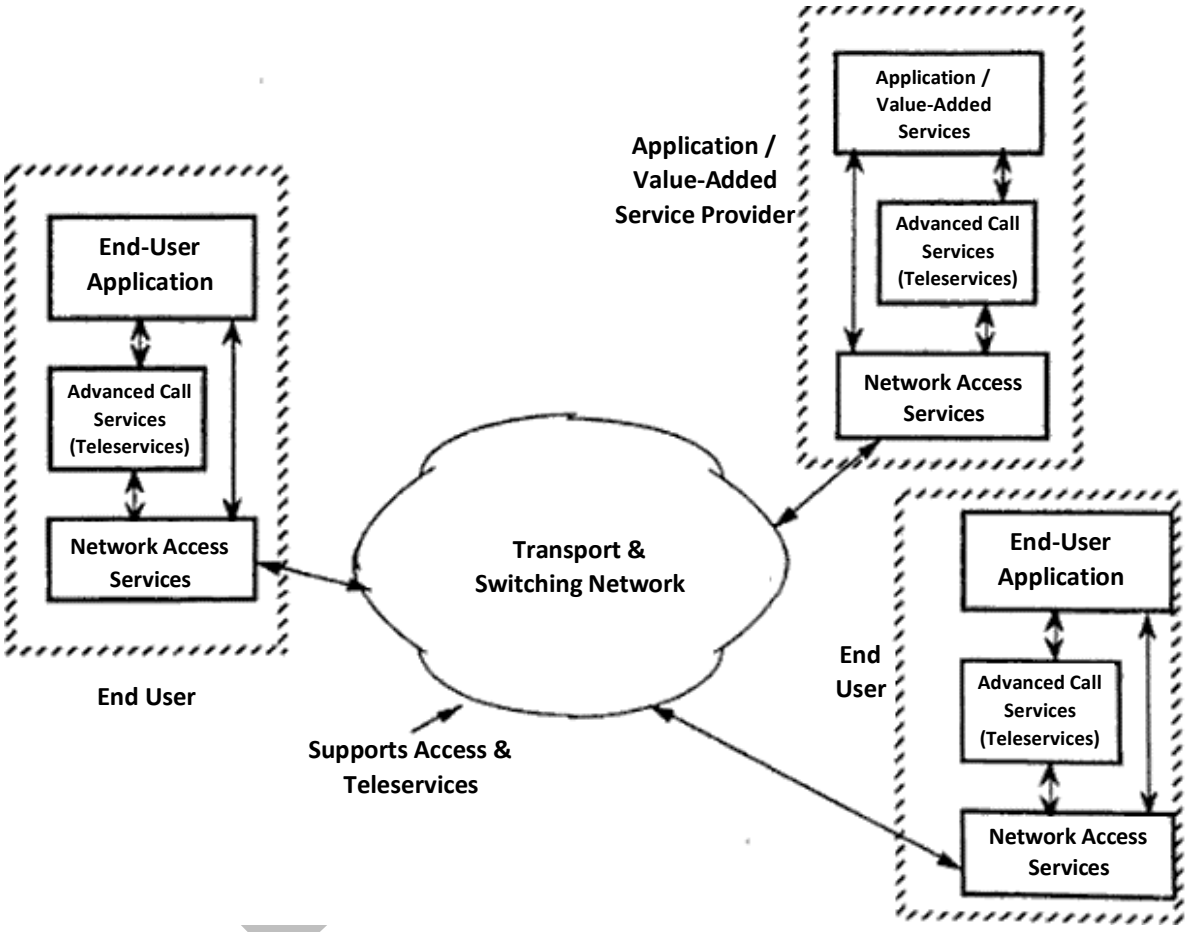


Figure 12.1 Architecture for network-based multimedia services

12.3.2 Middleware and Mediaware

The Advanced Call Services (Teleservices) are those needed to allow applications to make effective use of the underlying transport services which are emerging. These include media and session services.

- **Media services:** facilities for end-application collaboration, messaging and signalling; coordination of multiple media streams; distributed access to multiple media; and remote display and acquisition systems.

- **Session services:** facilities for multicast exchange of diverse data streams; security and privacy control; management and control of multipoint, multiuser, and multichannel connections.

We refer to the software systems which implement the media services as mediaware. Software which provides the session services is gaining recognition as middleware.

Media Services

The Media Services can be further classified into **Media Presentation** and **Media Control** services.

- **Media Presentation**: A layer of services to support access to distributed and multiple media information from heterogeneous customer premise equipment (CPE). These provide support for media data format conversion among different representations for a given media type and multimedia data synchronization to allow different types of media data to be presented with predefined spatial and temporal relationships

- **Media Control:** A layer of services to support sharing of and collaboration with multiple media information in multiuser communication. These include support for composite data management mechanisms for subscribers to compose documents consisting of different types of media; dynamic updates for shared data to update changes of shared composite data in a multiparty session; and arbitration and access control of shared data to provide arbitration and access control of shared composite data in a multiparty session.

Session Services

The session Services can also be classified into two subgroups: one providing **control** functions and another providing **advanced data exchange**.

- **Session Control**:

  A layer of services to establish and manage multipoint, multiuser, multinetwork (or channel) communication. These include facilities for multipoint session and multichannel session setup procedures allowing subscribers to create a session that consists of multiple channels and multiple endpoints; multichannel synchronization/coordination to synchronize the delivery of different types of media data being transported over different communication channels; dynamic session reconfiguration allowing subscribers to change the configuration of a session, such as adding or deleting parties within the session, without having to destroy and recreate sessions; and configurable call handling allowing subscribers to set up and change configuration preferences to allow for time-of-day and type-of-call-based call handling procedures.

- **Advanced Data Exchange**:

  A layer of services to support advanced data exchange mechanisms. These include message passing facilities to send information to all the parties in a session (broadcast) and to a subset of the session members (multicast); remote procedure call facilities to support distributed information processing via a client and server model; transactional call handling mechanisms allowing subscribers to submit a sequence of call commands in a single transaction; data security/privacy/ownership support mechanisms to allow subscribers to protect the security of their data using various kinds of locking and encryption; and deadlock avoidance mechanisms to provide adequate arbitration schemes to

avoid possible deadlocks and collisions in a multipoint communication environment.

Standards and Prototype Implementations

No general standard teleservices to support the above are available, and there is no attempt as yet to develop comprehensive standards for this layer of services. Various standards bodies such as ITU and ISO are focusing on specific capabilities. Examples are the representation and coding formultimedia information objects, document architectures, and message systems.

Various proprietary prototype services, which implement some subset of the functions mentioned above, are being developed worldwide. A listing of projects being pursued as part of various European research initiatives such as RACE (Research and Development in Advanced Communications Technologies in Europe) and DELTA (Development of European Learning through Technological Advance). The applications span many areas: banking, airline maintenance, education, and engineering design. None of these are as yet open systems which could interwork with others.

12.3.3 Access

The Access Services (Bearer Services) provide end users access to high-bandwidth data transfer mechanisms to support the needs of multimedia applications over wide area networks. A number of these services are emerging. (The data rates specified below are approximate.)

BISDN or Broadband Services

- **Switched Multimegabit Data Service (SMDS):** A connectionless cell-based (i.e., fixed-length packets) variable rate data service operating between 1.5 and 45 Mb/s

- **Cell Relay Service (CRS):** A connection-oriented cell-based data service operating from 45 to 150 Mb/s
- **Continuous Bit Rate Service:** A synchronous data service operating in the range 45 to 150 Mb/s (not standardized as yet)
- **Private Line Access Service (T3):** A synchronous data service at 45 Mb/s

Wideband Services

- **Primary Rate ISDN:** A channelized synchronous data service with 23 times 64 Kb/s channels and one 16 Kb/s channel, for a total of 1.5 Mb/s
- **Frame Relay Service (FRS):** A connection-oriented frame-based (i.e., variable-length packets) data service operating up to 1.5 Mb/s
- **Private Line Access Service (T1):** A synchronous data service at 1.5 Mb/s

The above services can be provided in a variety of flavours. Using the private line access services results in static private networks. *Virtual private networks* are created by using permanent virtual circuit-based FRS and CRS(primarily in the form of LAN interconnect services). Finally, on-demand switched broadband data transfer services are possible using SMDS, primary ISDN, or switched virtual circuit-based FRS, CRS. Another flavour of access is via metropolitan area networks (MANs). In this mode, shared access token passing schemes can be used on top of services such as SMDS to provide a wide area network service.

Interworking

The current widespread presence of narrowband services, coupled with the emerging broadband service availability, gives rise to the question of interworking. We would like to stress that the services mentioned here are access services and as such live on the periphery of the network. There is no

real incompatibility in allowing users to have simultaneous access, for instance, to narrowband ISDN and broadband continuous bitrate services. This is really a question of the availability of appropriate multiplexing and demultiplexing equipment and unified access to the network.

12.3.4 Transport, Switching, and Transmission

Data transport, switching, and transmission mechanisms, within the network, are required to support the network access services. Both private and switched network schemes are possible.

Private Networks

These can be provided by T-1 (1.5 Mb/s) and T-3 (45 Mb/s) access links, providing constant bitrate transport service over static private networks created with Digital Cross Connect switches (DCCS). The basic transmission services used are the standard digital hierarchies based on multiples of voice channel equivalents (64 Kb/s channels), in the bandwidth range 1.5 to 275 Mb/s for North American and ITU standards).

Switched Networks

The Asynchronous Transfer Mode (ATM) service can be used to provide high bandwidth (155 and 600 Mb/s) data transport. This is a connection-oriented, asynchronous cell transport service. The switching is provided by so-called fast packet switches or ATM switches - the speed being due to new switching fabrics, the fixed packet size, and the connection-oriented transfer.

The underlying digital transmission for this transport and switching mechanisms is the new optical transmission-based Synchronous Digital Hierarchy (SDH) with data rates from 52 Mb/s to 1 Gb/s [161. The SDH is the emerging international standard for these data rates. It is based On the North American SONET (Synchronous Optical NETwork) standard initiated by

Bellcore and adopted by the Tl Committee of the American National Standards Institute (ANSI).

Metropolitan Area Networks

Another form of transport network in the wide area environment is the so-called metropolitan area network configuration, where typically a logical ring/bus-type network is used with a token passing mechanism, using a protocol such as FDDI or IEEE 802.6. The bandwidths range from 10 to 100 Mb/s. Strictly speaking, this is an access arrangement, which would use underlying transport networks like that discussed above and in the Access subsection. An alternative is available where a physical fibre ring network is used to implement the transport itself.

12.4APPLICATIONS

12.4.1 The BISDN Reference Model

As already mentioned, the strictly one-dimensional layering of the OSI protocol stack is not appropriate for multimedia communications. The call and connection structures are fairly complex, multiple media streams have to be accommodated, and a more flexible structure is required, especially in the context of the public network. The BISDN reference model(shown in Figure 12.2), with its multidimensional layering, seems to be the most appropriate to support these requirements and the architecture presented previously. We will discuss protocols in the context of switched networks. The case of private networks is simpler and will be omitted from this discussion.

Note that there are two service planes for user (or data) and for control, and two management planes for layer management and for plane management. The service planes are layered, while the management planes are not layered. The plane management plane provides the protocols for management messages and actions for coordination between planes. The layer management plane takes care of operations and maintenance

communication for the layers. We will not consider the management aspects any further here. Of more direct interest for the purposes of multimedia services are the control and user planes.

An expansion of the BISDN reference model is proposed here for true provision of broadband services.

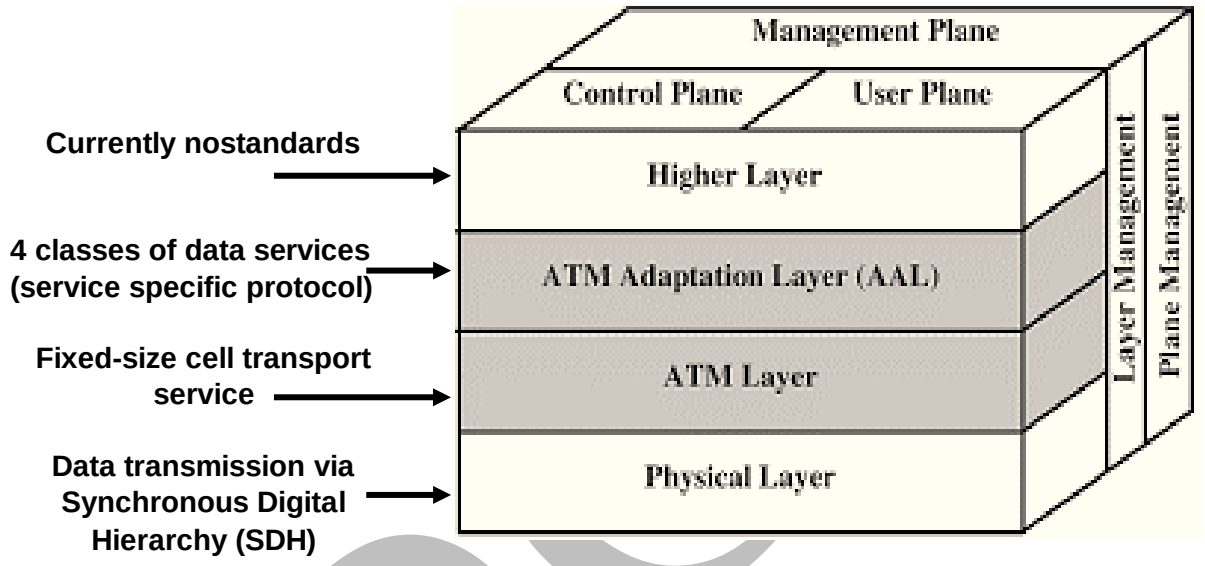


Figure 12.2 BISDN Protocol Reference Model

12.4.2 Advanced Call Services

The Advanced Call Services/Teleservices are implemented by the Media and Session Layers, which have components in both the user and control planes. The services provided by the sublayers Session Control, Media Control, Media Presentation, and Data Exchange have been described. As mentioned above, no standard protocols exist for these layers.

Capabilities within the Session Control and Media Control sublayers have been implemented on top of standard transport-level data protocols such as TCP/IP and private line access services. This implementation allows the establishment and management of multiparty, multistream sessions. It also allows for real-time sharing of data. The central concept here is of a session or workspace, defined in terms of the media streams available (voice, video,

data), where users can be added to or deleted. Many open issues exist in terms of performance and scalability to a larger population of users.
Other partial implementations of these layers have been reported. For a description of a model and a protocol implementation for call management functions, which are within the purview of the Session Control sublayer. The design is a distributed and object-oriented one, which seems quite suited to the nature of multimedia and broadband communications. The model is based on the notions of connections, which represent associations endpoints, and access edges, which represent media streams. It is fairly complex, and has focused at present only on the call management aspects.

Very little has been reported on implementations of the functions in the Media Presentation and Data Exchange sublayers. Existing data protocols at the OSI transport layer, such as TCP, provide no support for connection management involving multiple parties, dynamic control of resources during communication, and the other functions we have discussed. Moreover, for data exchange itself, they may be unacceptable in situations of higher bandwidths and large distances. This is due to the overhead in the traditional protocols, which were designed for high-speed processing and low-speed communication channels - just the opposite of the situation here. Note that this does not imply that protocols such as TCP cannot be used at all. Rather, they become a component of the Data Exchange Sublayer.

13.4.3 Access Services Layers

The services offered by the ATM Adaptation Layer (AAL) of the BISDN model correspond to the Access Services we have discussed earlier. The Adaptation Layer is defined to provide four classes of service (there is a possibility that the standards will subsequently be extended to provide five classes). These are based on combinations of the requirements for connection-oriented versus connectionless transfer, constant versus variable bitrates, and maintenance versus non-maintenance of timing between source and destination. The AAL

is implemented in sublayers, the lower sublayers implementing functions generic to the class of service and the upper sublayers being service-specific implementations. The access services mentioned previously—Switched Multimegabit Data Service (SMDS), Cell Relay Service (CRS), Frame Relay Service (FRS), and Continuous Bit Rate Service—are specific implementations of some of these classes. For MAN access arrangements, protocols such as FDDI or IEEE 802.6 are available.

13.4.4 Transport Service Layers

The Asynchronous Transfer Mode (ATM) Layer implements the standard connection-oriented, cell-transport service mentioned earlier as a uniform interface to underlying broadband and wideband networks. It provides high-bandwidth (approximately 155 Mb/s and 600 Mb/s) data transport. The cells are of fixed size (53 bytes, including a 5-byte header), and the connections are based on combinations of Virtual Channels and Virtual Paths (bundles of Virtual Channels), which have to be set up before the data transfer takes place. The service is asynchronous but cell sequence integrity is maintained on a Virtual Channel. A cell priority mechanism allows for connections with different qualities of service based on cell loss in the network (including zero loss connections).

12.5 SUMMARY

- The Advanced Call Services (Teleservices) are those needed to allow applications to make effective use of the underlying transport services which are emerging.

- The Media Services can be further classified into Media Presentation and Media Control services.

- The session Services can also be classified into two subgroups: one providing control functions and another providing advanced data

exchange

- The Access Services (Bearer Services) provide end users access to high-bandwidth data transfer mechanisms to support the needs of multimedia applications over wide area networks.

- Data transport, switching, and transmission mechanisms, within the network, are required to support the network access services

- Another form of transport network in the wide area environment is the so-called metropolitan area network configuration, where typically a logical ring/bus-type network is used with a token passing mechanism, using a protocol such as FDDI or IEEE 802.6.

- The Advanced Call Services/Teleservices are implemented by the Media and Session Layers, which have components in both the user and control planes

- The Adaptation Layer is defined to provide four classes of service (there is a possibility that the standards will subsequently be extended to provide five classes).

- The Asynchronous Transfer Mode (ATM) Layer implements the standard connection-oriented, cell-transport service mentioned earlier as a uniform interface to underlying broadband and wideband networks

12.7Unit End Exercises

1. Explain the architecture of network-based multimedia services
2. Write a note on Media Services.
3. Explain Session Services.

4. What are Private, Switched and Metropolitan Area Networks?
5. Explain BISDN Reference Model.

12.8Additional Reference

- *Multimedia Systems* by John F. Koegel Buford- Pearson Education

13

Multimedia Interchange

Unit Structure

13.1 Objectives
13.2 Introduction
13.3 QuickTime Movie File Format
13.4 Open Media Framework Interchange (OMFI)
13.5Multimedia and Hypermedia Information Encoding Expert Group (MHEG)
13.6 Format Function and Representation
13.7 Track Model and Object Model
13.8 Real-time Interchange
13.9 Summary
13.10 Unit End Exercises
13.11 Additional Reference

13.1 OBJECTIVES

In this Chapter you will understand:

- Multiple File Formats for Multimedia Interchange
- Multiple Models for Multimedia Interchange

13.2 INTRODUCTION

A Variety of multimedia applications running on different platforms will need to communicate with each other particularly if they are running on a distributed network.Until recently (and it may even still pose some problems) the lack of a common interchange file format was a serious impediment to development of a market of multimedia applications.A common interchange format needs to be widely adopted (be supported by many applications) and be sufficiently

expressive to represent a wide variety of media content. These may be conflicting requirements since only when a wide variety of media is supported will it be widely adopted. Propriety application on support a small variety of media they require and may not readily adapt to other formats. Fortunately, some widely accept standard that support a wide variety of media (with open standards even) are now developed.The need for interchange formats are significant in several applications:

- As a final storage model for the creation and editing of multimedia documents.
- As a format for delivery of final form digital media. E.g. Compact Discs to end-use players.
- As a format for real-time delivery over a distributed network
- for inter-application exchange of data.

Although it is generally believed that a standard interchange format will play a crucial role in the growth of the multimedia application market, different proposals currently being developed have divergent models and each appears to have primarily evolved using an experimental methodology. In this paper we first provide a survey of these current efforts, discussing goals, architecture, and abstractions of each format. We provide a comparative summary in terms of representational capability and functionality.

We classify the composition models as either track oriented or object oriented, and use this distinction to clarify differences in inter-object referencing, compositionality, and access/presentation procedures. We enumerate features that would enhance the ability of a format to support real-time interchange, and conclude with an overview of an approach to rigorously evaluate and compare such format models. This approach is based on a set of benchmark interchange cases and various parameters to be measured in a performance test.

In order for multimedia applications to work together and realize the benefits of distributed computing, a common interchange format for multimedia information is needed. It is not sufficient for the individual media formats to be standardized. The temporal, spatial, structural, and procedural relationships between the media components are an integral part of multimedia information and must also be represented. Today there is a growing realization that lack of a common format is a serious impediment to the development of the market for multimedia applications. Without a representation that is widely adopted and is sufficiently expressive, multimedia content that is created in one application cannot be read or reused by another application. Further, in order for multimedia information to be used on several platforms, each application-defined format must have a converter on each platform.

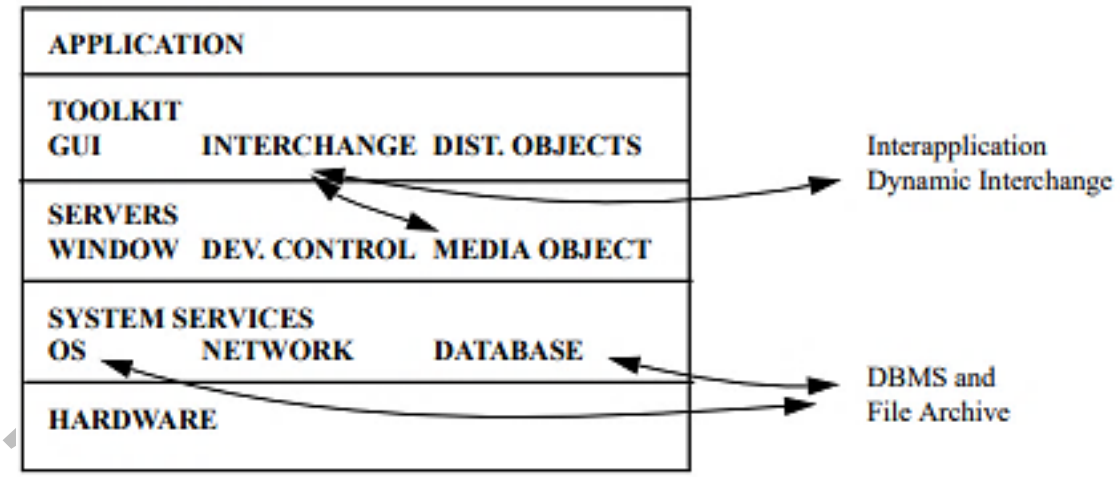


Figure 13.1 Multimedia Interchange context

In the architecture of multimedia systems, interchange appears in three different modes:

1. **Interapplication interchange:** two or more applications exchange multimedia information using either an interchange API or a distributed object API.

2. **Archive**: an application saves and accesses multimedia information through a file system or DBMS, also using an interchange API. Contemporary multimedia authoring packages follow this case.

3. **Presentation**: A media object server provides media objects to distributed clients in a networked environment. An example application is video-on-demand

The design of multimedia interchange formats can also be viewed in the context of the interchange format hierarchy (Table 13.1). In this diagram, levels correspond to increasing specificity. At the top level are general container formats. These formats are application independent. At the bottom level are media specific formats. These formats are optimized for a specific media type.

Category	Examples
General Container	GDID, ASN.1, Bento
Metalanguage	HyTime
Multimedia document architecture	HyTime DTD
Special purpose object container	QuickTime Movie File, MHEG, OMFI
Monomedia	MPEG, JPEG, Script Languages

Table 13.1 The Multimedia Interchange Format Spectrum

The major technical issues that must be addressed include:

1. **Multimedia data model:**

   A data model for structured time-based interactive media (multimedia and hypermedia) including temporal composition, synchronization, multiple media formats, addressing of media objects and composite media objects, hyperlinking, and an input model for interaction.

2. **Scriptware integration:**

   Many authoring tools integrate multimedia data with specialized procedural scriptware which may be text based or iconic languages. These tools have a tight association of scripts with media objects and media composites, in particular associating input semantics of input objects with script input processing. The interchange formats must retain the associations of the scriptware and the media objects. Further, scripts must be able to reference structured media objects for attribute control, retrieval, and presentation.

3. **Storage efficiency:**

   An encoding should be efficient for storage, but the container is a small fraction of the information in a typical multimedia presentation.

4. **Access efficiency:**

   An encoding should be efficient for time-constrained and resource-limited retrieval. Enhanced functions for progressive and multi-resolution delivery, flexible storage organization, media interleaving, index tables, and partial media referencing can support this goal.

5. **Portability**:

   GUI and platform architecture independence are essential, preferably without penalizing interchange on a single platform. Issues include look and feel independence, input architecture independence, file and object referencing, byte ordering, and data type encoding.

6. **Extensibility**:

   It should be possible to add new media formats, new media attribute, and other container extensions

13.3 QUICKTIME MOVIE FILE (QMF) FORMAT

QuickTime is a multimedia extension for Apple's System 7 operating system for the Macintosh personal computer. The QuickTime Movie File is a published file format for storing multimedia content for QuickTime presentation. Several QuickTime players are available for other platforms.

QMF uses a track model for organizing the temporally related data of a movie (Figure 13.2). A movie can contain one or more tracks which can be overlayed. A track is a time ordered sequence of a media type; the media is addressed using an edit list (Figure 13.3).

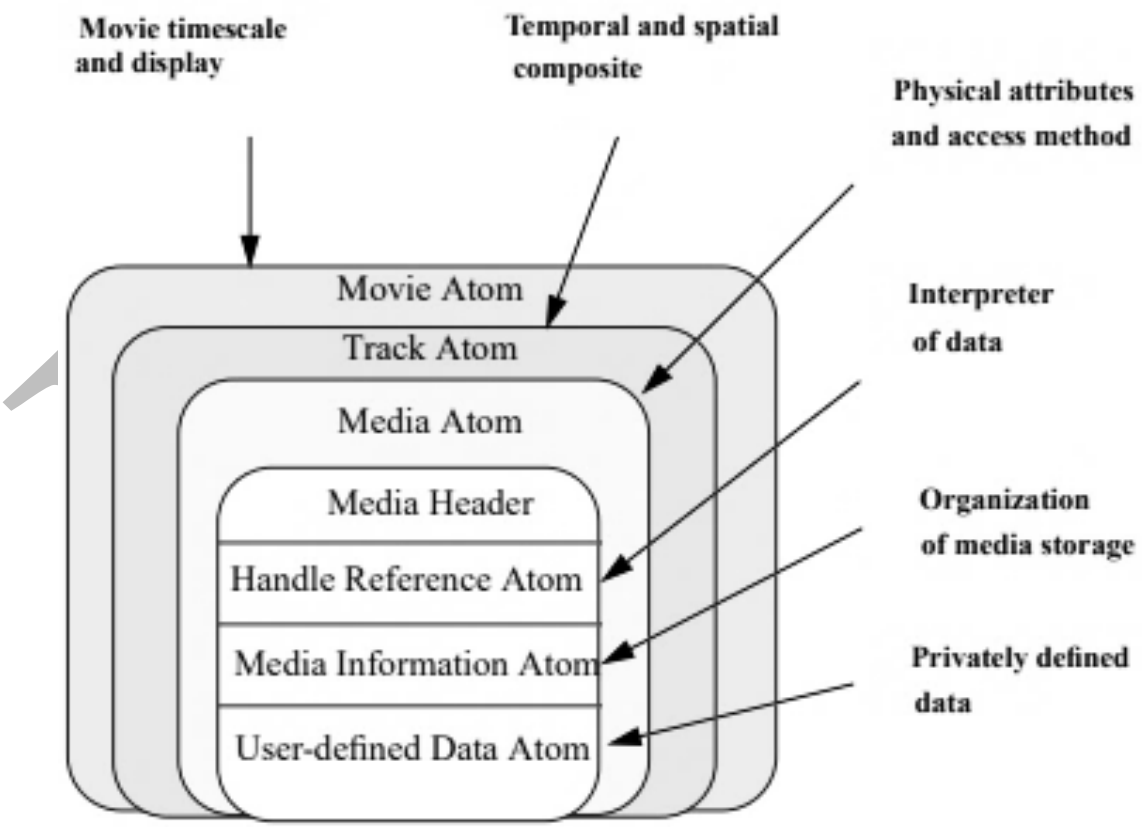


Figure 13.2 QuickTime™ abstract atom model

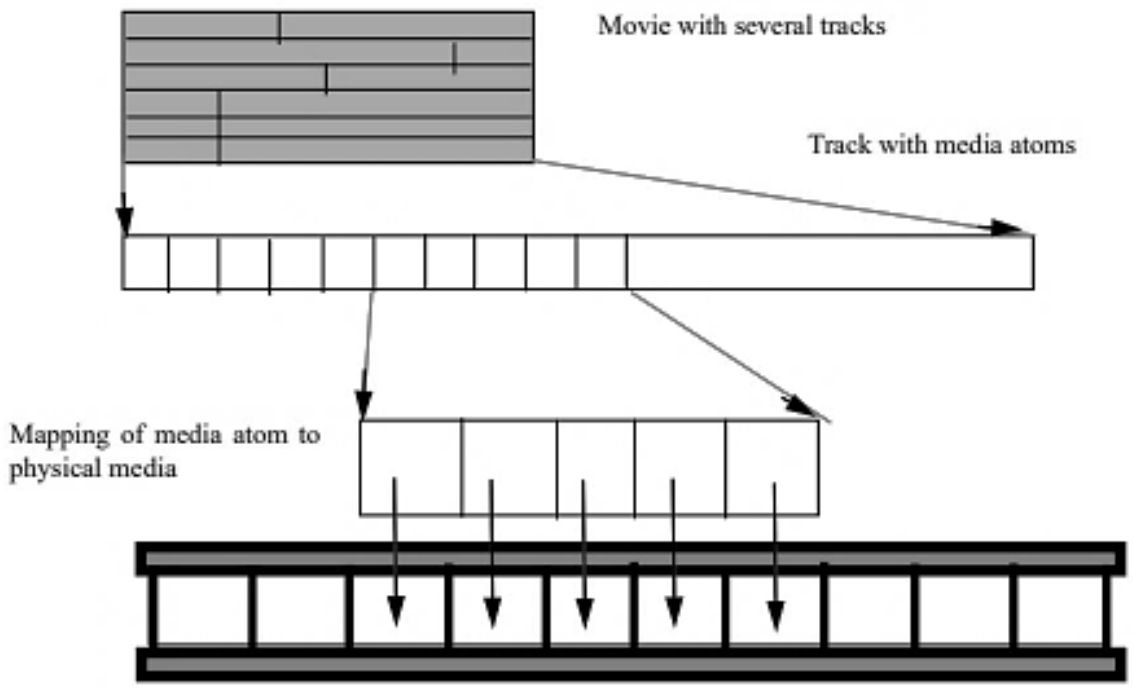


Figure 13.3 Components in QMF

13.4OPEN MEDIA FRAMEWORK INTERCHANGE (OMFI)

The Open Media Framework (OMF) is an industry standardization effort being led by Avid Technology to define a common framework and multimedia interchange format (OMFI).

The top-level composition elements are media objects (MOBS) (Figure 13.4). The logical MOB is the most important from the standpoint of composition. Media can be organized in parallel time (Track Group) or serial (Sequence). Like QMF, OMFI uses a track model (Figure 13.5).

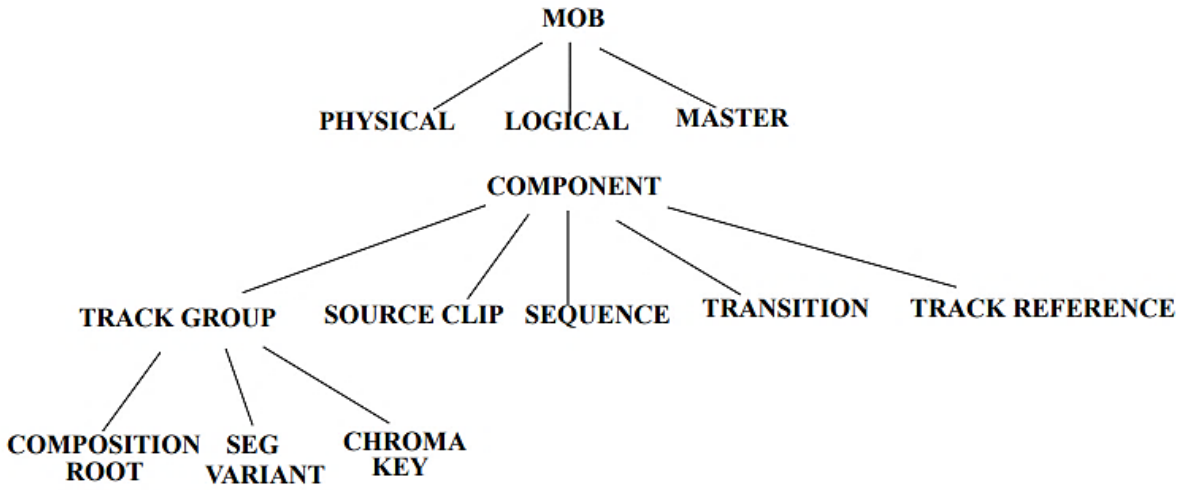


Figure 13.4 OMFI media object relationships

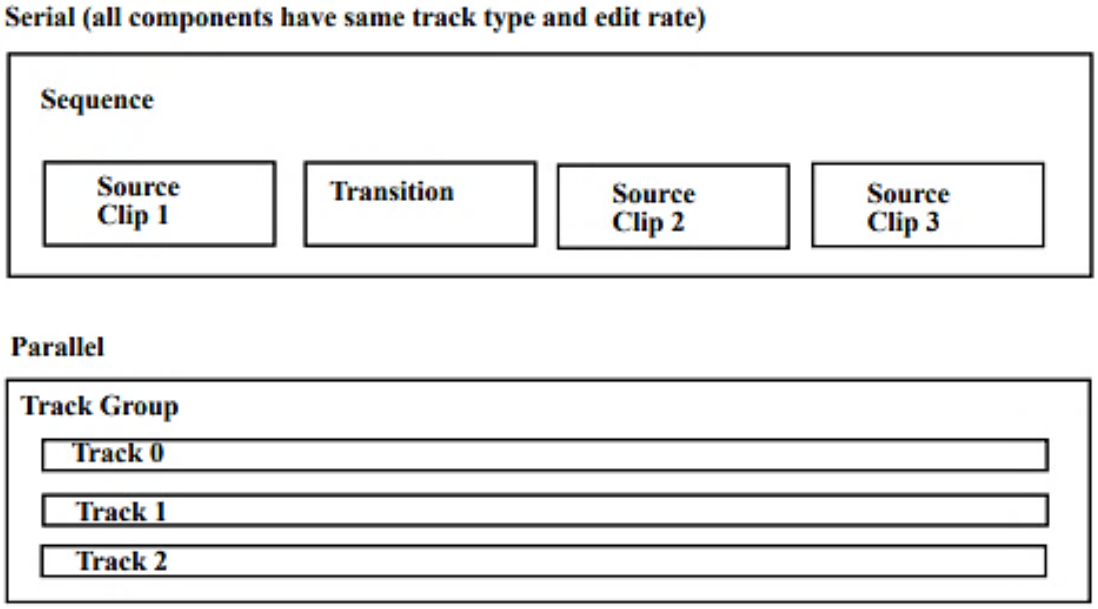


Figure 13.5 OMFI Serial and parallel temporal composition examples

13.5Multimedia and Hypermedia information encoding Expert Group (MHEG)

MHEG is an ISO working group that is defining an object-oriented model for multimedia and hypermedia interchange. The interchange model is intended for real-time final-form delivery environments. Here real-time refers to the need for systems to meet time constraints inherent in the media given the

resources of the delivery and presentation system. Final-form refers to the content and composition format being oriented towards delivery as opposed to authoring. MHEG uses an object-oriented model for composing multimedia (Figure 13.6).

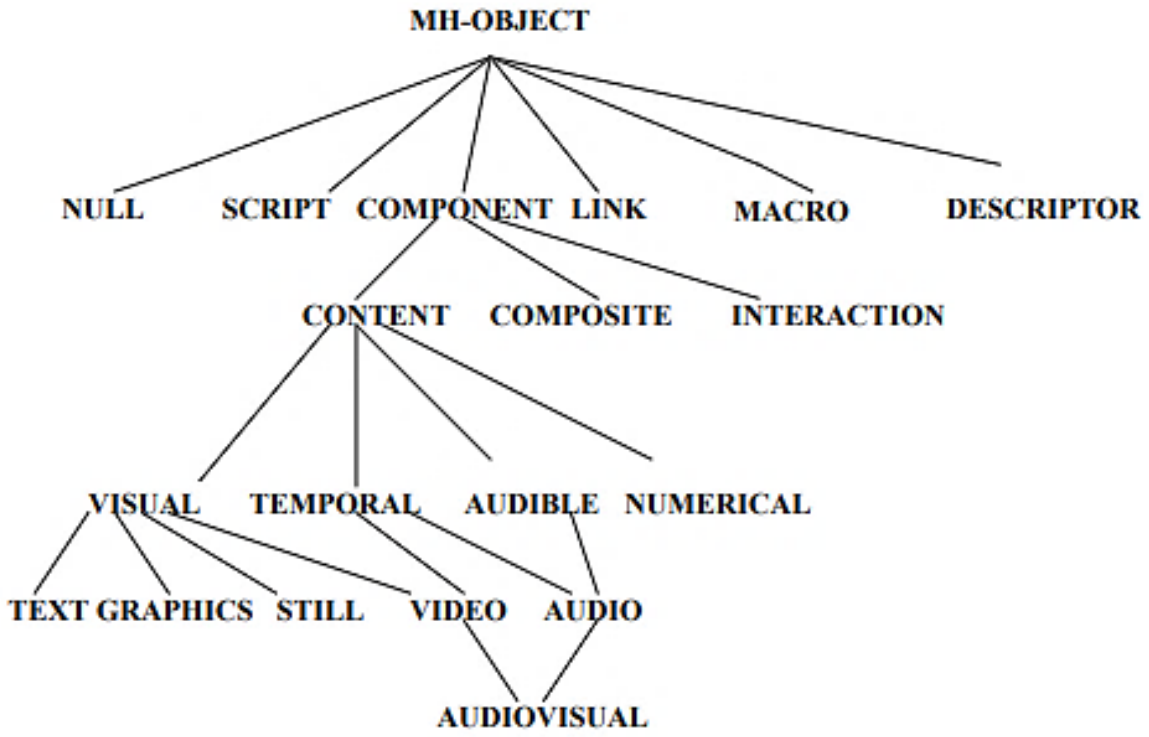


Figure 13.6 MHEG Object Tree

The composite class is used to associate objects (components) which have temporal or spatial relationships. Two binary temporal relationships are supported: serial and parallel. The link object defines one or more condition-action entries in the context of an interactive composite. This can be used to provide elementary input semantics or implement a hyperlink mechanism.

For more elaborate processing, script objects can be associated with specific objects. A GUI-independent input model is associated with the interaction class. Generic input types such as push button and text entry are defined.

13.5 FORMAT FUNCTION AND REPRESENTATION

Each of the three formats described previously has been developed with different but overlapping requirements. The formats deal with a complex problem: how to efficiently represent and encode the broad range of multimedia compositions in a way that is not dependent on a particular approach to multimedia or a particular presentation format. The format should be extensible to handle new requirements for multimedia content as they occur.

FEATURE	QMF	MHEG	OMFI
Model	Track	Object-oriented	Track
Encoding	Unique	ASN.1; others	IFF
Media object addressing	File name	Globally unique object id	Globally unique object id
Composition primitives	Movies, tracks, media	Composites, interactors, links	MOBS, tracks
Time Composition	Serial and parallel	Both serial and parallel	Sequence for serial, group for parallel
Component reference scope	Local for format units, global for media units	Global: All format units are objects and all have unique id	Nested for format units, global for media units
Media source referencing	No	No	Unique physical MOB
Input model	None	Interactor and link objects	None
Link model	None	Use of link class	None
Scriptware objects	Yes, if treated as media with own handler	Yes	Could be treated as media
Architecture independence	No, QuickTime and Mac dependencies	Yes, standard encoding and external referencing	Yes, handles byte ordering, data types, file names
Extensibility	By Apple, user defined atoms available	Yes, private data, private classes, private attributes	By members of OMF
File organization optimization	No specific features	Global object index	No specific features

Table 13.2 Functional comparison of the three formats

13.6 TRACK MODEL AND OBJECT MODEL

From the standpoint of media access, the composition model is a presentation list which defines the order in which media are accessed and displayed. In the track model, the primary access sequence is temporal. In an object model, the primary access sequence is through hierarchical descent of the tree; the ordering of the components in a composite would probably be based on display order. In the track model, multimedia presentation is viewed as a sequence of temporally oriented movie segments (Figure 13.7). In the object model, multimedia presentation is viewed as non-linear hyperlinking between (temporally) composite object trees (Figure 13.8).

The object model has the feature that all format units (containers, media objects, etc.) are fully visible and addressable units. The track models present provide addressing for media units but not for other container units. The trade-off here is between the overhead of providing object ids for each format unit versus the possibility of reusing container structure by allowing it to be referenced from several places.

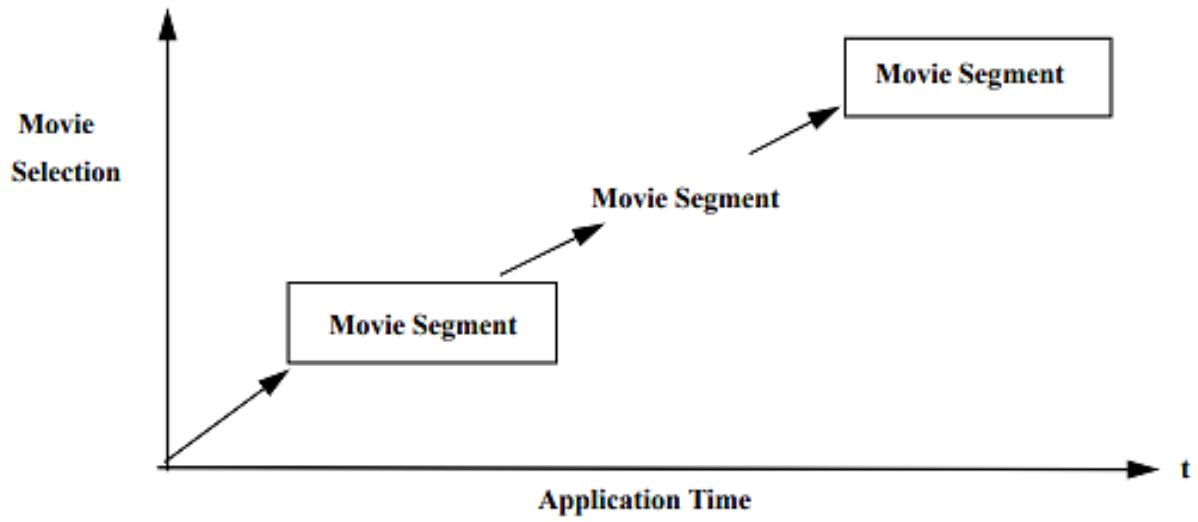


Figure 13.7 Multimedia presentation viewed as a sequence of temporally-oriented movie segments

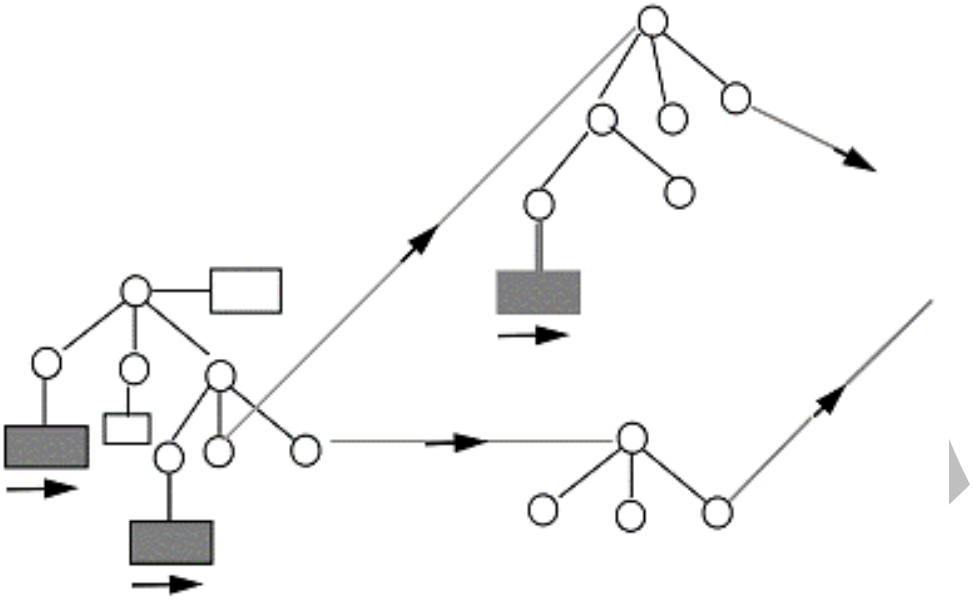


Figure 13.8 Multimedia presentation viewed as non-linear hyperlinking between (temporally) composite object trees

The implied presentation procedure for these two models differs as well (Programs1 and 2). The two track formats do not model input, leaving interaction support to scriptware or the application. Then the presentation process can be seen as clock-based control of a collection of media players (Program 1). In the object model, the input model allows some predefined input interpretation to be included in each composition, as well as providing links to the application and the script (Program 2).

```
// Open
Initialize clock
Initialize movie, track and media indices
Initialize handlers
// Players
Repeat
    Get next segment of media call handlers
    Advance clock
Until input condition or end of movie
```

Program 1: Abstract presentation procedure for track model

```
// Open
Get table of contents, object tables
Initialize presentation processes
//Present
Repeat
    Repeat
        Get set of objects to be presented
        Invoke players for each object
    Until media event or input condition
    Pass input condition to application or script or interpret
    Identify next set of objects to present
Until exit condition
```

Program 2: Abstract presentation procedure for object model

13.7 REALTIME INTERCHANGE

13.7.1 Definitions

For interchange during presentation to satisfy realtime, the temporal nature of the systems, objects, etc. are delivered or executed in a way that follows the original time-base as specified by the designer of the system. This implies that object retrieval, presentation, and response to user input must meet certain deadlines to be considered “realtime”. However, minimal configurations are allowed to present subsampled time-based data (e.g., reduced rate video) and still satisfy this “realtime” criteria. The interchange format can provide support for realtime by providing file and data organization techniques which enable a delivery engine to have faster access to objects. Some selected techniques are listed in the next section. Some of these techniques are independent of the format; all could potentially be available to the delivery system by preprocessing.

13.7.2 File Format Techniques for Supporting Realtime Interchange

1. **Object Placement Optimization:** Objects are stored so that objects which are likely to be accessed simultaneously are adjacent from the standpoint of the access mechanism.

2. **Partial Object Retrieval**: Large objects can be retrieved in sections since in many cases the entire content of such an object will not be presentable at one time.

3. **Object Sequencing**: The order in which objects are expected to be presented is maintained for use by the access mechanism.

4. **Global Object Index**: A table of all object ids and their position in the object set is provided to support fast lookup of objects.

5. **Object Interleaving**: Objects which are to be retrieved simultaneously are interleaved so that large objects don't cause delays for other objects.

6. **Separate Retrieval of Object Description and Object Content**: The object description can be retrieved without necessarily retrieving the content so that the system can use information about a set of objects to optimize the access for this set and so that resources needed for the access can be prepared.

7. **Progressive access of objects**: images can be retrieved and presented in increasing resolution for systems in which presentation delay is significant. Scalable versions of objects can be represented and retrieved for systems with insufficient resources for full fidelity presentation.

8. **Resource recommendations**: The resource requirements for retrieval and presentation by the target system are available by lookup rather than by derivation.

13.8 SUMMARY

- Many authoring tools integrate multimedia data with specialized procedural scriptware which may be text based or iconic languages

- The QuickTime Movie File is a published file format for storing multimedia content for QuickTime presentation

- The Open Media Framework (OMF) is an industry standardization effort being led by Avid Technology to define a common framework and multimedia interchange format (OMFI

- MHEG is an ISO working group that is defining an object-oriented model for multimedia and hypermedia interchange

- The composite class is used to associate objects (components) which have temporal or spatial relationships.

- The object model has the feature that all format units (containers, media objects, etc.) are fully visible and addressable units.

- QuickTime is a multimedia extension for Apple's System 7 Operating System for the Macintosh personal computer.

13.9Unit End Exercises

1. Explain the modes of Information Interchange in Multimedia.
2. What are the Major Technical issues in the design of multimedia interchange?
3. Write a note on QuickTime Movie File (QMF) format.
4. What is Open Media Framework Interchange?

5. Explain in brief: Multimedia and Hypermedia information encoding Expert Group.
6. Write a comparison note on QMF, MHEG and OMFI.
7. What is a track model and object model?
8. What are the file format techniques to support real-time interchange?

13.10Additional Reference

- *Multimedia Systems* by John F. Koegel Buford- Pearson Education
- Research Paper: *On the Design of Multimedia Interchange Formats,* by John F. Koegel available online at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.5430&rep=rep1&type=pdf
- *Multimedia Integration, Interaction and Interchange* study material by Dave Marshall available online at https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node284.html

14

Multimedia Conferencing

Unit Structure

14.1 Objectives
14.2 Introduction
14.3 Teleconferencing Systems
14.4 Requirements of Multimedia Communications
14.5Shared Application Architecture and embedded Distributed Objects
14.6 Multimedia Conferencing Architecture
14.7 Summary
14.8 Unit End Exercises
14.9 Additional Reference

14.1 OBJECTIVES

In this Chapter you will understand:

- Types of Teleconferences
- Categories of Quality-of-Service parameters
- Multimedia Conferencing Architecture

14.2 INTRODUCTION

Teleconferencing systems have become a critical part of many businesses today. That's because they make it easier for employees, clients and business partners to connect instantly from any point in the world. A typical teleconference system can allow exchange of information through audio, video or any other data service. Some of the most common means through which people can connect include telephone, computer, radio, telegraph or

teletypewriter. For any type company or business, choosing a robust video teleconference system is very important.

14.3 TELECONFERENCING SYSTEMS

Teleconferencing means meeting through a telecommunications medium. It is a generic term for linking people between two or more locations by electronics. There are at least six types of teleconferencing: audio, audiographic, computer, video, business television (BTV), and distance education. The methods used differ in the technology, but common factors contribute to the shared definition of teleconferencing:

- Use a telecommunications channel
- Link people at multiple locations
- Interactive to provide two-way communications
- Dynamic to require users' active participation

Interactive Technologies

The new systems have varying degrees of interactivity - the capability to talk back to the user. They are enabling and satellites, computers, teletext, view data, cassettes, cable, and videodiscs all fit the same emerging pattern. They provide ways for individuals to step out of the mass audiences and take an active role in the process by which information is transmitted. The new technologies are de-massified so that a special message can be exchanged with each individual in a large audience. They are the opposite of mass media and shift control to the user.

Many are asynchronous and can send or receive a message at a time convenient for individuals without being in communication at the same time. This overcomes time as a variable affecting communication. A video, data and voice delivery system reduce travel costs. When the material is

retrieved and saved to a video tape or disc, the material can be used at any time or anyplace.

As more interactive technologies emerge, the value of being an independent learner will increase. Research shows that learning from new technologies is as effective as traditional methods. Large groups are cost-effective and everyone gets the same information.

Types of Teleconferences

Audio Teleconference:

Voice-only; sometimes called conference calling. Interactively links people in remote locations via telephone lines. Audio bridges tie all lines together. Meetings can be conducted via audio conference. Preplanning is necessary which includes naming a chair, setting an agenda, and providing printed materials to participants ahead of time so that they can be reviewed.

Distance learning can be conducted by audio conference. In fact, it is one of the most underutilized, yet cost effective methods available to education. Instructors should receive training on how to best utilize audio conferences to augment other forms of distance learning.

Audiographics Teleconference:

Uses narrowband telecommunications channels to transmit visual information such as graphics, alphanumeric, documents, and video pictures as an adjunct to voice communication. Other terms are desk-top computer conferencing and enhanced audio. Devices include electronic tablets/boards, freeze-frame video terminals, integrated graphics systems (as part of personal computers), Fax, remote-access microfiche and slide

projectors, optical graphic scanners, and voice/data terminals.Audiographics can be used for meetings and distance learning.

Computer Teleconference:

Uses telephone lines to connect two or more computers and modems. Anything that can be done on a computer can be sent over the lines. It can be synchronous or asynchronous. An example of an asynchronous mode is electronic mail. Using electronic mail (E-Mail), memos, reports, updates, newsletters can be sent to anyone on the local area network (LAN) or wide area network (WAN). Items generated on computer which are normally printed and then sent by facsimile can be sent by E-Mail.

Computer conferencing is an emerging area for distance education. Some institutions offer credit programs completely by computer. Students receive texts and workbooks via mail. Through common files assigned to a class which each student can assess, teachers upload syllabi, lectures, grades and remarks. Students download these files, compose their assignment and remarks off-line, then upload them to the common files.

Students and instructors are usually required to log on for a prescribed number of days during the week. Interaction is a large component of the students' grades.

Through computers, faculty, students and administrators have easy access to one another as well as access to database resources provided through libraries. The academic resources of libraries and special resources can be accessed such as OCLC, ERIC, and Internet.

Administrators can access student files, retrieve institutional information from central repositories such as district or system offices, government

agencies, or communicate with one another. Other resources can be created such as updates on state or federal legislation.

Video Teleconference:

Combines audio and video to provide voice communications and video images. Can be one-way video/two-way audio, or two-way video/two-way audio. It can display anything that can be captured by a TV camera. The advantage is the capability to display moving images. In two-way audio/video systems, a common application is to show people which creates a social presence that resembles face-to-face meetings and classes and enables participants to see the facial expressions and physical demeanour of participants at remote sites. Graphics are used to enhance understanding. There are three basic systems**: freeze frame, compressed,** and **full-motion video**.

Video conferencing is an effective way to use one teacher who teaches to a number of sites. It is very cost effective for classes which may have a small number of students enrolled at each site. In many cases, video conferencing enables the institution or a group of institutions to provide courses which would be cancelled due to low enrolment or which could not be supported otherwise because of the cost of providing an instructor in an unusual subject area. Rural areas benefit particularly from classes provided through video conferencing when they work with a larger metropolitan institution that has full-time faculty.Through teleconferencing, institutions are able to serve all students equitably.

Why Use a Teleconference?

Videoconferencing increases efficiency and results in a more profitable use of limited resources. It is a very personal medium for human issues where

face-to-face communications are necessary. When you can see and hear the person you are talking to on a television monitor, they respond as though you were in the same room together. It is an effective alternative to travel which can easily add up to weeks of non-productive time each year. With videoconferencing, you never have to leave the office. Documents are available, and experts can be on hand. A crisis that might take on major proportions if you are out of town, can be handled because you're on the job. Videoconferencing maximizes efficiency because it provides a way to meet with several groups in different locations, at the same time.

As the limited resource of funding has decreased, limited resources now include instructors, parking spaces and buildings. Students now include time as a limited resource. Teleconferencing enables institutions to share facilities and instructors which will increase our ability to serve students.

Move Information - Not People

Electronic delivery is more efficient than physically moving people to a site, whether it is a faculty member or administrator.

Save Time: Content presented by one or many sources is received in many places simultaneously and instantly. Travel is reduced resulting in more productive time. Communication is improved and meetings are more efficient. It adds a competitive edge that face-to-face meetings do not.

Lower Costs: Costs (travel, meals, lodging) are reduced by keeping employees in the office, speeding up product development cycles, improving performance through frequent meetings with timely information.

Accessible: Through any origination site in the world. Larger Audiences: More people can attend. The larger the audience, the lower the cost per person.

Larger Audiences: More people can attend. The larger the audience, the lower cost per person.

Adaptable: Useful for business, associations, hospitals, and institutions to discuss, inform, train, educate or present.

Flexible: With a remote receive or transmit truck, a transmit or receive site can be located anywhere.

Security: Signals can be encrypted (scrambled) when it is necessary. Encryption prevents outside viewers.

Unity: Provides a shared sense of identity. People feel more a part of the group...more often. Individuals or groups at multiple locations can be linked frequently.

Timely: For time-critical information, sites can be linked quickly. An audio or point-to-point teleconference can be convened in three minutes.

Interactive: Dynamic; requires the user's active participation. It enhances personal communication. When used well for learning, the interactivity will enhance the learning and the teaching experience.

Satellite Communications

Long distance telephone calls, national and international televised sporting events, and cable movie channels operate via satellites. Satellites have been used for years.

Geostationary Orbit: British physicist and science fiction writer, Sir Arthur C. Clarke, invented satellite communication in his 1954 paper Wireless World, which explained this east-west orbit, 22,300 miles above the equator; three satellites based in this orbit could provide world-wide communications. Today, many satellites are arrayed in the Clarke belt. To earth stations, they appear fixed in space.

Satellite Footprint: In geostationary orbit, communications satellites have direct line-of-sight to almost half the earth - a large "footprint" which is a major advantage. A signal sent via satellite can be transmitted simultaneously to every U.S. city. Many downlinks can be aimed at one satellite and each can receive the same program; this is called point to multipoint.

Transponders: Via an uplink, video, audio or data signals can be transmitted to a satellite transponder. There may be up to 40 transponders per satellite; each can amplify and relay signals to earth which are picked up by earth stations.

C/Ku-Band: Domestic communications satellites operate on two frequency ranges designated C- and Ku-band. Each requires specific electronic equipment. C-band is less expensive; operates at 4 kHz. Ku-band operates at 12 kHz. Some teleconferences are broadcast on both bands.

Receivers: Convert satellite signals into channels viewed (one at a time) on a TV monitor; designed to tune-in the format, bandwidth, and audio sub-

carrier. Programs broadcast in code (encryption) are decoded at receive sites.

Basic Receivers: Lowest cost; limited (or manual) channel tuning capability; may use fixed antennas.
Multi-Format Receivers: Most versatile; adjusts for all broadcast formats; receive any satellite video program in six or more bandwidth selections, and two agile audio subcarrier switches; usually a motorized system.

Fixed Position System: Low cost systems limited to reception from one satellite and one band.

Motorized System: Receives programs on different satellites by adjusting the dish position.

Automated Systems: Microprocessor controlled for instant movement to satellites (positions stored in memory).

International Satellite

Alpha Lyracom Space Communications/Pan American Satellite is the world's first private international satellite system. PAS-1 carries many specialized communications services including full and part-time video, low and high-speed data, broadcast data and radio and business television to over 70 countries on three continents. It can be seen (received) by a 2.4-meter antenna. It has 18 C-band and six Ku-band transponders with a shared capacity that increases traffic.

PanAmSat handles all phases of an international broadcast as compared to INTELSAT (International Telecommunications Satellite Organization) where the customer must book the domestic and foreign half circuits and

pay for each downlink. INTELSAT was established primarily to handle the PTT telephone transmissions, while PanAmSat was established to be easily accessible by distance education institutions and private enterprise. The FCC licenses PanAmSattransportables for years, as compared to the FCC special temporary authority (STA) license for INTELSAT. PanAmSattransportable can uplink from any location without a special license.

PanAmSat writes yearly contracts with customers. It does not charge for multiple downlinks. Time on PAS-1 books from between $960 to $2,400 per hour depending on the volume discount based on yearly usage. To book time on PAS-1, call the day-of-air or future event number, with the origination site, uplink, downlink sites, and conference time. PanAmSat handles the rest. By booking time through satellite brokers (EDS, PSN, Satellite Management International) ad hoc users can reduce time costs. PanAmSat is negotiating for three more satellites to be in place in 1994-95.

Compressed Video

Digital compression means that the codec compresses the video signal or data to a fraction of its original size so that the data rate is appropriate to transmit over low-cost terrestrial telephone lines or on a fraction of a satellite transponder. Codecs (COder/DECorder) compress the video and audio signal allowing it to be transmitted in a smaller bandwidth which reduces the cost of the transmission.

Standard transmission rates for video teleconferencing are multiples of 64 Kbs up to the T1 rate of 1.54 Mbs. Some codecs allow speed selection to match the circuit used. The speed selected is based on the content. When close to full motion video is needed, higher rates are needed.

T1 circuits connect PBXs to the telephone company's central office and can carry up to 24 voice channels at a lower cost than 24 voice circuits. A 56 Kb or 64 KBS codec operates in the range of one voice channel. A standard video signal digitized at 90 Mbs is comprised of about 1400 voice channels.

Freeze Frame Video

Freeze frame video uses telephone channels to transmit video information. Because of the narrow bandwidth, the image takes a few moments to reach the receive site where it appears on the TV as a still picture. The advantages are lower costs and flexibility in linking multiple sites. Slow scan systems are similar to freeze frame and the terms are often used synonymously.

Freeze frame technologies include a range of features; analog, digital, monochrome or color pictures, resolutions, transmission speeds, and extra memory. Newer models provide multiple send times to select the resolution and transmission time through digital circuits and compression coding. Some units transmit video information in digital format over a data circuit which reduces the transmission time to about nine seconds to a 56-kilobit link. Because of the faster transmission rates, many new freeze frame applications use data circuits.

Compressed video (near motion) and full-motion video differ; compressed video uses compression techniques to reduce channel bandwidth; images may not look as natural and may blur or lose background resolution. The advantage is that the significant reduction in bandwidth reduces costs. Compressed video uses a telephone data circuit - currently a T1 carrier or 1.5 or 3 megabits - to transmit video, voice and data. It reduces video information (NTSC Standard-color video) with a compression technique to

eliminate redundant information and reduce the 100 million bits signal to 1.5 or 3 million bits.

Digital video signals are broken down into thousands of elements called pixels. Between frames, many are the same. A codec takes advantage of this duplication by sending complete information on the first pixel and a brief code to repeat the values. This reduces the information sent and the bandwidth required. Interframe coding for conditional replenishment compares the changes between two frames and transmits changes. Motion compensation predicts changes between frames and transmits only the difference. Software holds the compression algorithm which can be upgraded. The CCITT Px64 international standard requires rates to operate in multiples of 64.

Full-Motion Video

Standard TV signals are broadcast using a significant amount of the bandwidth of wideband channels - 4 to 6 megahertz for color analog - to send video, voice and data. Because of the large channel capacity, it transmits a picture with the full motion and resolution of broadcast TV. The bandwidth used is the digital equivalent of 80 Mbps or more which corresponds to a full satellite transponder or 1820 voice phone lines. This translates into high costs for signal transmission.

Compression for One-Way Video

Consumer application for compressed video systems use higher rates than two-way compressed video to achieve near-broadcast quality video image. A digitally compressed video signal can be broadcast over 1/20 of a regular transponder channel reducing costs to under $200 per hour.

One use of the technology is SKY PIX, a pay per view movie service based on a Compression Labs, Inc. codec marketed by NW Star Scan which offers viewers a choice of up to 40 movies. The picture quality is better than VHS transmission quality. Scientific Atlanta offers PrimeStar, a competing entertainment service, which transmits at a data rate of 4 to 4.5 Mbs. Using the same technology, they will offer B-Mac users compatibility with compressed video users at a lower price because the transmission uses a fraction of a regular transponder channel.

Compression Labs, Inc. has recently introduced the SpectrumSaver System which can broadcast a digital signal to a fraction of a satellite transponder. Because up to 15 or 18 signals can be carried on a transponder (depending upon the system configuration), the cost of satellite time is significantly reduced. The National Technological University (NTU) is using the system, as well as ITESM in Mexico. Each institution reports a savings of $1 million in satellite time during the first year of operation. The system is entirely digital.

Scientific Atlanta is about to bring its new digital satellite system to the market. This system is an upgrade to an existing Scientific Atlanta analog satellite system. As such, users will be able to broadcast in either analog or digital format.

Fiber Optic Systems

The transmission of voice, video and data by light wave signals inside a thin, transparent glass fiber cable, is providing more choices for telecommunications users and is rapidly bringing digital communication to the home and office.

One pair of fibers can carry up to 10,000 telephone calls simultaneously. Advantages: transmission clarity, speed, accuracy, security, and volume.

Disadvantages: Construction, installation and maintenance costs, but they are declining.

14.4REQUIREMENTS OF MULTIMEDIA COMMUNICATIONS

Recent experience with the Internet indicates that it is not well suited to time and loss sensitive multimedia applications such as voice and video. To support multimedia applications, the following six network criteria are critical:

- throughput
- transit delay
- delay variation
- error rate
- multicasting and broadcasting capabilities
- document caching capabilities

These network criteria are closely associated with quality of service(QoS). According to ITU-T Recommendation E.800, the QoS is defined as follows:*The QoS is the collective effect of service performances which determine the degree of satisfaction of a user of the service.*

This implies that the user is the final arbiter of 'good' or 'bad' QoS. Different applications demand different service qualities. Some need minimal delay and reliable response time, while others may need a good image quality. Table 14.1 summarises the five categories of QoS parameters.

The QoS is a difficult issue in that the relationship between application QoS parameters and network QoS parameters is very complex; QoS must be end-to-end; and the application QoS might change during connections.

Category	Example Parameters
Performance-oriented	end-to-end delay and bit rate
Format-oriented	video resolution, frame rate, storage format, and compression scheme
Synchronisation-oriented	skew between the beginning of audio and video sequences
Cost-oriented	connection and data transmission charges and copyright fees
User-oriented	subjective image and sound quality

Table 14.1 The five categories of QoS parameters

14.5SHARED APPLICATION ARCHITECTURE AND EMBEDDED DISTRIBUTED OBJECTS

Distributed Application Architecture

The Distributed Application Architecture (DAA) is designed to allow users of a computer network to access information, applications, and services, as well as to exchange information with others, through a single, consistent user environment. It enables the construction of new applications and services; it also provides facilities for the integration and migration of existing applications.

A complete system based on the DAA includes both components which supply services provided as part of the infrastructure and a set of conventions or policies defining how components are to interact with the

provided services and each other. These conventions, in particular, enable the integration of components in an enterprise-wide context.

Shared Application Architecture

Application sharing is an element of remote access, falling under the collaborative software umbrella, that enables two or more users to access a shared application or document from their respective computers simultaneously in real time.

Generally, the shared application or document will be running on a host computer, and remote access to the shared content will be provided to other users by the host user. To transfer one application from one computer to another, the application must reside on only one of the machines connected with each other.

Granting Access

Access is typically granted in one of three ways, depending on the architecture of the application sharing software.

1. If the software allows the shared content to be accessed from the web, the host user will usually define and provide a username/password combination to the remote users he/she wishes to grant access to; they can then enter the log-in information on the appropriate website and access the shared material. One example of software that features application sharing in this manner is Qnext.

2. If the software is required on both ends to access the shared content, granting access will be governed by the mechanisms of that particular software, but will usually require some sort of user

authentication as well. One example of software that featured application sharing in this manner was MSN Messenger.

3. The shared content (being an application or entire desktop) can be accessed using a permission-based software approach. This technique helps to ensure that the application or desktop being controlled cannot be accessed without direct live approval, helping to eliminate the security risk of taking control of a desktop when the user is not present.

Type of Access

Once the applications or documents to be shared and whom they are to be shared with have been determined, there are generally two types of access that can be granted to remote users.

1. **Control access** – the host user allows remote users to actually control, edit, and manipulate the shared content; most application sharing software allows the host to revoke control access at any time. During the remote-control session, keyboard and mouse are remotely controlled. Usually a hot key is provided to revoke access.
2. **View access** – the host user only allows remote users to passively view the shared content; remote users have no ability to edit or effect change in the shared content whatsoever.

Embedded Distributed Objects

Because the design of embedded systems frequently is driven by performance, footprint security features are frequently omitted. Yet embedded systems and their inter-communications are often the most accessible target of the potential attackers.

The notion of embedded application objects developed from the desire to create documents which are active, that is, documents in which the displayed components are connected to content – in databases, files, or other applications – that can be independently manipulated. When the content changes, the views change as well. One common scenario used to illustrate this functionality is a spreadsheet object embedded in a document. Rather than just being a static table, the embedded spreadsheet has all the functionality of a real spreadsheet. Additionally, the spreadsheet can be dynamically connected to data, so that if the stored data is changed, the change propagates to the spreadsheet view.

In order to generalize the notion of an embedded spreadsheet so that embedding of any application component in another application is possible, new mechanisms are needed for distributed application communication and presentation toolkits. At the same time, the application itself changes from monolithic and impenetrable to being a collection of inter-related objects which can export services to other applications.

14.6MULTIMEDIA CONFERENCING ARCHITECTURE

This section is gathers together in one place the set of assumptions behind the design of the Internet Multimedia Conferencing architecture, and the services that are provided to support it.Figure 14.1 shows an example time sequence involved in setting up a light-weightsession between two sites. In this case, site A creates a session advertisement, and sometime later starts sending a media stream even though there may be no receiver at that time. Sometime later, site B joins the session (the multicast routing protocol here is PIM), and starts to receive the traffic.

At the earliest opportunity site B also makes an RSVP reservation to ensure the flow quality is satisfactory. This example should be taken as illustrative only - there are different ways to join sessions, and different ways to get improved quality of service. The lightweightsessions model for Internet multimedia conferencing may not be appropriate for all conferences, but for those sessions that do not require tightly-coupled conference control, it provides an elegant style of conferencing that scales from two participants to millions of participants. It achieves this scaling by virtue of the way that multicast routing is receiver driven, keeping essential information about receivers local to those receivers.

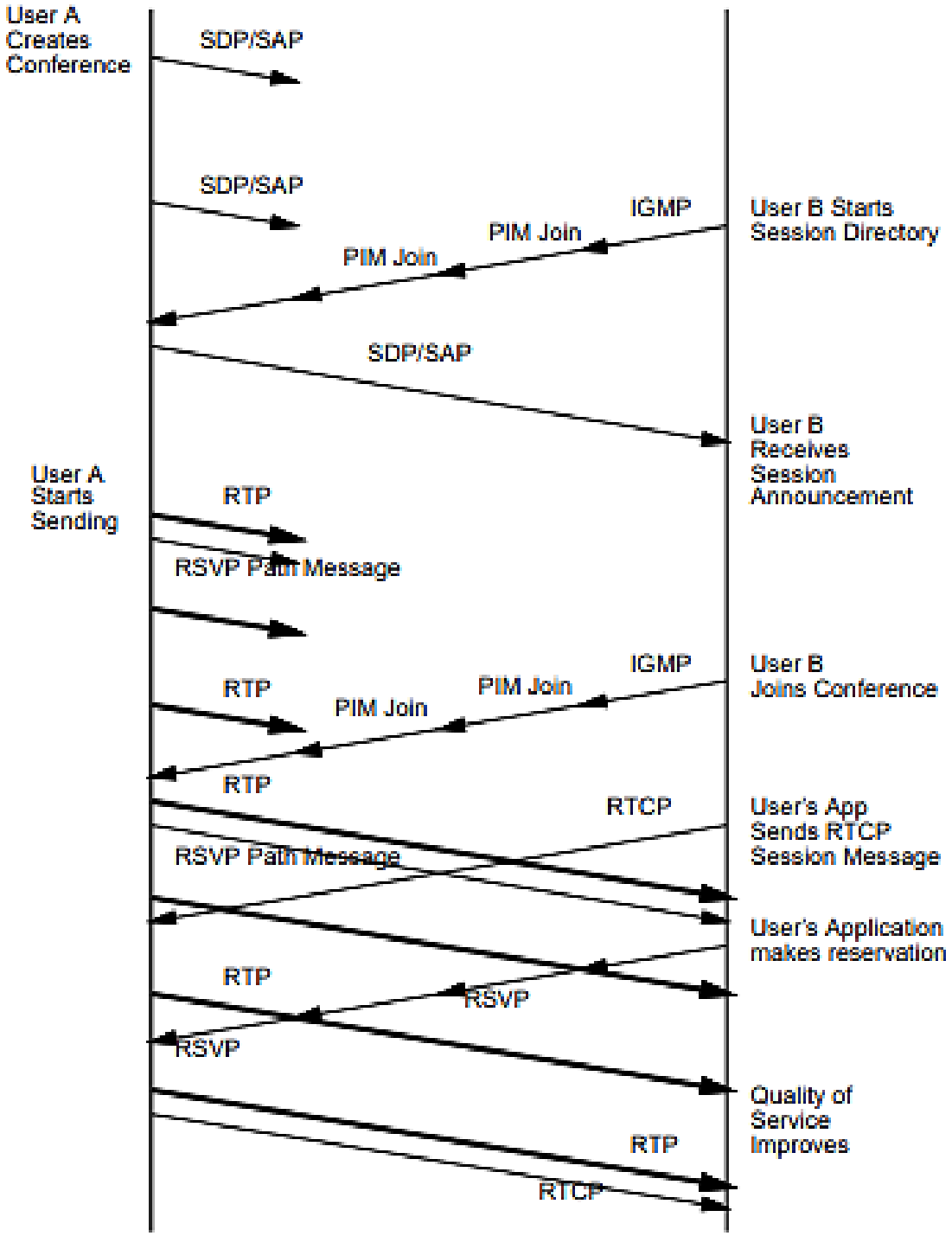


Figure 14.1 Joining a light-weight multimedia session

Each new participant only adds state close. to them in the network. It also scales by not requiring explicit conference join mechanisms; if everyone were to need to know exactly who is in the session at any time, the scaling would be severely adversely affected. RTCP provides membership

information that is accurate when the group is small, and increasingly only a statistical representation of the membership as the group grows. Security is handled through the use of encryption rather than through the control of data distribution.

14.7 SUMMARY

- A typical teleconference system can allow exchange of information through audio, video or any other data service
- There are three basic systems: **freeze frame, compressed,** and **full-motion video.**
- Freeze frame video uses telephone channels to transmit video information.
- There are generally two types of access that can be granted to remote users: **Control access**and **View access**

14.8Unit End Exercises

1. Explain different types of teleconferences.
2. Write a note on Satellite Communications.
3. What are the categories of QoS parameters?
4. Explain Shared Application Architecture in brief.

14.9Additional Reference

- Note on Teleconferencing, from "*The Distance Learning Technology Resource Guide*," by Carla Lane, available online at https://www.tecweb.org/eddevel/edtech/teleconf.html

- *Quality of Service Requirements for Multimedia Communications* by Xinping Guo and Colin Pattinson available online at https://www.hiraeth.com/conf/web97/papers/guo.html

- *The Distributed Application Architecture*by Joseph Sventek available online at https://pdfs.semanticscholar.org/2a38/514f081703a787ad1712606d27a517f61d8e.pdf

- Reference to Application sharing opted from Wikipedia page available at https://en.wikipedia.org/wiki/Applications_architecture

- Real-time and Embedded Distributed Object Computing Workshop material available online at https://pdfs.semanticscholar.org/04d5/98327e3924d7db93f47c00335ae1f36266eb.pdf?_ga=2.100970753.1278014659.1558291919-19577895.1558291919

- Research paper: *Very large conferences on the Internet: the Internet Multimedia Conferencing Architecture*, by M. Handley, J. Crowcroft, C. Bormann and J. Ott available online at https://www.cl.cam.ac.uk/~jac22/otalks/cnis.pdf

15

Multimedia Groupware

Unit Structure

15.1 Objectives
15.2 Introduction
15.3 Computer and Video fusion approach to open shared workspace
15.4 High Definition Television and desktop computing
15.5HDTV Standards
15.6 Summary
15.7 Unit End Exercises
15.8 Additional Reference

15.1 OBJECTIVES

In this Chapter you will understand:

- The concept of TeamWorkStation
- Advanced Television
- High-Definition Television computing
- HDTV Standards

15.2 INTRODUCTION

Groupware is intended to create a shared workspace that supports dynamic collaboration in a work group over space and time constraints. To gain the collective benefits of groupware use, the groupware must be accepted by a majority of workgroup members as a common tool. Groupware must overcome the hurdle of critical mass.

15.3 COMPUTER AND VIDEO FUSION APPROACH TO OPEN SHARED WORKSPACE

In order to provide distributed users with an "open shared workspace" where every member can see, point to and draw on simultaneously using heterogeneous personal tools, researchers designed "**TeamWorkStation**". **TeamWorkStation** integrates two existing kinds of individual workspaces: **computers** and **desktops**. Because each co-worker can continue to use his/her favourite application programs or manual tools simultaneously in the virtual shared workspace, the cognitive discontinuity (seam) between the individual and shared workspaces is greatly reduced.

TeamWorkStation (TWS) provides a "shared screen" in addition to an individual screen. The shared screen supports a shared drawing window for concurrent pointing, writing, drawing, and live face windows for face-to-face conversation.TeamWorkStation demonstrates the power of two different uses of the translucent overlay technique: fused overlay of drawing surface images for seamless shared workspace, and selective overlay to save screen space.

The new shared workspace is required to be "open," in the sense that no new piece of technology should block the potential use of already existing tools and methods. A new piece of technology inevitably introduces with it the burden of learning. It is also often coupled with the introduction of seams and discontinuities from the old work practices. The world is filled with many seams. The current variety of application programs running on the same or different platforms creates seams of incompatible data formats and inconsistent human-computer interfaces. These seams increase the users' cognitive load.

Cognitive Seamlessness Achieved in TeamworkStation

TWS was designed to bridge the gaps between personal computer, desktopand telecommunication. Through the experimental use of TWS described next, however, researchers found just a superficial view of seams and TWS functions. They realized that the essence of TWS approach is not the functional seamlessness but the cognitive seamlessness that is achieved in the following two points.

(1) Since TWS allows users to keep using their favorite individual tools (in whatever form) while collaborating in a desktop shared workspace, there is no need to master the usage of new sophisticated groupware.

(2) Because TWS's multiscreen architecture allows users to move any application program window between the individual and shared screens just by dragging the mouse, it is easy to bring the data and tools in each personal computer to the shared workspace. The information on paper and books can also be easily shared just bybringing them under the CCD camera attached to the desk lamp.

15.4HIGH DEFINITION TELEVISION AND DESKTOP COMPUTING

High definition television (HDTV) is defined as having twice the vertical and twice the horizontal resolution of conventional television, a picture aspect ratio of 16:9, a frame rate at least 24 Hz, and at least two channels of CD-quality sound. HDTV studio equipment commercially available at the moment has about two megapixels per frame (in a 1920×1035 format), six times the number of pixels as conventional television. The data rate of current studio-quality HDTV is about 120megabytes per second.

The parameters of HDTV are optimized for a viewing distance of about three times picture height. This enables a horizontal picture viewing angle of about

thirty degrees, three times that of conventional television. This gives the viewer a much greater sense of involvement in the picture.

.

Advanced Television (ATV) refers to transmission systems designed for the delivery of entertainment to consumers, at quality levels substantially improved over conventional television. ATV transmission systems have been deployed in Japan, and a number of systems have been proposed for adoption in the United States. The parameters of these systems vary widely, although the data rate of each proposed systems is about 20megabits per second.

Standards for film, video and entertainment have historically developed in a three-level hierarchy: production, exchange and distribution. Production refers to the shooting and editing of material. Exchange refers to the buying and selling of programs (often on 35mm film). Distribution refers to the delivery of programs to consumers, which can take place using transmission as in conventional broadcasting, or using physical media such as videotapes and videodiscs. Transmission can be through conventional terrestrial broadcasting (VHF/UHF), cable television (CATV), direct broadcast by satellite (DBS), or potentially through telecommunications (E.g."fibre to the home", FTTH).

The development of HDTV and ATV standards is currently in disarray. The standards process is confused among production, exchange and distribution. Certain organizations have much to lose if the adoption of standards accelerates the introduction of HDTV; these organizations have deterred standards development. The U.S. Federal Communications Commission has jurisdiction only over standards for terrestrial VHF/UHF broadcast, but has no jurisdiction and no official position on other delivery standards, and no mandate to discuss production or exchange standards. Political, techno-nationalist and industrial policy concerns are frequently evident in standards discussions. The lack of formal standards for HDTV and ATV could encourage

introduction of consumer ATV using physical media, since this would bypass the need for formal standards.

The television industry has reacted with alarm at the possibility of the introduction of ATV broadcasting. Networks and television stations face the prospect of making huge capital outlays to upgrade their plants for ATV, with no corresponding increase of revenue. Traditional consumer equipment manufacturers and traditional broadcasting interests have proposed ATV systems with parameters closely tied to existing practice, in the hope of minimizing new investment. For example, many U.S. interests hope to retain the same troublesome 59.94Hz field rate as NTSC just to be “compatible”. Unfortunately, their European counterparts have also chosen to maintain their field rate of 50Hz, and the discrepancy between these two rates makes it unlikely that a common distribution standard will emerge.

Despite the chaos in distribution standards, the Society of Motion Picture and Television Engineers (SMPTE) has adopted SMPTE Standard240M for 1125/60 studio equipment. The emergence of this standard has encouraged equipment manufactures to invest in tooling to bring studio production equipment to the market. Much commercial studio production equipment that conforms to this standard is available for acquisition, recording, processing, transmission and display.

HDTV is different from video. It has greater resolution (2Mpx vs. 300Kpx) and improved color accuracy. Its color is coded with the component system (instead of the composite system, which suffers color quality impairments). In its digital form, HDTV is poised to exploit the emerging digital infrastructure in a way that NTSC and PAL cannot.

HDTV is different from film: has no judder, no weave, and no scratches! Being electronic instead of photochemical, HDTV offers consistent, reliable color

reproduction compared to film, and has all the advantages of electronic, digital post-production. The image quality of HDTV is good enough for Hollywood: many feature productions are exploiting HDTV technology in effects sequences and synthetic computer graphics sequences.

HDTV is different from computer graphics. Commercial equipment is available today for real-time acquisition, recording, processing and transmission. HDTV technology is poised to bring motion and real-life images to computer graphics. HDTV interchange standards have been designed to deliver convincing, emotive, artifact-free pictures. The coding of pictures into HDTV signals takes into account the perceptual characteristics of human vision to make the most effective use of the available bandwidth, and a standardized set of interchange parameters has been adopted, so that HDTV images maintain their quality (including color accuracy) when exchanged.

Low cost HDTV and ATV equipment is inevitable as HDTV moves towards the high unit volumes that will come from consumer acceptance. Large consumer electronic companies recognize the difficulty of the consumer's encountering the "home theatre experience" on a direct-view CRT display at most 32inches in diameter, the largest practical size of tube for the home. So despite the immense technological difficulties, flat panel displays are likely to emerge for HDTV. The computer industry will be the first beneficiaries of these products — at a high price — but the goal of the manufactures is to reap profits from consumer unit volumes, not from the comparatively small volumes of the computer industry.

Quality

The picture quality of HDTV is superior to that of 35mm motion picture film, but less than the quality of 35mm still film. Motion picture film is conveyed vertically through the camera and projector, so the width - not the height - of the film is 35mm. Cinema usually has an aspect ratio of 1.85:1, so the

projected film area is about 21mm × 11mm, only three tenths of the 36mm × 24mm projected area of 35mm still film. In any case the limit to the resolution of motion picture film is not the static response of the film, but judder and weave in the camera and the projector.

The colorimetry obtainable with the color separation filters and CRT phosphors of a video system is greatly superior to that possible with the photochemical processes of a color film system. There are other issues related to the subjective impressions that a viewer obtains from viewing motion picture film - the film look- that are still being explored in HDTV. For example, specular highlights captured on film have an appearance that is subjectively more pleasing than when captured in video.

15.5HDTV STANDARDS

Standards for motion pictures and video exist in three tiers: **production, exchange,** and **distribution**.

Production is the shooting and assembling of program material. Exchange of programs takes place among program producers and distributors. Distribution to the consumer may take place using physical media such as videotape or videodisc, or through one of four transmission media: terrestrial VHF/UHF broadcast, cable television (CATV), direct broadcast from satellite (DBS) or telecommunications.

The film production community will produce material in the electronic domain only if it can be assured the access to international markets afforded by 24Hz, which translates easily to both 29.97Hz and 25Hz with minimal artifacts. Electronic origination at either 25Hz or 30Hz would introduce serious artifacts upon conversion to the other. It is my contention that a single worldwide

standard for HDTV production is feasible only if it accommodates distribution of material originated at 24Hz.

Broadcasters have recently made proposals to produce HDTV material in Widescreen-525 or Widescreen-625 production formats with a 16:9 aspect ratio. These proposals would use conventional video production equipment, modified for wide aspect ratio. This technique would allow broadcasters and perhaps even local stations to originate wide-aspect-ratio programming with minimum expenditure. However, programs would contain approximately the same amount of picture detail as conventional television, therefore the viewer could not take advantage of the wide viewing angle and the increased sense of involvement which in researchers opinion is the key to consumer differentiation of HDTV.

SMPTE240M: 1125/60 Production Standard

The technical parameters of the 1125/60 production system were standardized in SMPTEStandard 240M, adopted in February, 1989. Disclaimers on this document indicate that it is applicable to HDTV production only, although the MUSE broadcasting system in use in Japan is based on 1125/60 parameters. SMPTE240M applies to the 1125/60 analog signal. A digital representation of 1125/60 having a sampling structure of 1920×1035 and a sampling rate of 74.25MHz has recently been adopted in SMPTE Standard260M.

SMPTE 240M specifies RGB or YP_bP_rcolor components, with carefully-specified colorimetry and transfer functions. Luma (Y) bandwidth is specified as 30MHz, about five or six times the bandwidth of current broadcast television. Not all currently-available HDTV equipment meets this bandwidth, and most of the proposed transmission systems do not come close to the performance of the studio production equipment.

Although the field rate of SMPTE240M system is exactly 60 Hz - emphasized by the 60.00 notation in the document - certain organizations propose operating at a field rate of 59.94 Hz to maximize compatibility with existing NTSC equipment and production processes. Some current HDTV studio equipment is configurable for operation at either rate. There are several system problems with 59.94Hz field rate. In 59.94Hz operation with standard digital audio sampling frequencies of either 44.1kHz or 48kHz, there is not an integer number of audio samples in each frame. This, and the requirement for dropframe timecode, imposes a penalty on operation at that rate. However the production procedures for 59.94Hz are of course well established for conventional 525/59.94 video, troublesome though they may be.

1125/60 Studio Equipment

Commercial hardware operating with the 1125-line system is now widely available. Professional studio equipment that is commercially available for purchase includes:

- Cameras
- Videotape recorder (analog, 1inch open reel)
- Videotape recorder (digital, 1inch open reel)
- Videotape recorder (analog “Uni-Hi”, 19mm cassette)
- Videodisc
- Telecine (film-to-video)
- Film recorders
- Video monitors
- Video projectors
- Still and sequence stores
- Up-converters
- Line doublers
- Cross-converters
- Down-converters

- Production Switchers
- Graphics and Paint Systems
- Blue-screen matte (Ultimatte)
- Test equipment

Sony has demonstrated a fourth-generation 1125/60 CCD camera that has resolution, sensitivity and noise performance comparable to the best film cameras and motion picture film.A digital studio HD-VTR records a raw data rate of about 1.2 gigabits per second (Gb/s); this is the state-of-the-art for digital magnetic recording.

Real-time digital video effects equipment has been demonstrated by several manufacturers, but no commercial DVE is deployed in an independent commercial facility at the time of writing. HDTV is being used for the production of material to be released on theatrical (cinema) film. Its acceptance as a production medium for cinema awaits the wider availability of HDTV production facilities and more knowledge of HDTV production techniques on the part of the film production community.

HDTV Exchange Standards

The *de facto* international television program distribution standard has been for the last 40 years, and continues to be, 35mm motion picture film. In North America, film is transferred to video using 3-2pulldown, which involves scanning successive film frames alternately to form first three then two video fields. The film is run 0.1% slower than 24 frames per second, to result in the required 59.94Hz field rate. In Europe, film is run four percent fast with 2-2 pulldown to result in a 50Hz frame rate.

Discussions of exchange standards are in an early stage, but there is general agreement that film “friendliness” will be important for ATV: it is certain that the primary origination medium for consumer ATV in any form will initially be 35mm motion picture film, due to the vast amount of existing program material in that medium.

ATV Transmission Standards

All proposed transmission standards involve the reduction of transmission bandwidth by exploiting the statistical properties of “typical” images and the perceptual properties of the human visual system. Terrestrial and satellite broadcasting requires spectrum allocation, which is subject to domestic and international political concerns. Broadcasting standards are agreed by the International Radio Consultative Committee (CCIR), a treaty organization that is part of the United Nations. CCIR Recommendations — which might be called standards — are agreed unanimously and internationally. The CCIR started setting broadcasting standards well before the introduction of video recording, and the CCIR has inherited video production and exchange standards even though they do not strictly speaking involve the radio spectrum. The CCIR has adopted Recommendation709 for an HDTV production system. HDTV colorimetry has been agreed, but the recommendation is in a half-finished state reflecting the lack of international agreement on remaining parameters, particularly frame rate and raster structure.

Broadcasters in Europe and the U.S. have proposed transmission systems having on 50Hz and 59.94Hz field rates respectively, citing requirements for “compatibility” with local broadcast standards. No commercial equipment, and very little experimental equipment, exists for either of these standards.

ATV Transmission in Europe

The standardization process in Europe is substantially different from the standardization process in North America. Most broadcasting organizations are state-owned. Standards are agreed upon by the European Broadcasting Union, whose only members are broadcasters. These meetings are closed; manufacturers (and other interested parties) attend only when invited.

Systems based on 1250/50 scanning, with a raster structure of 1920×1152, have been proposed by the Eureka-95 project in Europe. These proposals are HDTV extensions to the MAC system (HD-MAC). British and Swedish researchers have introduced a revolutionary broadcasting technology called orthogonal frequency division multiplexing (OFDM) that may accelerate the consideration of digital broadcasting in Europe.

The Europeans (and the Australians) had a strong political interest in basing HDTV on MAC, due its recent deployment. Receiver manufacturers include MAC decoders in their new receivers, but consumers must install set-top converters in order for old receivers to receive MAC. The European broadcasting community would have found it embarrassing to require consumers to purchase new converters for another new standard— for digital ATV — just a few years after the heavily-publicized introduction of MAC. MAC is therefore currently being promoted in Europe as being capable of upgrade for HDTV (HD-MAC), but the recent business failures associated with D-MAC make extension of the service to HD-MAC problematic.

Commercial/Industrial/ScientificApplications

Current 525-line video systems have about 640 visible pixels per line and about 480 visible lines per frame for a total of about 0.3 megapixels. Current workstations have between 1 and 1.25 megapixels per frame (e.g.1152×900 or 1280×1024). HDTV has approximately two megapixels per frame, or roughly twice the pixel count of current workstations. This pixel count,

combined with a picture aspect ratio of 16:9, allows a display measuring 19 inches by 11 inches at 100 dots per inch. This is sufficient for two 8.5 by 11-inch (A4) pages sideby-side or an 11 by 17inch (B-size or A3) engineering drawing, with a few inches left over on the side for menus and icons. Many computer workstation users today obtain a two-megapixel display by attaching two screens to one workstation; this “two-headed configuration” is also typical of computer animation and medical systems. This interim solution to increased pixel count will be remedied by HDTV displays.

Computer users have for many years been plagued by a wide variety of incompatible monitor interface standards. For example, at one megapixel, the user doesn’t particularly care whether the display is 1152×900 (Sun), 1120×832 (NeXT), 1152×870 (Mac) or 1024×864 (DEC), but each manufacturer carries the burden of specifying its own unique monitors and monitor interface standards.

HDTV offers a common scanning and interface standard for the next generation of workstations. This will simplify the interfacing of workstation to monitors, projectors, downconverters, film recorders and other peripheral equipment. 1125/60 is available as an output standard for computer graphics equipment such as workstations from Silicon Graphics and Symbolics. Third-party HDTV adapters are available for Sun, Macintosh and other computers.

Use of the HDTV production standard by the computer industry will open access to equipment for image capture, recording, transmission, distribution and display. Interface to 525-line video equipment has been difficult due to the disparity in interface standards between computer graphics equipment and video equipment. Further, poor detail and color resolution in NTSC have precluded its use in application areas such as medicine and graphics arts. HDTV remedies those deficiencies by adopting component color coding

(instead of composite coding as in NTSC), zero setup for accurate reproduction of blacks (instead of 7.5percent setup as in NTSC and in EIA-343-A), a single well-characterized colorimetry standard (as opposed to the wide variety of phosphor chromaticity and white point values currently in use in computer graphics) and a well-defined transfer function that will allow accurate gamma correction.

In the past, many application areas have been forced to adopt proprietary display interface standards because the resolution or color accuracy available from standard workstation platforms has been inadequate. HDTV has a display quality that will meet the requirements of even the most exacting users and this will allow the use of platform technology in place of proprietary solutions in applications such as printing and publishing. Quantel's HDTV Graphic Paintbox is optimized for printing and publishing applications, and includes interfaces to prepress equipment. The Rebo Research ReStore offers access to HDTV through a Macintosh computer, and thereby allows the use of HDTV imagery and equipment with commercially-available Macintosh programs for retouching, presentation, color separation, and many other applications.

15.6 SUMMARY

- TeamWorkStation (TWS) provides a "shared screen" in addition to an individual screen.

- High definition television (HDTV) is defined as having twice the vertical and twice the horizontal resolution of conventional television, a picture aspect ratio of 16:9, a frame rate at least 24 Hz, and at least two channels of CD-quality sound.

- Advanced Television (ATV) refers to transmission systems designed for the delivery of entertainment to consumers, at quality levels substantially improved over conventional television.

- Despite the chaos in distribution standards, the Society of Motion Picture and Television Engineers (SMPTE) has adopted SMPTE Standard 240M for 1125/60 studio equipment
- The picture quality of HDTV is superior to that of 35mm motion picture film, but less than the quality of 35mm still film.

- The colorimetry obtainable with the color separation filters and CRT phosphors of a video system is greatly superior to that possible with the photochemical processes of a color film system.

- Standards for motion pictures and video exist in three tiers: production, exchange, and distribution.

- The standardization process in Europe is substantially different from the standardization process in North America.

- Current 525-line video systems have about 640 visible pixels per line and about 480 visible lines per frame for a total of about 0.3 megapixels

- Quantel's HDTV Graphic Paintbox is optimized for printing and publishing applications, and includes interfaces to prepress equipment.

- The Rebo Research ReStore offers access to HDTV through a Macintosh computer, and thereby allows the use of HDTV imagery and equipment with commercially-available Macintosh programs for retouching, presentation, color separation, and many other applications.

15.7Unit End Exercises

1. Explain the concept of TeamWorkStation.
2. Write a note on HDTV Quality.
3. Explain different HDTV Standards.
4. Write a note on HDTV Exchange Standards.

15.8Additional Reference

- Research Paper: *ClearFace: Translucent Multiuser Interface for TeamWorkStation* by Hiroshi Ishii and KazuhoArita available online at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.119.4184&rep=rep1&type=pdf
- Research Paper: *Toward an open shared workspace: Computer and Video Fusion approach or teamworkstation* by Hiroshi Ishii and Naomi Miyake available online athttp://www.audentia-gestion.fr/MIT/p37-ishii.pdf
- Article: High Definition Television and Desktop Computing by Charles A. Poynton available online athttps://poynton.ca/PDFs/HDTVDesktopComputing.pdf

16

Knowledge based Multimedia Systems

Unit Structure

16.1 Objectives
16.2 Introduction
16.3 The Anatomy of an Intelligent Multimedia System
16.4 Summary
16.5 Unit End Exercises
16.6 Additional Reference

16.1 OBJECTIVES

In this Chapter you will understand:

- Multimedia System Design
- Concept of Multi-modal language understanding
- Multi-modal language generation

16.2 INTRODUCTION

Multimedia communication is a part of everyday life and its appearance in computer applications is increasing in frequency and diversity. Intelligent or knowledge-based computer supported communication promises a number of benefits including increased interaction efficiency and effectiveness.

This Chapter defines the area of intelligent multimedia communication, outlines fundamental research questions, summarizes the associated scientific and technical history, identifies current challenges and concludes by predicting future breakthroughs including multi-lingually.

16.3 THE ANATOMY OF AN INTELLIGENT MULTIMEDIA SYSTEM

Intelligent Multimedia can be defined as the interaction between human-human, human-system, and human-information. This includes interfaces to people, interfaces to applications, and interfaces to information.

A multimedia computer system is a computer system that can create, import, integrate, store, retrieve, edit, and delete two or more types of media materials in digital form, such as audio, image, full-motion video, and text information. Multimedia computer systems also may have the ability to analyze media materials (e.g., counting the number of occurrences of a word in a text file). A multimedia computer system can be a single- or multiple-user system. Networked multimedia computer systems can transmit and receive digital multimedia materials over a single computer network or over any number of interconnected computer networks. As multimedia computer systems evolve, they may become intelligent systems by utilizing expert system technology to assist users in selecting, retrieving, and manipulating multimedia information.

- Multimedia: physical means via which information is input, output and/or stored (e.g., interactive devices such as keyboard, mouse, displays; storage devices such as disk or CD-ROM).

- Multimodal: human perceptual processes such as vision, audition, taction.

- Multicodal: representations used to encode atomic, elements, syntax, semantics, pragmatics and related data structures (e.g., lexicons, grammars) associated with media and modalities.

The majority of computational efforts have focused on multimedia human computer interfaces. There exists a large literature and associated techniques to develop learnable, usable, transparent interfaces in general.

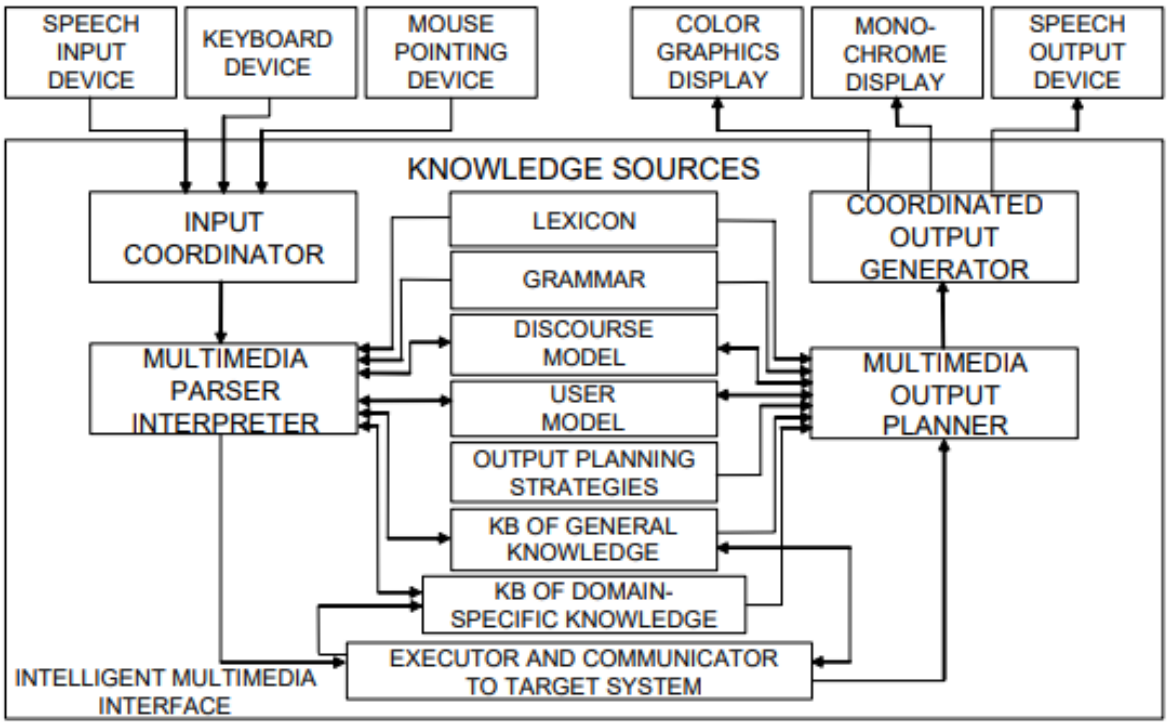


Figure 16.1: Multimedia System Design

Significant progress has been made in multimedia interfaces, integrating language, speech, and gesture. For example, Figure 1 shows the CUBRICON system architecture. CUBRICON enables a user to interact using spoken or typed natural language and gesture, displaying results using combinations of language, maps, and graphics. Interaction management is affected via models of the user and the ongoing discourse, which not only influence the generated responses but also manage window layout, based on user focus of attention.

Role of Multiple Languages in multimedia communication interaction

Amultimodal interaction resides in the integration of multiple subfields. When extending techniques and methods to multiple languages, we have the benefit of drawing upon previous monolingual techniques. For example, language

generation techniques and components, built initially for monolingual generation, can often be reused across languages. Analogously, interaction management components can be reused.

Of course, many language specific phenomena remain to be addressed. For example, in generation of multilingual and multimedia presentations, lexical length affects the layout of material both in space and in time. For instance, in laying out a multilingual electronic yellow page, space may be strictly limited given a standard format and so variability in linguistic realization across languages may pose challenges. In a multimedia context, one might need to not only generate language specific expressions, but also culturally appropriate media.

Making further progress in this area, researchers may take advantage of some unique resources to help develop systems perform multimedia information access, including dubbed movies, multilingual broadcast news, that might help accelerate the development of, for example, multilingual video corpora.

Multi-Modal Language Understanding

People commonly and naturally use coordinated simultaneous natural language and graphic gestures when working at graphic displays. These modes of communication combine synergistically to form an efficient language for expressing definite references and locative adverbials.

One of the benefits of this multi-modal language is that it eliminates the need for the lengthy definite descriptions that would be necessary for unnamed objects if only natural language were used. Instead, a terse reference such as "this SAM" (surface-to-air missile system) accompanied by a point to an entity on the display can be used. CUBRICON accepts such NL accompanied by simultaneous coordinated pointing gestures. The NL can be input via the

keyboard, the speech recognition system, or a mixture of both. CUBRICON provides:

- variety in the object types that can be targets of point gestures; these object types include windows, form slots, table entries, icons, and geometric points;
- variety in the number of point gestures allowed per phrase; each noun phrase can be accompanied by zero or more-point gestures; such a phrase may contain no words, just the pointing gestures;
- variety in the number of multi-modal phrases allowed per sentence; deictic gestures can accompany more than one phrase per sentence.

Just as natural language used alone has shortcomings, so also does the use of pointing gestures alone. Pointing used alone has the following problems:

(1) a point gesture can be ambiguous if the point touches the area where two or more graphical figures or icons overlap or

(2) the user may inadvertently miss the object at which he intended to point.

To handle these pointing problems, some systems use default techniques such as having a point handler return the entity represented by

a) the "top" or "foremost" icon where the system has a data structure it uses to remember the order in which icons are "painted" on the display (i.e., which are further in the background and which are foremost in the foreground) or

b) the icon whose "center" is closest to the location on the screen/window touched by the point.

A serious disadvantage of such default point-interpretation techniques is that it is difficult, if not impossible, for certain icons to be selected via a point reference. CUBRICON's acceptance of dual-media input (NL accompanied by coordinated pointing gestures) overcomes the limitations of the above weak

default techniques and provides an efficient expressive referencing capability. The CUBRICON methodology for handling dualmedia input is a decision-making process that depends on a variety of factors such as the types of candidate objects being referenced, their properties, the sentential context, and the constraints on the participants or fillers of the semantic case frame for the verb of any given sentence.

Multi-Modal Language Generation

Just as CUBRICON accepts NL accompanied by deictic and graphic gestures during input, CUBRICON can generate multi-modal language output that combines NL with deictic gestures and graphic expressions. An important feature of the CUBRICON design is that NLand graphics are incorporated in a single language generator providing a unified multi-modal language with speech and graphics synchronized in real time.

Another important aspect of the CUBRICON system is that it distinguishes between spoken and written (to a CRT display) NL. CUBRICON uses graphic and deictic gestures with spoken NL only (not with written NL), since a pointing or graphic gesture needs to be temporally synchronized with the corresponding verbal phrase, allowing for multiple graphic gestures within any individual sentence. The coordination between a graphic gesture and its co-referring verbal phrase is lost if printed text is used instead of speech.

A pointing gesture can be used very effectively with a terse NL phrase (e.g., "this SAM") to reference an object that is visible on one of the displays (by the system as well as the user). When CUBRICON generates written NL, however, deictic/graphic expressions are not used, but, instead, definite descriptions are generated as noun phrases with sufficient specificity to hopefully avoid ambiguous references.

Multimedia Message Systems

Multimedia message systems are extensions of contemporary electronic mail and conference systems which include multimedia data handling capabilities. Multimedia message systems can create, transmit, receive, reply to, forward, save, retrieve, and delete multimedia messages. As part of the message creation and editing processes, multimedia message systems can import different media materials and integrate them.

Since multimedia message systems can incorporate sophisticated data handling capabilities, the distinction between this type of system and multimedia database systems can sometimes appear hazy; however, the primary purpose of these two kinds of systems is quite different. Multimedia database systems are optimized for database functions, while multimedia message systems are optimized for communication functions.

The CUBRICON Intelligent Window Manager

The CUBRICON Intelligent Window Manager (CIWM) is a knowledge-based system that automates windowing operations. The CIWM is a component of CUBRICON, a prototype knowledge-based multi-media human-computer interface. CUBRICON accepts inputs and generates outputs using integrated multiple media/modalities including speech, printed/typed natural language, tables, forms, maps, graphics, and pointing gestures. The CIWM automatically performs window management functions on CUBRICON's color and monochrome screens. These functions include window creation, sizing, placement, removal, and organization. These operations are accomplished by the CIWM without direct human inputs, although the system provides for user override of the CIWM decisions.

The Intelligent Window Manager automatically performs all window placement and manipulation functions within the CUBRICON system. The decision to

automate window management functions was based on the premise that this would reduce the user efforts required for window management, and thus free user mental and temporal resources for task domain activities. The goal was to automatically perform window management functions well enough so that the user would not need to manipulate the windows directly. The window management functions performed by the CUBRICON window manager include window: creation; placement; sizing; moving; and removal. The fact that the time spent manipulating windows m a windowing system consumes a significant portion of overall problem-solving time has been demonstrated experimentally [Davies85; B1y86], at least for certain types of tasks. Davies et al. found that for tasks requiring supplemental information relative to a primary task, the windowing environment allowed more error-free performance but took significantly longer. Their study indicates that the additional time spent, was due to window management operations (E.g., displaying and positioning windows, scrolling to desired locations within windows). Their data also indicates that the reduction in errors was not simply the result of having spent more time on the task. The time differential was evident even when all errors had to be corrected. Apparently, the overhead of window management adds a significant time burden.

Tables are used to display voluminous homogeneous information. The functions performed on table windows are creation, placement, sizing and removal. Tables can be temporary or permanent. Generally, if the information to be presented is on an existing table, the appropriate table entries are highlighted. However, if numerous table entries (more than four) are highlighted at various positions in the table, the user may find it difficult to compare information particularly if the information is not visible on one screen. Therefore, a temporary table is created which contains the information requested by the user, making it easy to view contiguously only the requested information. This table is referred to as temporary since it is visible for one user interaction. Permanent windows remain on the display until they are

removed, due to space constraints. Permanent table windows can be placed either on the monochrome display or the color graphics display whereas temporary table windows are placed only on the monochrome display. In addition, permanent tables can be related to another display, such as a map. A table which is related to a map identifies the important attributes of the objects contained in the map.

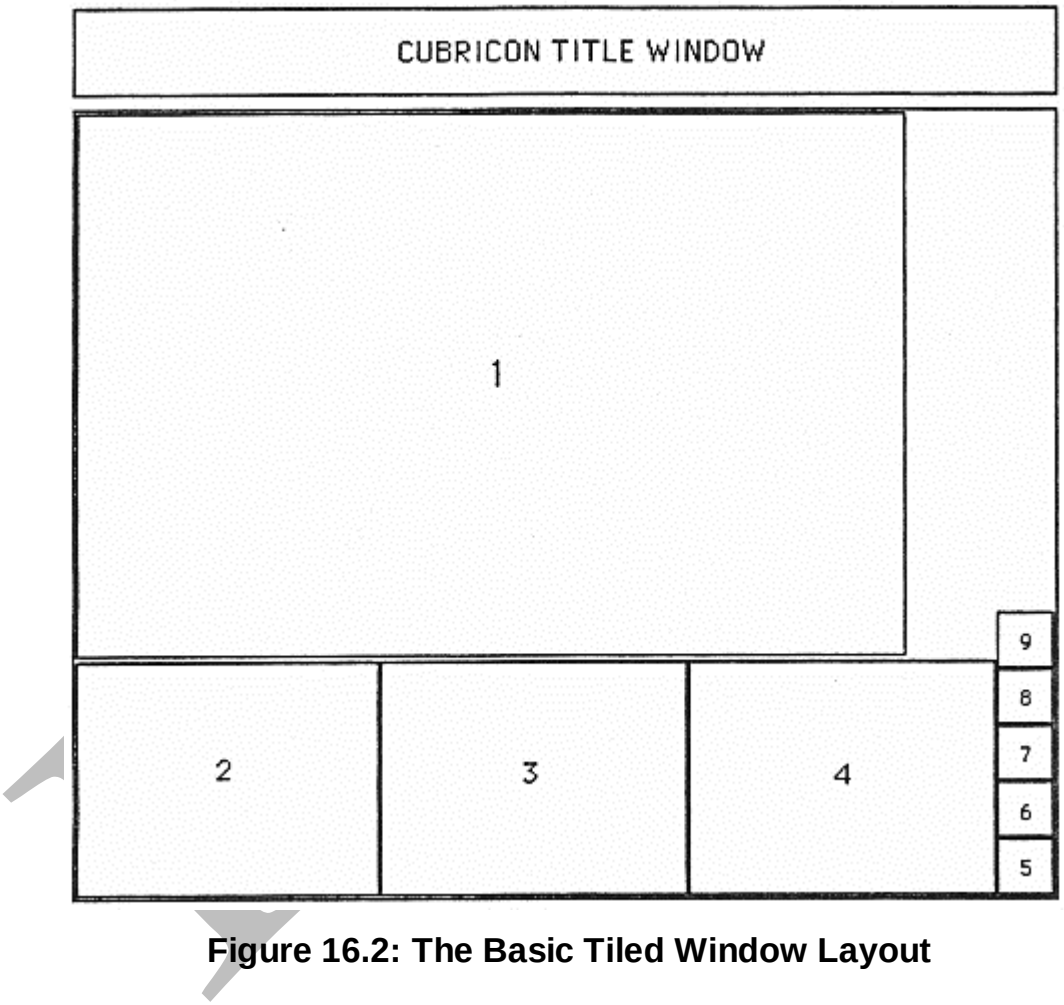


Figure 16.2: The Basic Tiled Window Layout

The strength of an overlapping window configuration is the ability of windows to conform to their contents, maximizing the visibility of these contents. CUBRICON uses a tiled windowing approach as a default, but allows the "tiled" windows to overlap adjacent windows when necessary based on window contents. This allows CUBRICON to realize the advantages of both types of windowing systems.

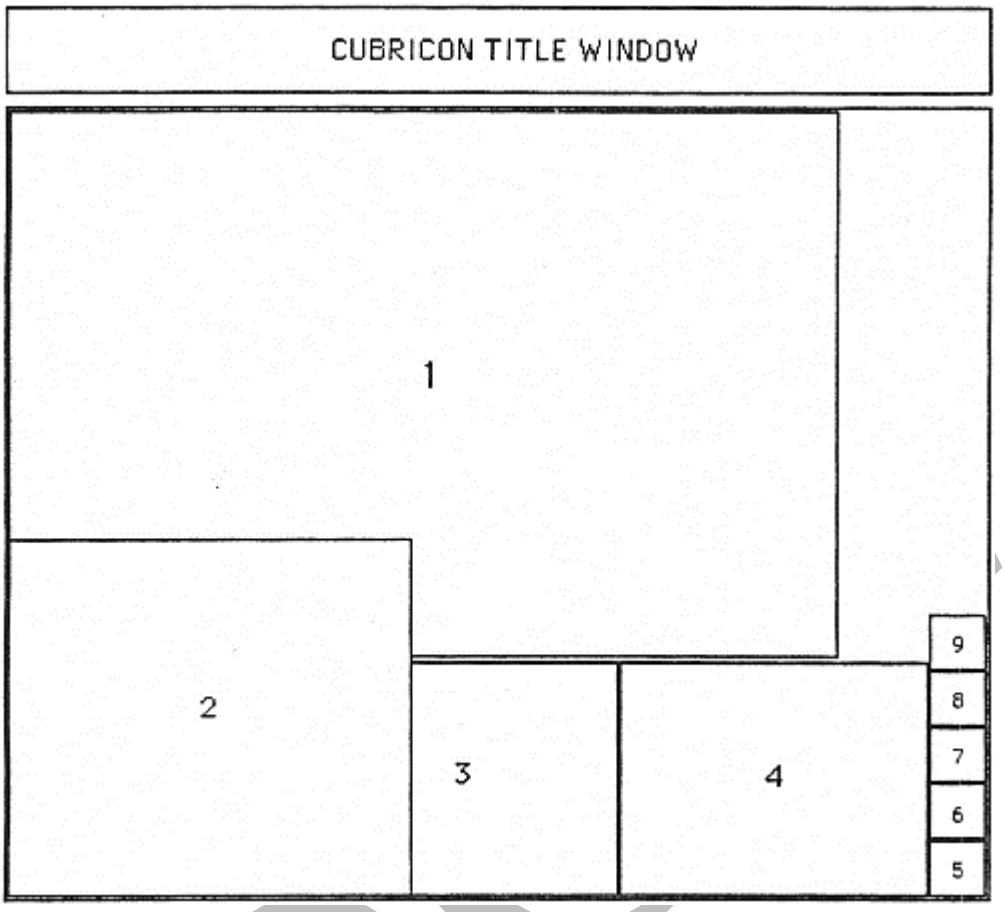


Figure 16.3 The Overlapping Window Layout.
The window in position 2 is overlapping the windows in position 1 and 3 because its contents could not fit otherwise

Examples of Multimedia Message Systems

BBN Software Products Corp. is marketing BBN/Slate, a multimedia electronic mail and conferencing system.The system runs on SUN workstations. BBN/Slate messages can include five types of information: bit-map images, geometric graphics, speech annotations, spreadsheets with associated charts, and text. An integrated media editor, which has specialized editing functions for each type of information, permits the user to easily edit messages.

The system allows users to create, store, retrieve, send, receive, sort, reply to, forward, and delete messages. If the user sends a message to a colleague whose system does not have multimedia capability, BBN/Slate will

automatically send the message in text-only form. Utilizing the conferencing capability of BBN/Slate, geographically dispersed users can jointly edit documents. The system can be tailored to meet the user's needs with a built-in programming language called the Slate Extension Language.

16.4 SUMMARY

- Intelligent Multimedia can be defined as the interaction between human-human, human-system, and human-information
- CUBRICON enables a user to interact using spoken or typed natural language and gesture, displaying results using combinations of language, maps, and graphics
- Amultimodal interaction resides in the integration of multiple subfields
- One of the benefits of this multi-modal language is that it eliminates the need for the lengthy definite descriptions that would be necessary for unnamed objects if only natural language were used.
- Multimedia message systems are extensions of contemporary electronic mail and conference systems which include multimedia data handling capabilities.
- BBN/Slate messages can include five types of information: bit-map images, geometric graphics, speech annotations, spreadsheets with associated charts, and text
- The CUBRICON Intelligent Window Manager (CIWM) is a knowledge-based system that automates windowing operations.

16.5Unit End Exercises

1. Write a note onMultimedia System Design.
2. Explain the concept of Multi-modal language.
3. What is Multi-modal language generation?
4. Explain in brief Cubricon Intelligent Window Manager.
5. Give some examples of Multimedia Message Systems.

16.6Additional Reference

- *Natural Language with Integrated Deictic and Graphic Gestures* by J.G. Neal, C.Y. Thielman, Z. Dobes, S.M. Haller, S.C. Shapiro available online athttps://www.aclweb.org/anthology/H89-2054

- *Report for the Intelligent Multi-Media Interfaces Project*available online athttps://cse.buffalo.edu/~shapiro/Papers/NealEtAlDraft.pdf