Chapter one
Introduction to Multimedia Systems

What is Multimedia?

Before we go on, it is important to define multimedia. Let us define it from two perspectives:

1) In terms of what multimedia is all about.
It refers to the storage, transmission, interchange, presentation and perception of different information types (data types) such as text, graphics, voice, audio and video where:
Storage- refers to the type of physical means to store data.
-Magnetic tape
-Hard disk
-Optical disk
-DVDs
-CD-ROMs, etc.
Presentation- refers to the type of physical means to reproduce information to the user.
-Speakers
-Video windows, etc.
Representation- related to how information is described in an abstract form for use within an electronic system. E.g. to present text to the user, the text can be coded in raster graphics, primitive graphics, or simple ASCII characters. The same presentation, different representation.
Perception- describes the nature of information as perceived by the user
-Speech
-Music
-Film

2) Based on the word "Multimedia"
It is composed of two words:
Multi- multiple/many
Media- source
Source refers to different kind of information that we use in multimedia.
This includes:
-text
-graphics
-audio
-video
-images
Multimedia refers to multiple sources of information. It is a system which integrates all the above types.

Definitions:
Multimedia means computer information can be represented in audio, video and animated format in addition to traditional format. The traditional formats are text and graphics.

General and working definition:
Multimedia is the field concerned with the computer controlled integration of text, graphics, drawings, still and moving images (video), animation, and any other media where every type of information can be represented, stored, transmitted, and processed digitally.

What is Multimedia Application?
A Multimedia Application is an application which uses a col-lection of multiple media sources e.g. text, graphics, images, sound/audio, animation and/or video.

What is Multimedia system?
A Multimedia System is a system capable of processing multimedia data. A Multimedia System is characterized by the processing, storage, generation, manipulation and rendition of multimedia information.

Characteristics of a Multimedia System
A Multimedia system has four basic characteristics:
- ✓ Multimedia systems must be computer controlled
- ✓ Multimedia systems are integrated
- ✓ The information they handle must be represented digitally
- ✓ The interface to the final presentation of media is usually interactive

Multimedia Applications (where it is applied)
- ✓ Digital video editing and production systems
- ✓ Home shopping
- ✓ Interactive movies, and TV
- ✓ Multimedia courseware
- ✓ Video conferencing
- ✓ Virtual reality(the creation of artificial environment that you can explore, e.g. 3-D images, etc)
- ✓ Distributed lectures for higher education
- ✓ Tele-medicine
- ✓ Digital libraries
- ✓ World Wide Web
- ✓ On-line reference works e.g. encyclopedias, games, etc.
- ✓ Electronic Newspapers/Magazines
- ✓ Games
- ✓ Groupware (enabling groups of people to collaborate on projects and share information)

World Wide Web (WWW) and Multimedia

Multimedia is closely tied to the World Wide Web (WWW). Without networks, multimedia is limited to simply displaying images, videos, and sounds on your local machine. The true power of multimedia is the ability to deliver this rich content to a large audience.

Features of Multimedia

Multimedia has three aspects:

Content: movie, production, etc.

Creative Design: creativity is important in designing the presentation

Enabling Technologies: Network and software tools that allow creative designs to be presented.

History of Multimedia Systems

Newspaper was perhaps the first mass communication medium, which used mostly text, graphics, and images.

In 1895, Gugliemo Marconi sent his first wireless radio transmission at Pontecchio, Italy. A few years later (in 1901), he detected radio waves beamed across the Atlantic. Initially invented for telegraph, radio is now a major medium for audio broadcasting.
Television was the new media for the 20th century. It brings the video and has since changed the world of mass communications.

On computers, the following are some of the important events:

1945 -Vannevar Bush (1890-1974) wrote about Memex.
MEMEX stands for MEMory Extension and it amounts to hypermedia system. A Memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.

1960s-Ted Nelson started Xanadu project (Xanadu – a kind of deep Hypertext**).**
Project Xanadu was the explicit inspiration for the World Wide Web, for Lotus Notes and for HyperCard, as well as less-well-known systems.

1967 - Nicholas Negroponte formed the Architecture Machine Group at MIT. A combination lab and think tank responsible for many radically new approaches to the human-computer interface. Nicholas Negroponte is the Wiesner Professor of Media Technology at the Massachusetts Institute of Technology.

1968 - Douglas Engelbart demonstrated NLS (Online Systems) system at SRI.
Shared-screen collaboration involving two persons at different sites communicating over a network with audio and video interface is one of the many innovations presented at the demonstration.

1969 - Nelson & Van Dam hypertext editor at Brown University

1976 - Architecture Machine Group proposal to DARPA: Multiple Media

1985 - Negroponte, and Wiesner opened MIT Media Lab, a leading research institution

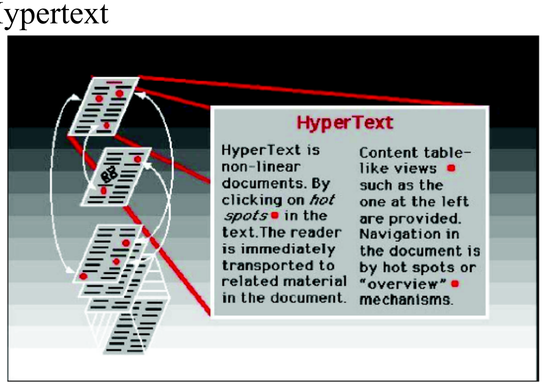
investigating digital video and multimedia
1989 - Tim Berners-Lee proposed the World Wide Web to CERN (European Council for Nuclear Research)
1990 - K. Hooper Woolsey, Apple Multimedia Lab gave education to 100 people
1991 – MPEG-1 was approved as an international standard for digital video. Its further development led newer standards MPEG-2 and MPEG-4.
1992- JPEG was accepted as the international standard for digital image compression.
1992 - The first M-Bone audio multicast on the net (MBONE- Multicast Backbone)
1993 – U. Illinois National Center for Supercomputing Applications introduced NCSA Mosaic (a web browser)
1994 - Jim Clark and Marc Andersen introduced Netscape Navigator (web browser)
1995 - Java for platform-independent application development.
1996 – DVD video was introduced; high quality, full-length movies were distributed on a single disk. The DVD format promised to transform the music, gaming and computer industries.
1998 – hand-held mp3 devices first made into the consumer market
2000 – WWW size was estimated at over 1 billion pages

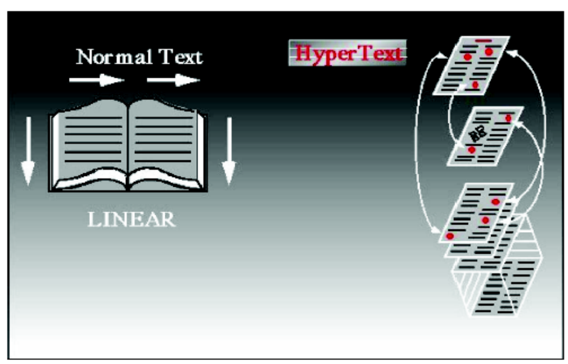
Hypermedia/Multimedia

What is Hypertext and Hypermedia?
Hypertext is a text, which contains links to other texts. The term was invented by Ted Nelson around 1965. Hypertext is usually non-linear (as indicated below).

Hypermedia is not constrained to be text-based. It can include other media, e.g., graphics, images, and especially the continuous media -- sound and video. Apparently, Ted Nelson was also the first to use this term.
The World Wide Web (www) is the best example of hypermedia applications.

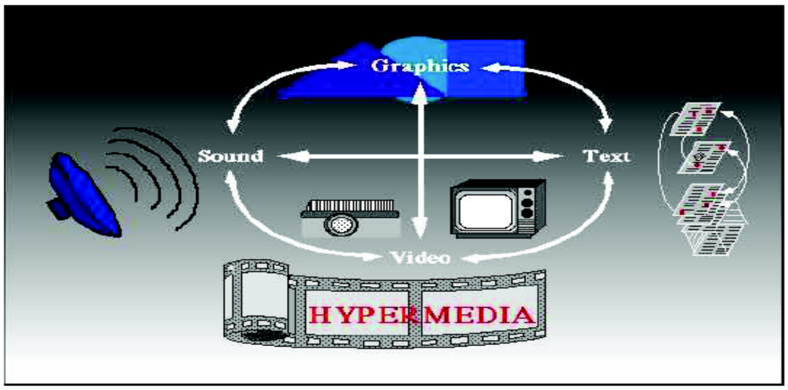
Hypertext





Hypertext is therefore usually non-linear (as indicated above).

Hypermedia



Hypermedia is the application of hypertext principles to a wider variety of media, including audio, animations, video, and images.

Examples of Hypermedia Applications:

- ✓ The World Wide Web (WWW) is the best example of hyper-media applications.
- ✓ PowerPoint
- ✓ Adobe Acrobat
- ✓ Macromedia Director

Desirable Features for a Multimedia System

Given the above challenges, the following features are desirable for a Multimedia System:

1. *Very high processing speed processing power*. Why? Because there are large data to be processed. Multimedia systems deals with large data and to process data in real time, the hardware should have high processing capacity.
2. It should support *different file formats*. Why? Because we deal with different data types (media types).
3. *Efficient and High Input-output*: input and output to the file subsystem needs to be efficient and fast. It has to allow for real-time recording as well as playback of data. *e.g.* Direct to Disk recording systems.
4. *Special Operating System*: to allow access to file system and process data efficiently and quickly. It has to support direct transfers to disk, real-time scheduling, fast interrupt process-ing, I/O streaming, *etc.*
5. *Storage and Memory*: large storage units and large memory are required. Large Caches are also required.
6. *Network Support*: Client-server systems common as distributed systems common.
7. *Software Tools*: User-friendly tools needed to handle media, design and develop

applications, deliver media.

Challenges of Multimedia Systems

1) *Synchronization issue*: in MM application, variety of media are used at the same instance. In addition, there should be some relationship between the media. E.g between Movie (video) and sound. There arises the issue of synchronization.
2) *Data conversion*: in MM application, data is represented digitally. Because of this, we have to convert analog data into digital data.
3) *Compression and decompression*: Why? Because multimedia deals with large amount of data (e.g. Movie, sound, etc) which takes a lot of storage space.
4) Render different data at same time — continuous data.

Multimedia System Requirement

1) Software tools
2) Hardware Requirement

Software Requirement

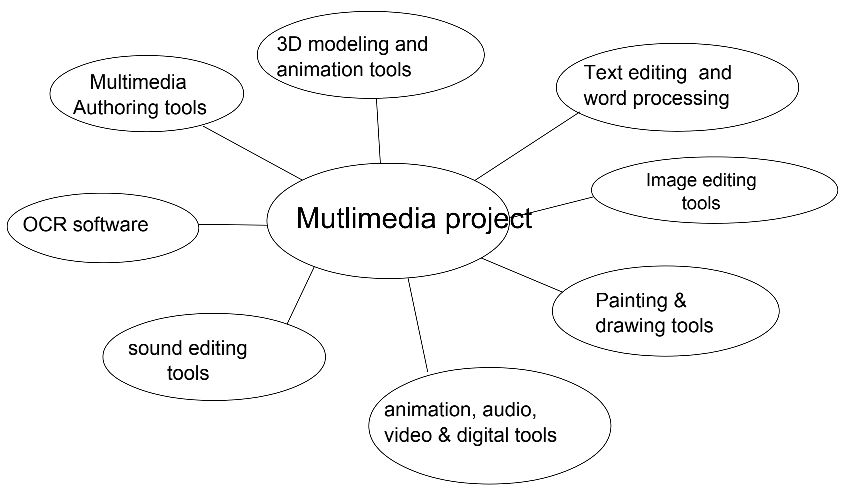


Figure multimedia software tools

3-D and Animation Tools:

These software provide 3D clip art object such as people, furniture, building, car, airplane, tree, etc. You can use these objects in your project easily.

A good 3D modeling tool should include the following features:

- ✓ Ability to drag and drop primitive shape into screen
- ✓ Ability to create objects from scratch
- ✓ Ability to add realistic effects such as transparency, shadowing, fog, etc.
- ✓ Multiple window that allow user to view model in each dimension
- ✓ Color and texture mapping

Examples:

-3Ds Max
-Maya
-Logomotion
-SoftImage

Text editing and word processing tools:

Word processors are used for writing letters, invoices, project content, etc. They include features like:

- ✓ spell check
- ✓ table formatting
- ✓ thesaurus
- ✓ templates (e.g. letters, resumes, & other common documents)

Examples:

-Microsoft Word,
-Word perfect,
-Open Office Word
-Note pad

In word processors, we can actually embed multimedia elements such as sound, image, and video.

Sound Editing Tools

They are used to edit sound (music, speech, etc).The user can see the representation of sound in fine increment, score or wave form. User can cut, copy, and paste any portion of the sound to edit it. You can also add other effects such as distort, echo, pitch, etc.

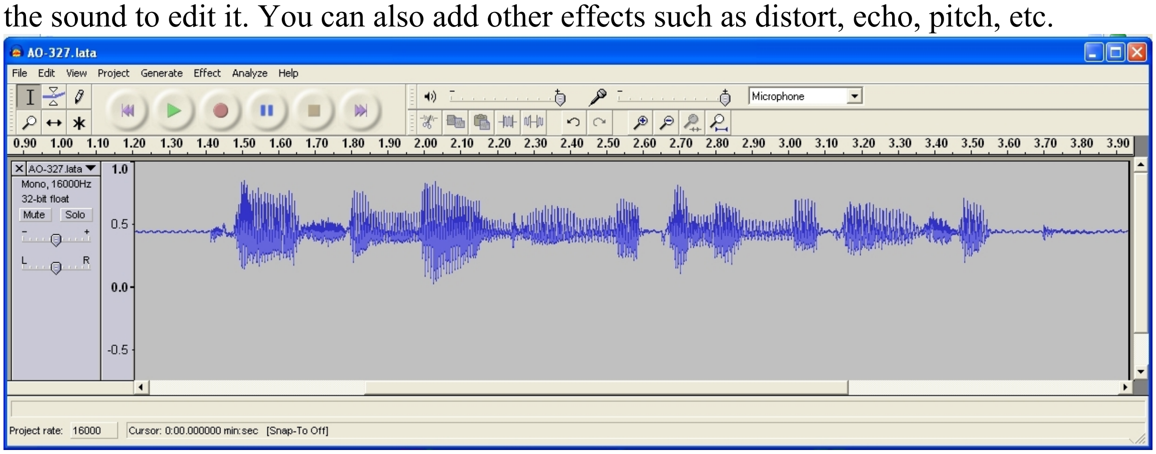


Fig sound displayed in wave form by Audacity

Examples:

-sound forge
-Audacity
-Cool Edit

Multimedia Authoring Tools
Multimedia authoring tools provide important framework that is needed for organizing and editing objects included in the multimedia project (e.g. graphics, animation, sound, video, etc). They provide editing capability to limited extent. (More on this in chapter 2).

Examples:

-Macromedia Flash
-Macromedia Director
-Macromedia Authoware

OCR Software
These software convert printed document into electronically recognizable ASCII character. It is used with scanners. Scanners convert printed document into bitmap. Then these software break the bitmap into pieces according to whether it contains text or graphics. This is done by examining the texture and density of the bitmap and by detecting edges.

Text area → ASCII text
Bitmap area→ bitmap image

To do the above, these software use probability and expert system.

Use:

- ✓ To include printed documents in our project without typing from keyboard
- ✓ To include documents in their original format e.g. signatures, drawings, etc

Examples:

-OmniPage Pro
-Perceive
-IRIS

Painting and Drawing Tools

To create graphics for web and other purposes, painting and editing tools are crucial.

Painting Tools: are also called image-editing tools. They are used to edit images of different format. They help us to retouch and enhance bitmap images. Some painting tools allow to edit vector based graphics too.
Some of the activities of editing include:

- ✓ blurring the picture
- ✓ removing part of the picture
- ✓ add texts to picture
- ✓ merge two or more pictures together, etc

Examples:

-Macromedia Fireworks
-Adobe Photoshop

Drawing Tool: used to create vector based graphics.
Examples:

- -Macromedia Freehand
- -CorelDraw
- -Adobe Illustrator

Drawing and painting tools should have the following features:

- Scalable dimension for restore, stretch, and distorting images/graphics
- Customizable pen and brush shapes and sizes
- Multiple undo capabilities
- Capacity to import and export files in different formats
- Ability to create geometric shapes from circle, rectangle, line, etc.
- zooming for magnified image editing
- Support for third party plug-ins.

Video Editing

Animation and digital video movie are sequence of bitmapped graphic frames rapidly played back. Some of the tools to edit video include:

- -Adobe premier
- -Adobe After Effects
- -Deskshare Video Edit Magic
- -Videoshop

These applications display time references (relationship between time & the video), frame counts, audio, transparency level, etc.

Hardware Requirement
Three groups of hardware for multimedia:
1) Memory and storage devices
2) Input and output devices
3) Network devices

1) Memory and Storage Devices

Multimedia products require high storage capacity than text-based data. Huge drives are essential for the enormous files used in multimedia and audiovisual creation.
*I) **RAM**:* is the primary requirement for multimedia system. Why?
Reasons:

- you have to store authoring software itself. E.g. Flash takes 20MB of memory, Photoshop 16-20MB, etc.
- digitized audio and video is stored in memory
- Animated files, etc.

To store this at the same time, you need large amount of memory

II) ***Storage Devices***: large capacity storage devices are necessary to store multimedia data.
Floppy Disk: not sufficient to store multimedia data. Because of this, they are not used to store multimedia data.
Hard Disk: the capacity of hard disk should be high to store large data.
CD: is important for multimedia because they are used to deliver multimedia data to users. A wide variety of data like:

- ✓ Music(sound, & video)
- ✓ Multimedia Games
- ✓ Educational materials
- ✓ Tutorials that include multimedia
- ✓ Utility graphics, etc

DVD: have high capacity than CDs. Similarly, they are also used to distribute multimedia data to users. Some of the characteristics of DVD:

- ✓ High storage capacity →4.7-17GB
- ✓ Use narrow tracks than CDs→ high storage capacity
- ✓ High data transfer rate

2) Input-Output Devices
I) ***Interacting with the system:*** to interact with multimedia system, we use either keyboard, mouse, track ball, or touch screen, etc.

Mouse: multimedia project is typically designed to be used with mouse as an input pointing device. Other devices like track ball and touch screen could be used in place of mouse. Track ball is similar with mouse in many ways.
Wireless mouse: important when the presenter has to move around during presentation

Touch Screen: we use fingers instead of mouse to interact with touch screen computers. There are three technologies used in touch screens:
i) Infrared light: such touch screens use invisible infrared light that are projected across the surface of screen. A finger touching the screen interrupts the beams generating electronic signal. Then it identifies the x-y coordinate of the screen where the touch occurred and sends signals to the operating system for processing.
ii) Texture-coated: such monitors are coated with texture material that is sensitive towards pressure. When user presses the monitor, the texture material on the monitor extracts the x-y coordinate of the location and send signals to operating system

iii) Touch mate:

Use: touch screens are used to display/provide information in public areas such as
-air ports
-Museums
-transport service areas
-hotels, etc

Advantage:

-user friendly
-easy to use even for non technical people
-easy to learn how to use

II) Information Entry Devices: the purpose of these devices is to enter information to be included in our multimedia project into our computer.

Graphical Tablets/ Digitizer: both are used to convert points, lines, and curves from sketch into digital format. They use a movable device called stylus.

Scanners: they enable us to use OCR software convert printed document into ASCII file. It also enables us to convert printed images into digital format.
Two types of scanners:

-flat bed scanners
-portable scanners

Microphones: they are important because they enable us to record speech, music, etc. The microphone is designed to pick up and amplify incoming acoustic waves or harmonics precisely and correctly and convert them to electrical signals. You have to purchase a superior, high-quality microphone because your recordings will depend on its quality.

Digital Camera and Video Camera (VCR): are important to record and include image and video in MMS respectively. Digital video cameras store images as digital data, and they do not record on film. You can edit the video taken using video camera and VCR using video editing tools.
Remark: video takes large memory space.

II) Output Devices

Depending on the content of the project, & how the information is presented, you need different output devices. Some of the output hardware are:
Speaker: if your project includes speeches that are meant to convey message to audience, or background music, using speaker is obligatory.
Projector: when to use projector:

-if you are presenting on meeting or group discussion,
-if you are presenting to large number of audience

Types of projector:

-LCD projector
-CRT projector

Plotter/printer: when the situation arises to present using papers, you use printers and/or

plotters. In such cases, print quality of the device should be taken into consideration.
Impact printers: not good quality graphics/poor quality→not used
Non-impact printers: good quality graphics

3) Network Devices

Why do we require network devices?
The following network devices are required for multimedia presentation:

i) *Modem:* which stands for **mo**dulator **dem**odulator, is used to convert digital signal into analog signal for communication of the data over telephone line which can carry only analog signal. At the receiving end, it does the reverse action i.e. converts analog to digital data.

Currently, the standard modem is called **v.90** which has the speed of 56kbps (kilo bits per second). Older standards include **v.34** which has the speed of 28kbps.
Types:

- ✓ External
- ✓ Internal

Data is transferred through modem in compressed format to save time and cost.

ii) ISDN: stands for Integrated Services Digital Network. It is circuit switched telephone network system, designed to allow digital transmission of voice and data over ordinary telephone copper wires. This has the advantage of better quality and higher speeds than available with analog systems.

- ✓ It has higher transmission speed i.e faster data transfer rate.
- ✓ They use additional hardware hence they are more expensive.

iii) Cable modem: uses existing cables stretched for television broadcast reception. The data transfer rate of such devices is very fast i.e. they provide high bandwidth. They are primarily used to deliver broadband internet access, taking advantage of unused bandwidth on a cable television network.

iv) *DSL:* provide digital data transmission over the telephone wires of local telephone network. The speed of DSL is faster than using telephone line with modem. How? They carry a digital signal over the unused frequency spectrum (analog voice transmission uses limited range of spectrum) available on the twisted pair cables running between the telephone company's central office and the customer premises.

Summary

Multimedia Information Flow

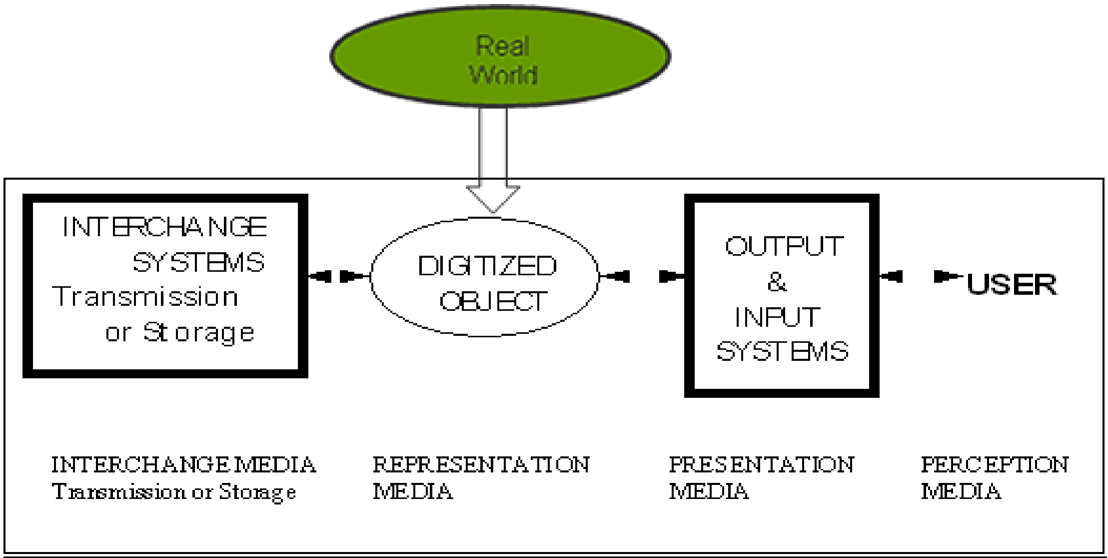


Fig Generalized multimedia information flow

Chapter 2

Multimedia Authoring Tools

What is Authoring System?

Authoring is the process of creating multimedia applications.
An authoring system is a program which has pre-programmed elements for the development of interactive multimedia presentations.

Authoring tools provide an integrated environment for binding together the different elements of a Multimedia production.

Multimedia Authoring Tools provide tools for making a complete multimedia presentation where users usually have a lot of interactive controls.

Multimedia presentations can be created using:

- ✓ simple presentation packages such as PowerPoint
- ✓ powerful RAD tools such as Delphi, .Net, JBuilder;
- ✓ true Authoring environments, which lie somewhere in between in terms of technical complexity.

Authoring systems vary widely in:

- ✓ Orientation
- ✓ Capabilities, and
- ✓ Learning curve: how easy it is to learn how to use the application

Why should you use an authoring system?

- ✓ Can speed up programming i.e. content development and delivery
- ✓ Time gains i.e. accelerated prototyping
- ✓ The content creation (graphics, text, video, audio, animation) is not affected by choice of authoring system

Authoring Vs Programming

There is big distinction between authoring and programming

Authoring	*Programming*
Assembly of multimedia	Involves low level assembly of multimedia
High level graphical interface design	Construction and control of multimedia
Some high level scripting e.g. lingo, ActionScript	Involves real languages like C and Java

Table 1 Authoring vs. Programming

Characteristics of Authoring Tools

A good authoring tool should be able to:

- ✓ integrate text, graphics, video, and audio to create a single multimedia presentation
- ✓ control interactivity by the use of menus, buttons, hotspots, hot objects etc.
- ✓ publish as a presentation or a self-running executable; on CD/DVD, Intranet, WWW
- ✓ Be extended through the use of pre-built or externally supplied components, plug-ins etc
- ✓ let you create highly efficient, integrated workflow
- ✓ Have a large user base.

Multimedia Authoring Paradigms

The *authoring paradigm*, or *authoring metaphor*, is the methodology by which the authoring system accomplishes its task.

There are various paradigms:

- ✓ Scripting Language
- ✓ Icon-Based Control Authoring Tool
- ✓ Card and Page Based Authoring Tool
- ✓ Time Based Authoring Tool
- ✓ Tagging Tools

Scripting Language

- ✓ Closest in form to traditional programming. The paradigm is that of a programming language, which specifies:
  - multimedia elements,
  - sequencing of media elements,
  - hotspots (e.g links to other pages),
  - synchronization, etc.

- ✓ Usually use a powerful, object-oriented scripting language
- ✓ Multimedia elements and events become objects that live in a hierarchical order
- ✓ In-program editing of elements (still graphics, video, audio, etc.) tends to be minimal or non-existent.
- ✓ Most authoring tools provide visually programmable interface in addition to scripting language.
- ✓ media handling can vary widely

Examples

- The Apple's HyperTalk for HyperCard,
- Asymetrix's OpenScript for ToolBook and
- Lingo scripting language for Macromedia Director
- ActionScript for Macromedia Flash

Here is an example lingo script to jump to a frame

```
global gNavSprite
on exitFrame
   go the frame
   play sprite gNavSprite
end
```

Iconic/Flow Control Tools

In these authoring systems, multimedia elements and interaction cues (or events) are organised as objects in a structural framework.

- ✓ Provides visual programming approach to organizing and presenting multimedia
- ✓ The core of the paradigm is the icon palette. You build a structure and flowchart of events, tasks, and decisions by dragging appropriate icons from icon palette library. These icons are used to represent and include menu choice, graphic images, sounds, computations, video, etc.
- ✓ The flow chart graphically depict the project logic
- ✓ Tends to be the speediest in development time. Because of this, they are best suited for rapid prototyping and short-development time projects.
- ✓ These tools are useful for storyboarding because you can change the sequence of objects, restructure interaction, add objects, by dragging and dropping icons.

Examples:

- Authorware
- IconAuthor

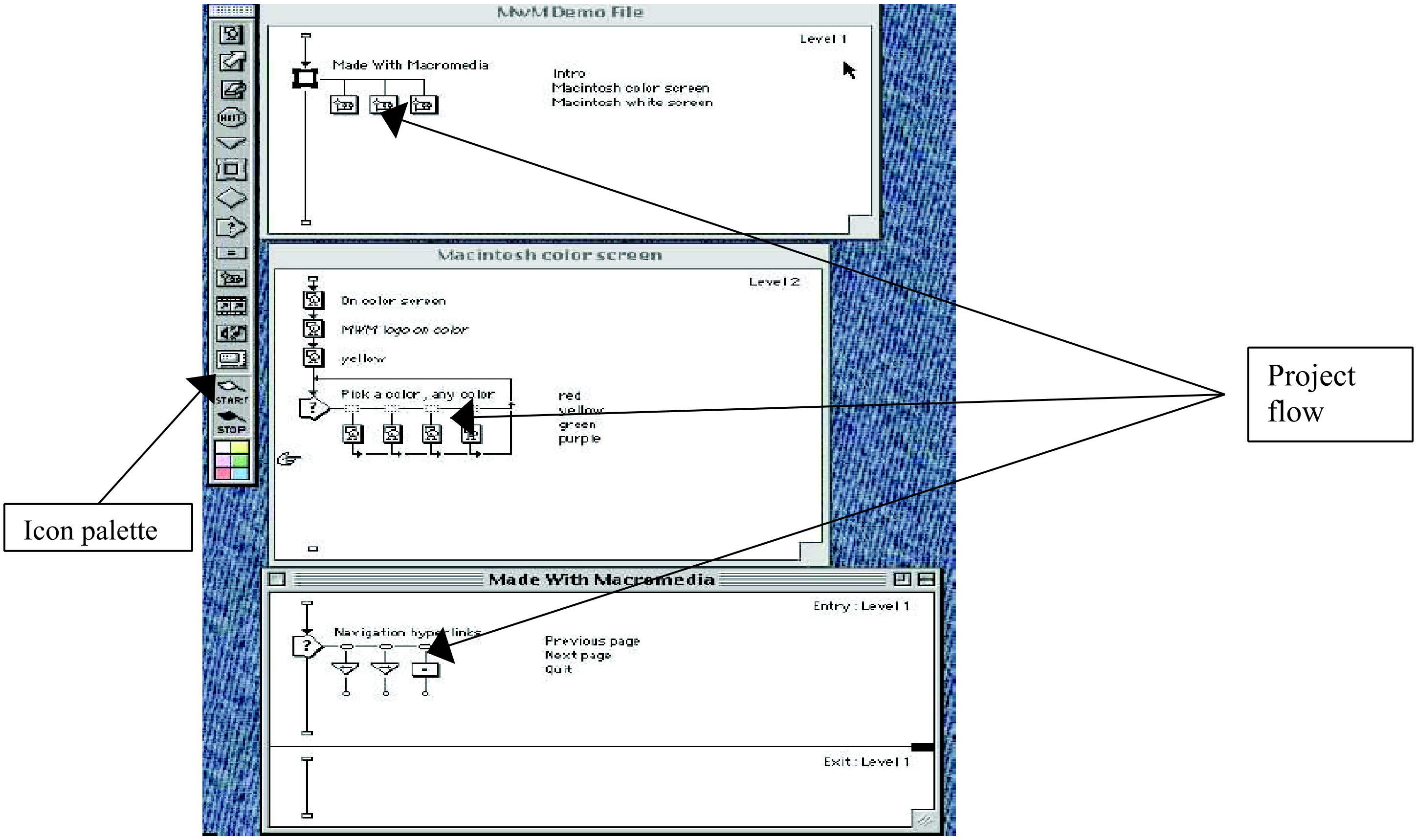


Fig 1 Macromedia Authorware Iconic/Flow Control Examples

Card and page Based Tools

In these authoring systems, elements are organised as pages of a book or a stack of cards. The authoring system lets you link these pages or cards into organised sequences. You can jump, on command, to any page you wish in a structured navigation pattern.

- ✓ Well suited for Hypertext applications, and especially suited for navigation intensive applications
- ✓ They are best suited for applications where the bulk of the content consist of elements that can be viewed individually
- ✓ Extensible via XCMDs (External Command) and DLLs (Dynamic Link Libraries).
- ✓ All objects (including individual graphic elements) to be scripted;
- ✓ Many entertainment applications are prototyped in a card/scripting system prior to compiled-language coding.
- ✓ Each object may contain programming script that is activated when an event occurs.

Examples:

- HyperCard (Macintosh)
- SuperCard(Macintosh)
- ToolBook (Windows), etc.

Time Based Authoring Tools

In these authoring systems elements are organised along a time line with resolutions as high as 1/30th second. Sequentially organised graphic frames are played back at a speed set by developer. Other elements, such as audio events, can be triggered at a given time or location in the sequence of events.

- ✓ Are the most popular multimedia authoring tool
- ✓ They are best suited for applications that have a message with beginning and end, animation intensive pages, or synchronized media application.

Examples

- Macromedia Director
- Macromedia Flash

Macromedia Director

Director is a powerful and complex multimedia authoring tool which has broad set of features to create multimedia presentation, animation, and interactive application. You can assemble and sequence the elements of project using cast and score. Three important things that Director uses to arrange and synchronize media elements:

Cast

Cast is multimedia database containing any media type that is to be included in the project. It imports wide range of data type and multimedia element formats directly into the cast. You can also create elements from scratch and add to cast. To include multimedia elements in cast into the stages, you drag and drop the media on the stage.

Score

This is where the elements in the cast are arranged. It is sequence for displaying, animating, and playing cast members. Score is made of frames and frames contain cast member. You can set frame rate per second.

Lingo

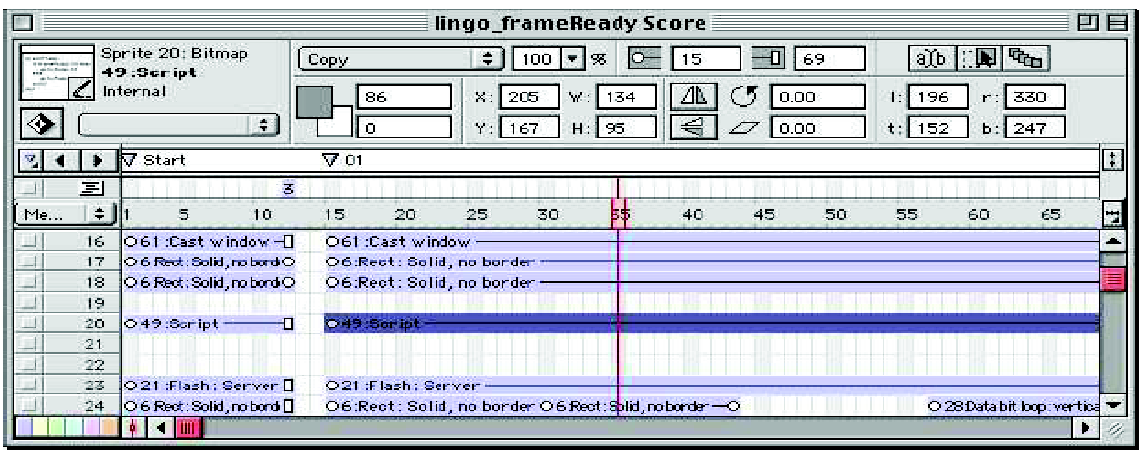
Lingo is a full-featured object oriented scripting language used in Director.

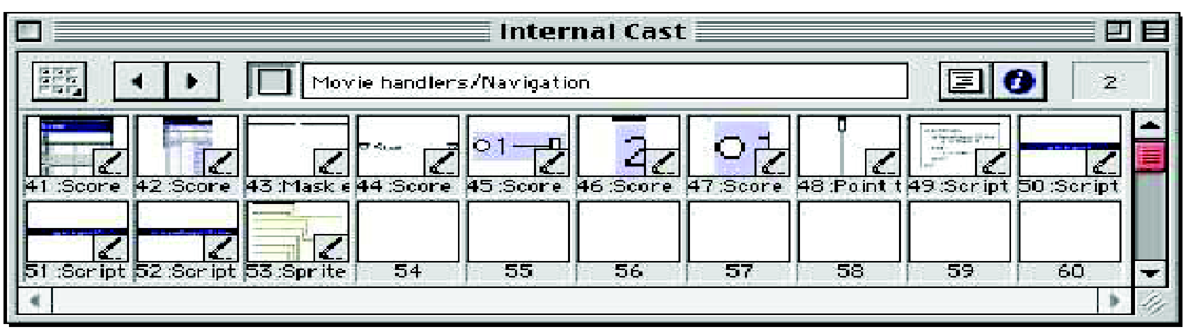
- ✓ It enables interactivity and programmed control of elements
- ✓ It enables to control external sound and video devices
- ✓ It also enables you to control operations of internet such as sending mail, reading documents, images, and building web pages.

Macromedia Flash

- ✓ Can accept both vector and bitmap graphics
- ✓ Uses a scripting language called ActionScript which gives greater capability to control the movie.
- ✓ Flash is commonly used to create animations, advertisements, to design web-page elements, to add video to web pages, and more recently, to develop Rich Internet Applications. Rich Internet Applications (RIA) are web applications that have the features and functionality of traditional desktop applications. RIA's uses a client side technology which can execute instructions on the client's computer (no need to send every data to the server).

Flash uses:
<u>Library</u>: a place where objects that are to be re-used are stored.
<u>Timeline:</u> used to organize and control a movie content over time.
<u>Layer:</u> helps to organize contents. Timeline is divided into layers.
<u>ActionScript</u>: enables ineractivity and control of movies





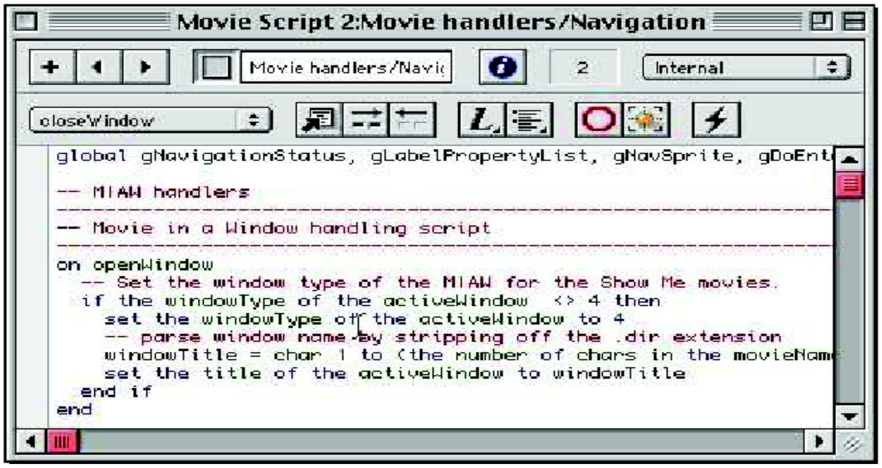


Fig 2 Macromedia Director Score, cast and Script windows respectively

Tagging

Tags in text files (e.g. HTML) to:
- ✓ link to pages,
- ✓ provide interactivity, and
- ✓ Integrate multimedia elements.

Examples:
- SGML/HTML
- SMIL (Synchronized Media Integration Language)
- VRML
- 3DML

- ✓ Most of them are displayed in web browsers using plug-ins or the browser itself can understand them.
- ✓ This metaphor is the basis of WWW
- ✓ It is limited but can be extended by the use of suitable multimedia tags

Selecting Authoring Tools

The multimedia project you are developing has its own underlying structure and purpose. When selecting tools for your project you need to consider that purpose.
Some of the features that you have to take into consideration when selecting authoring tools are:

1) Editing Feature: editing feature for multimedia data especially image and text are often included in authoring tools. The more editors in your authoring system, the less specialized editing tools you need. The editors that come with authoring tools offer only subset of features found in dedicated in editing tool. If you need more capability, still you have to go to dedicated editing tools (e.g. sound editing tools for sound editing).

2) Organizing feature: the organization of media in your project involves navigation diagrams, or flow charts, etc. Some authoring tools provides a visual flowcharting

facility. Such features help you for organizing the project.
e.g IconAuthor, and AuthorWare use flowcharting and navigation diagram method to organize media.

3) Programming feature: there are different types of programming approach:

i)Visual programming: this is programming using cues, icons, and objects. It is done using drag and drop. To include sound in your project, drag and drop it in stage.
Advantage: the simplest and easiest authoring process.
It is particularly useful for slide show and presentation.

ii) Programming with scripting language: Some authoring tool provide very high level scripting language and interpreted scripting environment. This helps for navigation control and enabling user input.

iii) Programming with traditional language such as Basic or C. Some authoring tools provide traditional programming tools like program written in C. We can call these programs to authoring tools. Some authoring tools allow to call DLL (Dynamic Link Library).

iv) Document development tools

4) Interactivity feature: interactivity offers to the end user of the project to control the content and flow of information. Some of interactivity levels:

i) Simple branching: enables the user to go to any location in the presentation using key press, mouse click, etc.
ii) conditional branching: branching based on if-then decisions
iii) Structured branching: support complex programming logic such as nested if-then sub-routines.

5) Performance-tuning features: accomplishing synchronization of multimedia is sometimes difficult because performance varies with different computers. In such cases you need to use authoring tools own scripting language to specify time and sequence on system.

6) Playback feature: easy testing of the project. Testing enables you to debug the system and find out how the user interacts with it.
- ✓ Not waste time in assembling and testing the project

7) Delivery feature: delivering your project needs building run-time version of the project using authoring tools. Why run time version (executable format):
- ✓ It does not require the full authoring software to play
- ✓ It does not allow users to access or change the content, structure, and programming of the project.

Distribute→run-time version

8) Cross platform feature: multimedia projects should be compatible with different platform like Macintosh, Windows, etc. This enables the designer to use any platform to design the project or deliver it to any platform.

9) Internet playability: web is significant delivery medium for multimedia. Authoring tools typically provide facility so that output can be delivered in HTML or DHTML format.

10) Ease of learning: is it easy to learn? The designer should not waste much time learning how to use it. Is it easy to use?

Chapter 3
Data Representations

Graphic/Image Data Representation

An image could be described as two-dimensional array of points where every point is allocated its own color. Every such single point is called pixel, short form of picture element. Image is a collection of these points that are colored in such a way that they produce meaningful information/data.

Pixel (picture element) contains the color or hue and relative brightness of that point in the image. The number of pixels in the image determines the resolution of the image.

- ✓ A digital image consists of many picture elements, called *pixels*.
- ✓ The number of pixels determines the quality of the image→image *resolution*.
- ✓ Higher resolution always yields better quality.
- ✓ Bitmap resolution most graphics applications let you create bitmaps up to 300 dots per inch (dpi). Such high resolution is useful for print media, but on the screen most of the information is lost, since monitors usually display around 72 to 96 dpi.
- ✓ A *bit-map* representation stores the graphic/image data in the same manner that the computer monitor contents are stored in video memory.
- ✓ Most graphic/image formats incorporate compression because of the large size of the data.

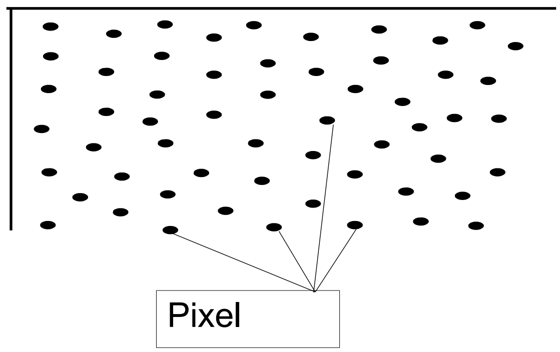


Fig 1 pixels

Types of images

There are two basic forms of computer graphics: bit-maps and vector graphics.
The kind you use determines the tools you choose. Bitmap formats are the ones used for digital photographs. Vector formats are used only for line drawings.

Bit-map images (also called Raster Graphics)
They are formed from pixels—a matrix of dots with different colors. Bitmap images are defined by their dimension in pixels as well as by the number of colors they represent. For example, a 640X480 image contains 640 pixels and 480 pixels in horizontal and vertical direction respectively. If you enlarge a small area of a bit-mapped image, you can clearly see the pixels that are used to create it (to check this open a picture in flash and change the magnification to 800 by going into View->magnification->800.).

Each of the small pixels can be a shade of gray or a color. Using 24-bit color, each pixel can be set to any one of 16 million colors. All digital photographs and paintings are bitmapped, and any other kind of image can be saved or exported into a bitmap format. In fact, when you print any kind of image on a laser or ink-jet printer, it is first converted by either the computer or printer into a bitmap form so it can be printed with the dots the printer uses.

To edit or modify bitmapped images you use a paint program. Bitmap images are widely used but they suffer from a few unavoidable problems. They must be printed or displayed at a size determined by the number of pixels in the image. Bitmap images also have large file sizes that are determined by the image's dimensions in pixels and its color depth. To reduce this problem, some graphic formats such as GIF and JPEG are used to store images in compressed format.

Vector graphics

They are really just a list of graphical objects such as lines, rectangles, ellipses, arcs, or curves—called *primitives*. Draw programs, also called vector graphics programs, are used to create and edit these vector graphics. These programs store the primitives as a set of numerical coordinates and mathematical formulas that specify their shape and position in the image. This format is widely used by computer-aided design programs to create detailed engineering and design drawings. It is also used in multimedia when 3D animation is desired. Draw programs have a number of advantages over paint-type programs.

These include:

- ✓ Precise control over lines and colors.
- ✓ Ability to skew and rotate objects to see them from different angles or add perspective.
- ✓ Ability to scale objects to any size to fit the available space. Vector graphics always print at the best resolution of the printer you use, no matter what size you make them.
- ✓ Color blends and shadings can be easily changed.
- ✓ Text can be wrapped around objects.

Monochrome/Bit-Map Images

- ✓ Each pixel is stored as a single bit (0 or 1)
- ✓ The value of the bit indicates whether it is light or dark
- ✓ A 640 x 480 monochrome image requires 37.5 KB of storage.
- ✓ Dithering is often used for displaying monochrome images

Fig 3 Monochrome bit-map image

Gray-scale Images

- ✓ Each pixel is usually stored as a byte (value between 0 to 255)
- ✓ This value indicates the degree of brightness of that point. This brightness goes from black to white
- ✓ A 640 x 480 grayscale image requires over 300 KB of storage.

Fig 4 Gray-scale bit-map image

8-bit Color Images

- ✓ One byte for each pixel
- ✓ Supports 256 out of the millions possible, acceptable color quality
- ✓ Requires Color Look-Up Tables (LUTs)
- ✓ A 640 x 480 8-bit color image requires 307.2 KB of storage (the same as 8-bit greyscale)
- ✓ Examples: GIF

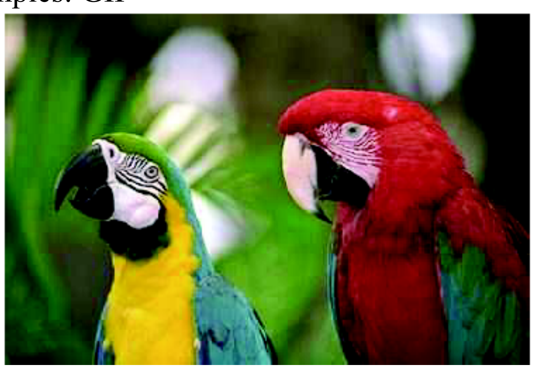


Fig 5 8-bit Color Image

24-bit Color Images

- ✓ Each pixel is represented by three bytes (e.g., RGB)
- ✓ Supports 256 x 256 x 256 possible combined colors (16,777,216)
- ✓ A 640 x 480 24-bit color image would require 921.6 KB of storage
- ✓ Most 24-bit images are 32-bit images,
    - the extra byte of data for each pixel is used to store an alpha value representing special effect information

Image Resolution

Image resolution refers to the spacing of pixels in an image and is measured in pixels per inch, ***ppi***, sometimes called dots per inch, ***dpi***. The higher the resolution, the more pixels in the image. A printed image that has a low resolution may look pixelated or made up of small squares, with jagged edges and without smoothness.
Image size refers to the physical dimensions of an image. Because the number of pixels in an image is fixed, increasing the size of an image decreases its resolution and decreasing its size increases its resolution.

Popular File Formats

Choosing the right file type for your image to save in is of vital importance. If you are, for example, creating image for web pages, then it should load fast. So such images should be small size. The other criteria to choose file type is taking into consideration the quality of the image that is possible using the chosen file type. You should also be concerned about the portability of the image.
To choose file type:

- ✓ resulting size of the image→large file size or small
- ✓ quality of image possible by the file type
- ✓ portability of file across different platforms

The most common formats used on internet are the GIF, JPG, and PNG.

Standard System Independent Formats

GIF

- ✓ **G**raphics **I**nterchange **F**ormat (GIF) devised CompuServe, initially for transmitting graphical images over phone lines via modems.
- ✓ Uses the Lempel-Ziv Welch algorithm (a form of Huffman Coding), modified slightly for image scan line packets (line grouping of pixels).
- ✓ LZW compression was patented technology by the UNISYS Corp.
- ✓ Limited to only 8-bit (256) color images, suitable for images with few distinctive colors (e.g., graphics drawing)
- ✓ Supports one-dimensional interlacing (downloading gradually in web browsers. Interlaced images appear gradually while they are downloading. They display at a low blurry resolution first and then transition to full resolution by the time the download is complete.)
- ✓ Supports animation—multiple pictures per file (animated GIF)
- ✓ GIF format has long been the most popular on the Internet, mainly because of its small size
- ✓ GIFs allow single-bit transparency, which means when you are creating your image, you can specify one colour to be transparent. This allows background colours to show through the image.

PNG

- ✓ stands for Portable Network Graphics
- ✓ It is intended as a replacement for GIF in the WWW and image editing tools.
- ✓ GIF uses LZW compression which is patented by Unisys. All use of GIF may have to pay royalties to Unisys due to the patent.
- ✓ PNG uses unpatented zip technology for compression
- ✓ One version of PNG, PNG-8, is similar to the GIF format. It can be saved with a maximum of 256 colours and supports 1-bit transparency. Filesizes when saved in a capable image editor like FireWorks will be noticeably smaller than the GIF counterpart, as PNGs save their colour data more efficiently.
- ✓ PNG-24 is another version of PNG, with 24-bit colour support, allowing ranges of colour to a high colour JPG. However, PNG-24 is in no way a replacement format for JPG, because it is a loss-less compression format which results in large filesize.
- ✓ Provides transparency using alpha value
- ✓ Supports interlacing
- ✓ PNG can be animated through the MNG extension of the format, but browser support is less for this format.

JPEG/JPG

- ✓ A standard for photographic image compression
- ✓ created by the Joint Photographic Experts Group
- ✓ Intended for encoding and compression of photographs and similar images
- ✓ Takes advantage of limitations in the human vision system to achieve high rates of compression
- ✓ Uses complex lossy compression which allows user to set the desired level of quality (compression). A compression setting of about 60% will result in the optimum balance of quality and filesize.
- ✓ Though JPGs can be interlaced, they do not support animation and transparency unlike GIF

TIFF

- ✓ Tagged Image File Format (TIFF), stores many different types of images (e.g., monochrome, grayscale, 8-bit & 24-bit RGB, etc.)
- ✓ Uses tags, keywords defining the characteristics of the image that is included in the file. For example, a picture 320 by 240 pixels would include a 'width' tag followed by the number '320' and a 'depth' tag followed by the number '240'.
- ✓ Developed by the Aldus Corp. in the 1980's and later supported by the Microsoft
- ✓ TIFF is a lossless format (when not utilizing the new JPEG tag which allows for JPEG compression)
- ✓ It does not provide any major advantages over JPEG and is not as user-controllable.
- ✓ Do not use TIFF for web images. They produce big files, and more importantly, most web browsers will not display TIFFs.

System Dependent Formats

Microsoft Windows: BMP

- ✓ A system standard graphics file format for Microsoft Windows
- ✓ Used in Many PC Graphics programs
- ✓ It is capable of storing 24-bit bitmap images

Macintosh: PAINT and PICT

- ✓ PAINT was originally used in MacPaint program, initially only for 1-bit monochrome images.
- ✓ PICT is a file format that was developed by Apple Computer in 1984 as the native format for Macintosh graphics.
- ✓ The PICT format is a meta-format that can be used for both bitmap images and vector images though it was originally used in MacDraw (a vector based drawing program) for storing structured graphics
- ✓ Still an underlying Mac format (although PDF on OS X)

X-windows: XBM

- ✓ Primary graphics format for the X Window system
- ✓ Supports 24-bit colour bitmap
- ✓ Many public domain graphic editors, e.g., xv
- ✓ Used in X Windows for storing icons, pixmaps, backdrops, etc.

Digital Audio and MIDI

What is Sound?

Sound is produced by a rapid variation in the average density or pressure of air molecules above and below the current atmospheric pressure. We perceive sound as these pressure fluctuations cause our eardrums to vibrate. These usually minute changes in atmospheric pressure are referred to as ***sound pressure*** and the fluctuations in pressure as ***sound waves***. Sound waves are produced by a vibrating body, be it a guitar string, loudspeaker cone or jet engine. The vibrating sound source causes a disturbance to the surrounding air molecules, causing them bounce off each other with a force proportional to the disturbance. The back and forth oscillation of pressure produces a sound waves.

Source — Generates Sound
- ✓ Air Pressure changes
- ✓ *Electrical* —Microphone produces electric signal
- ✓ *Acoustic* — Direct Pressure Variations

Destination — Receives Sound
- ✓ *Electrical* — Loud Speaker
- ✓ *Ears* — Responds to pressure hear sound

How to Record and Play Digital Audio

In order to play digital audio (i.e WAVE file), you need a card with a Digital To Analog Converter (DAC) circuitry on it. Most sound cards have both an ADC (Analog to Digital Converter) and a DAC so that the card can both record and play digital audio. This DAC is attached to the Line Out jack of your audio card, and converts the digital audio values back into the original analog audio. This analog audio can then be routed to a mixer, or speakers, or headphones so that you can hear the recreation of what was originally recorded. Playback process is almost an exact reverse of the recording process.

First, to record digital audio, you need a card which has an Analog to Digital Converter (ADC) circuitry. The ADC is attached to the Line In (and Mic In) jack of your audio card, and converts the incoming analog audio to a digital signal. Your computer software can store the digitized audio on your hard drive, visually display on the computer's monitor, mathematically manipulate in order to add effects, or process the sound, etc. While the incoming analog audio is being recorded, the ADC is creates many digital values in its conversion to a digital audio representation of what is being recorded. These values must be stored for later playback.

Digitizing Sound

- ✓ Microphone produces analog signal
- ✓ Computers understands only discrete(digital) entities

This creates a need to convert Analog audio to Digital audio — specialized hardware
This is also known as Sampling.

Common Audio Formats

There are two basic types of audio files: the traditional *discrete* audio file, that you can save to a hard drive or other digital storage medium, and the *streaming* audio file that you listen to as it downloads in real time from a network/internet server to your computer.

Discrete Audio File Formats

Common discrete audio file formats include WAV, AIF, AU and MP3. A fifth format, called MIDI is actually not a file format for storing digital audio, but a system of instructions for creating electronic music.

WAV

The WAV format is the standard audio file format for Microsoft Windows applications, and is the default file type produced when conducting digital recording within Windows. It supports a variety of bit resolutions, sample rates, and channels of audio. This format is very popular upon IBM PC (clone) platforms, and is widely used as a basic format for saving and modifying digital audio data.

AIF/AIFF

The Audio Interchange File Format (AIFF) is the standard audio format employed by computers using the Apple Macintosh operating system. Like the WAV format, it supports a variety of bit resolutions, sample rates, and channels of audio and is widely used in software programs used to create and modify digital audio.

AU

The AU file format is a compressed audio file format developed by Sun Microsystems and popular in the Unix world. It is also the standard audio file format for the Java programming language. Only supports 8-bit depth thus cannot provide CD-quality sound.

MP3

MP3 stands for Motion Picture Experts Group, Audio Layer 3 Compression. MP3 files provide near-CD-quality sound but are only about 1/10th as large as a standard audio CD file. Because MP3 files are small, they can easily be transferred across the Internet and played on any multimedia computer with MP3 player software.

MIDI/MID

MIDI (Musical Instrument Digital Interface), is not a file format for storing or transmitting recorded sounds, but rather a set of instructions used to play electronic music on devices such as synthesizers. MIDI files are very small compared to recorded audio file formats. However, the quality and range of MIDI tones is limited.

Streaming Audio File Formats

Streaming is a network technique for transferring data from a server to client in a format that can be continuously read and processed by the client computer. Using this method, the client computer can start playing the initial elements of large time-based audio or video files before the entire file is downloaded. As the Internet grows, streaming technologies are becoming an increasingly important way to deliver time-based audio and video data.

For streaming to work, the client side has to receive the data and continuously ‘feed’ it to the ‘player’ application. If the client receives the data more quickly than required, it has to temporarily store or ‘buffer’ the excess for later play. On the other hand, if the data doesn't arrive quickly enough, the audio or video presentation will be interrupted.
There are three primary streaming formats that support audio files: RealNetwork's RealAudio (RA, RM), Microsoft’s Advanced Streaming Format (ASF) and its audio subset called Windows Media Audio 7 (WMA) and Apple’s QuickTime 4.0+ (MOV).

RA/RM

For audio data on the Internet, the de facto standard is RealNetwork's RealAudio (.RA) compressed streaming audio format. These files require a RealPlayer program or browser plug-in. The latest versions of RealNetworks' server and player software can handle multiple encodings of a single file, allowing the quality of transmission to vary with the available bandwidth. Webcast radio broadcast of both talk and music frequently uses RealAudio. Streaming audio can also be provided in conjunction with video as a combined RealMedia (RM) file.

ASF

Microsoft's Advanced Streaming Format (ASF) is similar to designed to RealNetwork's RealMedia format, in that it provides a common definition for internet streaming media and can accommodate not only synchronized audio, but also video and other multimedia elements, all while supporting multiple bandwidths within a single media file. Also like RealNetwork's RealMedia format, Microsoft's ASF requires a program or browser plug-in.

The pure audio file format used in Windows Media Technologies is Windows Media Audio 7 (WMA files). Like MP3 files, WMA audio files use sophisticated audio compression to reduce file size. Unlike MP3 files, however, WMA files can function as either discrete or streaming data and can provide a security mechanism to prevent unauthorized use.

MOV

Apple QuickTime movies (MOV files) can be created without a video channel and used as a sound-only format. Since version 4.0, Quicktime provides true streaming capability. QuickTime also accepts different audio sample rates, bit depths, and offers full functionality in both Windows as well as the Mac OS.

Popular audio file formats are:

- au (*Unix*)
- aiff (*MAC*)
- wav (*PC*)
- mp3

MIDI

MIDI stands for Musical Instrument Digital Interface.

Definition of MIDI:

✓ MIDI is a protocol that enables computer, synthesizers, keyboards, and other musical device to communicate with each other. This protocol is a language that allows interworking between instruments from different manufacturers by providing a link that is capable of transmitting and receiving digital data. MIDI transmits only commands, it does not transmit an audio signal.

✓ It was created in 1982.

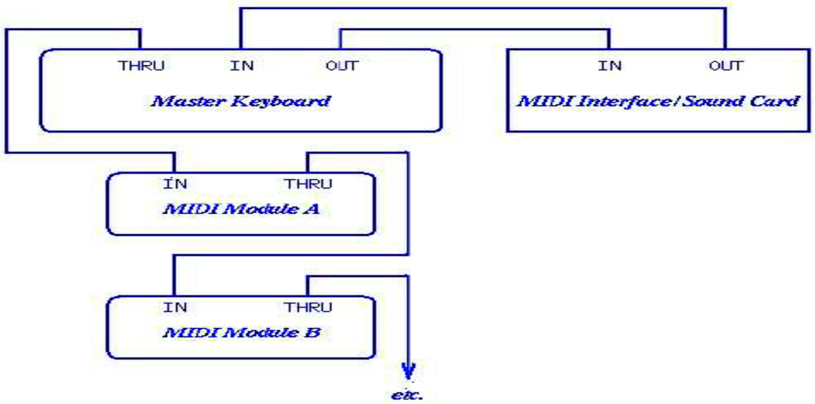


Figure 8 MIDI and Computer connection

Components of a MIDI System

Synthesizer:

• It is a sound generator (various pitch, loudness, tone color).

• A good (musician's) synthesizer often has a microprocessor, keyboard, control panels, memory, etc.

Sequencer:

✓ It can be a stand-alone unit or a software program for a personal computer. (It used to be a storage server for MIDI data. Nowadays it is more a software music editor on the computer.)

✓ It has one or more MIDI INs and MIDI OUTs.

Basic MIDI Concepts

Track:

✓ Track in sequencer is used to organize the recordings.

✓ Tracks can be turned on or off on recording or playing back.

Channel:

✓ MIDI channels are used to separate information in a MIDI system.

✓ There are 16 MIDI channels in one cable.

✓ Channel numbers are coded into each MIDI message.

Timbre:

✓ The quality of the sound, e.g., flute sound, cello sound, etc.

✓ Multitimbral – capable of playing many different sounds at the same time (e.g., piano, brass, drums, etc.)

Pitch:

✓ The Musical note that the instrument plays

Voice:

✓ Voice is the portion of the synthesizer that produces sound.

✓ Synthesizers can have many (12, 20, 24, 36, etc.) voices.

- ✓ Each voice works independently and simultaneously to produce sounds of
- ✓ Different timbre and pitch.

Patch:

- ✓ The control settings that define a particular timbre.

Hardware Aspects of MIDI

MIDI connectors:
Three 5-pin ports found on the back of every MIDI unit
MIDI IN: the connector via which the device receives all MIDI data.
MIDI OUT: the connector through which the device transmits all the MIDI data it generates itself.
MIDI THROUGH: the connector by which the device echoes the data receives from MIDI IN.
(See picture 8 for diagrammatical view)

MIDI Messages

MIDI messages are used by MIDI devices to communicate with each other.
MIDI messages are very low bandwidth:

- Note On Command
  - Which Key is pressed
  - Which MIDI Channel (what sound to play)
  - 3 Hexadecimal Numbers
- Note Off Command Similar
- Other command (program change) configure sounds to be played.

Advantages:

- ✓ Because MIDI is a digital signal, it's very easy to interface electronic instruments to computers, and then do manipulations on the MIDI data on the computer with software. For example, software can store MIDI messages to the computer's disk drive. Also, the software can playback MIDI messages upon all 16 channels with the same rhythms as the human who originally caused the instrument(s) to generate those messages.

How is MIDI file Different from a WAV or MP3 Files?

A MIDI file stores MIDI messages. These messages are commands that tell a musical device what to do in order to make music. For example, there is a MIDI message that tells a device to play a particular note. There is another MIDI message that tells a device to change its current "sound" to a particular patch or instrument. Etc.

The MIDI file also stores timestamps, and other information that a sequencer needs to play some "musical performance" by transmitting all of the MIDI messages in the file to all MIDI devices. In other words, a MIDI file contains hundreds (to thousands) of instructions that tell one or more sound modules (either external ones connected to your sequencer's MIDI Out, or sound modules built into your computer's sound card) how to reproduce every single, individual note and nuance of a musical performance.

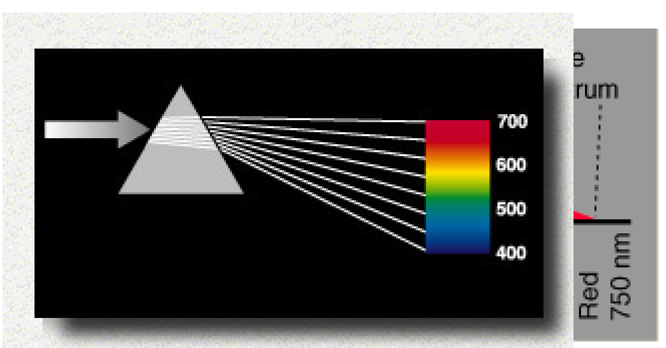
A WAVE and MP3 files store a digital audio waveform. This data is played back by a device with a Digital to Analog Converter (DAC) such as computer sound card's DAC. There are no timestamps, or other information concerning musical rhythms or tempo stored in a WAVE or MP3 files. There is only digital audio data.

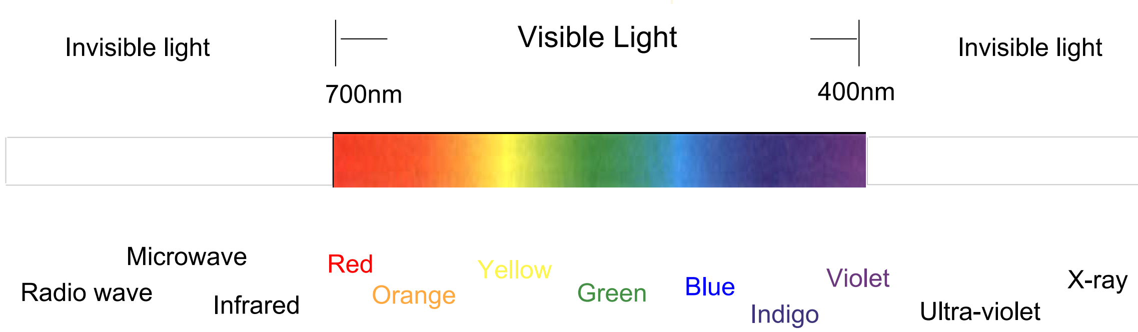
Chapter 4
Color in Image and Video

Color in Image and Video — Basics of Color

Light and Spectra

In 1672, Isaac Newton discovered that white light could be split into many colors by a prism. The colors produced by light passing through prism are arranged in precise array or spectrum. The colors' spectral signature is identified by its wavelength.





Visible light is an electromagnetic wave in the 400nm – 700 nm range (Blue~400nm, Red~700nm, Green~500nm). Most light we see is not one wavelength, it's a combination of many wavelengths. For Example purple is a mixture of red and violet. $1nm=10^{-9}m$

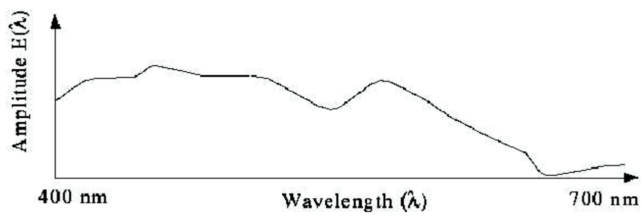


Fig Light wavelengths

The profile above is called a spectrum.

The Color of Objects

Here we consider the color of an object illuminated by white light. Color is produced by the absorption of selected wavelengths of light by an object. Objects can be thought of as absorbing all colors except the colors of their appearance which are reflected back. A blue object illuminated by white light absorbs most of the wavelengths except those corresponding to blue light. These blue wavelengths are reflected by the object.

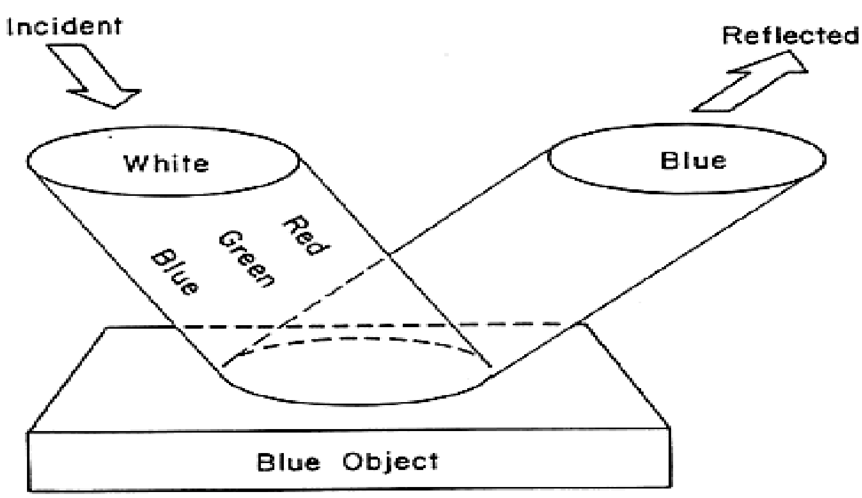


Fig White light composed of all wavelengths of visible light incident on a pure blue object. Only blue light is reflected from the surface.

The Eye and Color Sensation

Our perception of color arises from the composition of light - the energy spectrum of photons - which enter the eye. The retina on the inner surface of the back of the eye contains photosensitive cells. These cells contain pigments which absorb visible light. Two types of photosensitive cells

- ✓ Cones

- ✓ Rods

Rods: are not sensitive to color. They are sensitive only to intensity of light. They are effective in dim light and sense differences in light intensity - the flux of incident photons. Because rods are not sensitive to color, in dim light we perceive colored objects as shades of grey, not shades of color.

Cones: allow us to distinguish between different colors. Three types of cones:

- ✓ Red cones: responds to red light
- ✓ Green cones : respond to green light
- ✓ Blue cones: responds to blue light

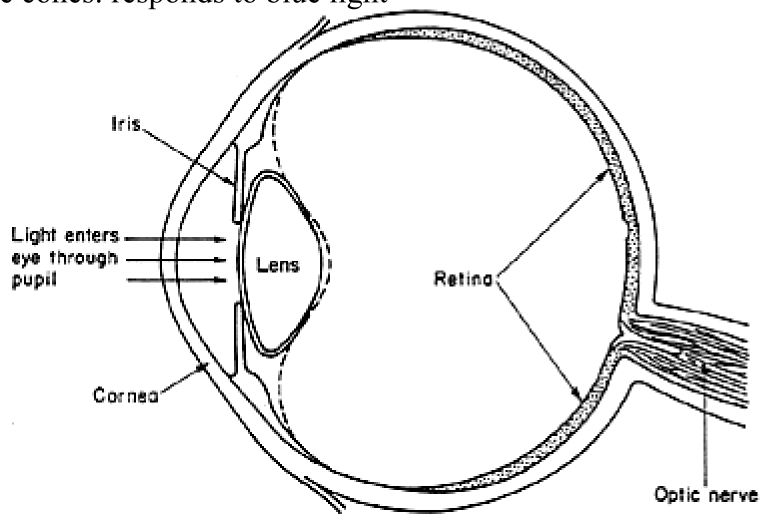


Fig cross-sectional representation of the eye showing light entering through the pupil

The color signal to the brain comes from the response of the three cones to the spectra being observed. That is, the signal consists of 3 numbers:

- Red
- Green
- Blue

✓ A color can be specified as the sum of three colors. So colors form a 3 dimensional vector space.

For every color signal or photons reaching the eye, some ratio of response within the three types of cones is triggered. It is this ratio that permits the perception of a particular color.

- ✓ The following figure shows the spectral-response functions of the cones and the luminous-efficiency function of the human eye.
- ✓ Eye responds differently to changes in different color and luminance.

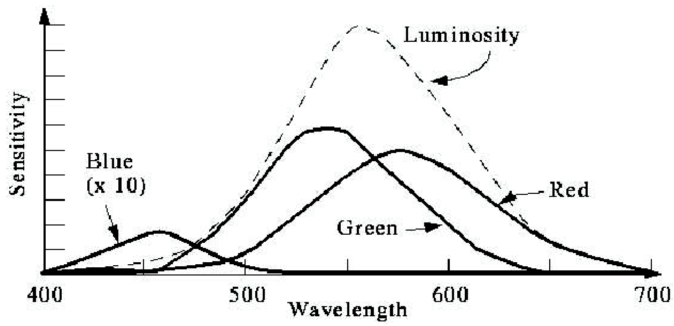


Fig Cones and luminous-efficiency function of the human eye

Color Spaces

Color space specifies how color information is represented. It is also called color model. Any color could be described in a three dimensional graph, called a color space. Mathematically the axis can be tilted or moved in different directions to change the way the space is described, without changing the actual colors. The values along an axis can be linear or non-linear. This gives a variety of ways to describe colors that have an impact on the way we process a color image.

There are different ways of representing color. Some of these are:

- ✓ RGB color space
- ✓ YUV color space
- ✓ YIQ color space
- ✓ CMY/CMYK color space
- ✓ CIE color space
- ✓ HSV color space
- ✓ HSL color space
- ✓ YCbCr color space

RGB Color Space

RGB stands for Red, Green, Blue. RGB color space expresses/defines color as a mixture of three primary colors:

- ✓ Red
- ✓ Green
- ✓ Blue

All other colors are produced by varying the intensity of these three primaries and mixing the colors. It is used self-luminous devices such as TV, monitor, camera, and scanner.

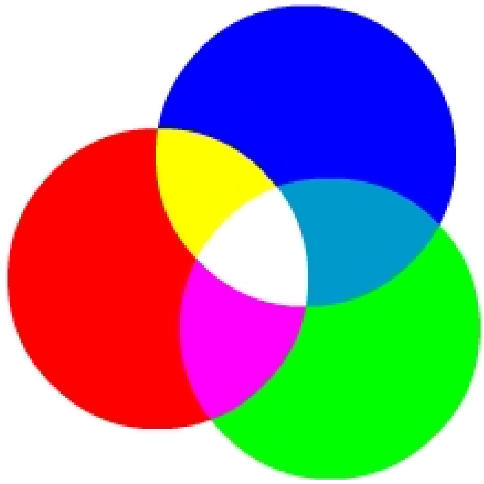


Fig RGB color model

Color images can be described with three components, commonly Red, Green, and Blue. It combines (adds) the three components with varying intensity to make all other colors. Absence of all colors (zero values for all the components) create black. The presence of the three colors form white. These colors are called additive colors since they add together the way light adds to make colors, and is a natural color space to use with video displays.

Grey is any value where R=G=B, thus it requires all three (RGB) signals to produce a "black and white" picture. In other words, a "black and white" picture must be computed - it is not inherently available as one of the components specified.

Pure black (0,0,0)
Pure white(255,255,255)

CRT Displays

- ✓ CRT displays have three phosphors (RGB) which produce a combination of wavelengths when excited with electrons.
- ✓ The gamut of colors is all colors that can be reproduced using the three Primaries.
- ✓ The gamut of a color monitor is smaller than that of color models, E.g. CIE (LAB) Model

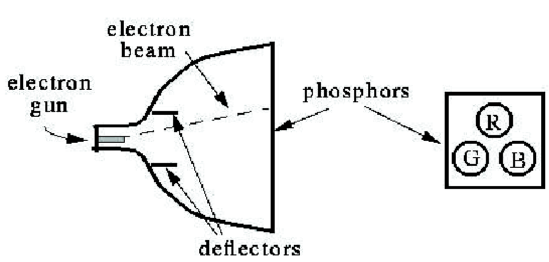


Fig CRT display

CYM and CYMK

A color model used with printers and other peripherals. Three primary colors, cyan (C), magenta (M), and yellow (Y), are used to reproduce all colors.

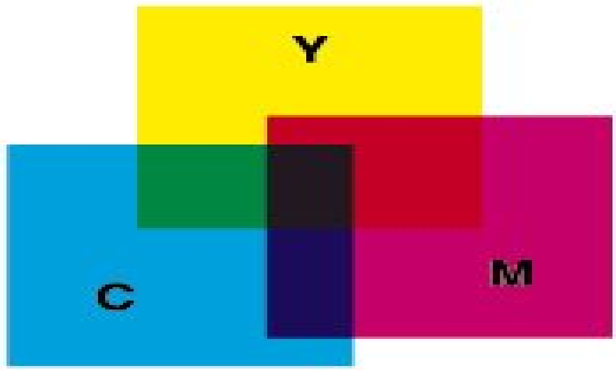


Fig CMY color space

The three colors together absorb all the light that strikes it, appearing black (as contrasted to RGB where the three colors together made white). "Nothing" on the paper is white (as contrasted to RGB where nothing was black). These are called the subtractive or "paint" colors.

In practice, it is difficult to have the exact mix of the three colors to perfectly absorb all light and thus produce a black color. Expensive inks are required to produce the exact color, and the paper must absorb each color in exactly the same way. To avoid these problems, a forth color is often added - black - creating the CYMK color "space", even though the black is mathematically not required.

YCbCr

This color space is closely related to the YUV space, but with the coordinates shifted to allow all positive valued coefficients. It is a scaled and shifted YUV.

- ✓ The luminance (brightness), Y, is retained separately from the chrominance (color).

Y-Luma component
Cb
Cr Chrominace

During development and testing of JPEG it became apparent that chrominance sub sampling in this space allowed a much better compression than simply compressing RGB or CYM. Sub sampling means that only one half or one quarter as much detail is retained for the color as for the brightness.

- ✓ It is used in MPEG and JPEG compressions

Y = + 0.299 * R + 0.587 * G + 0.114 * B
Cb = 128 - 0.168736 * R - 0.331264 * G + 0.5 * B
Cr = 128 + 0.5 * R - 0.418688 * G - 0.081312 * B

CIE

In 1931, the CIE (Commite Internationale de E'clairage) developed a color model based on human perception. They are based on the human eyes' response to red green and blue colors, and are designed to accurately represent human color perception. The CIE is a device-independent color model and because of this it is used as a standard for other colors to compare with. Device-independent means color can be reproduced faithfully on any type of device, such as scanners, monitors, and printers (color quality does not vary depending on the device).

There are different versions of CIE color model. The most commonly used are:
- ✓ CIE XYZ color model
- ✓ CIE L*a*b color model

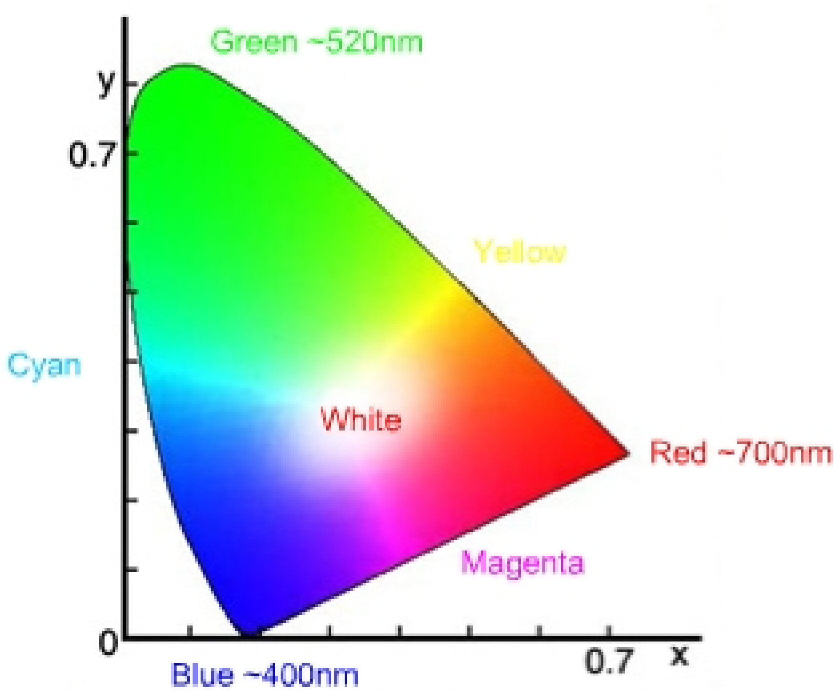


Fig CIE color model

CIE XYZ

CIE XYZ color model defines three primaries called X, Y, and Z that can be combined to match any color humans see. This relates to color perception of human eye. The Y primary is defined to match the luminous efficiency of human eye. X and Z are obtained based on experiment involving human observers.

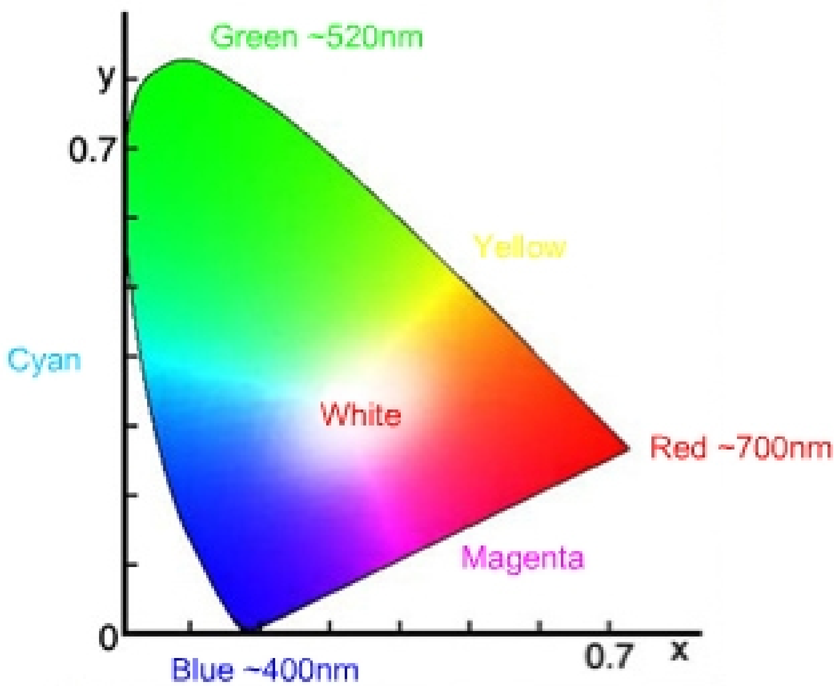


Fig CIE XYZ chromacity diagram

- ✓ Edges represent pure colors
- ✓ Every color could be assigned a particular point on the coordinate plane
- ✓ The spectral purity of colors decreases as you move from the edges to the center of the diagram
- ✓ Brightness is not taken into consideration in this model

CIE Lab Color Model

- ✓ A refined CIE model, named CIE L*a*b is introduced in 1976
- ✓ It is an improvement of CIE XYZ color model

L– represents Luminance
a – ranges from green to red
b – ranges from blue to yellow   represents chrominance

- ✓ Used by Photoshop

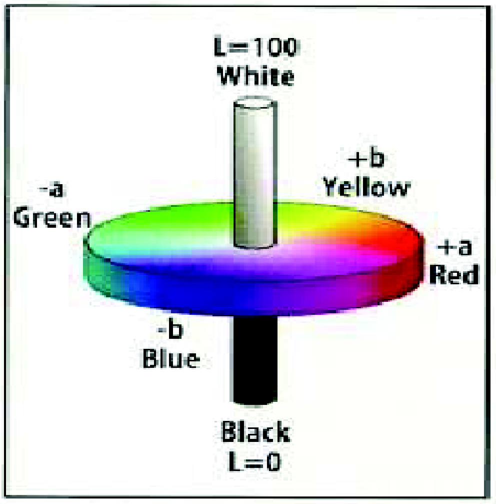


Fig LAB model

The color space choice

In August 1991, the International Group 4 color fax Committee decided to assume YCbCr would be the standard as they continued their studies. They noted that YCbCr was mandatory for compatibility with business image systems such as desktop publishing. For professional graphics, it was mandatory along with CIELAB for calibration. At the high end of publishing, many color spaces had to be supported, including YCbCr. In fact, YCbCr was the most widely used color space in all areas.

By the November 1992 Group 4 color fax meeting in Tokyo, CIELAB 1976 was selected as the primary color space, with YCbCr as one of several secondary options. Some of the people involved argue that the particular meeting was dominated by people with special interests, and don't believe that decision will stand.

If CIELAB becomes the fax standard, it would logically be our choice. However, YCbCr is much more widely used, and preferred by many technical experts.

Beside the RGB representation, YIQ and YUV are the two commonly used in video.

YIQ Color Model

YIQ is used in color TV broadcasting, it is downward compatible with Black and White TV. The YIQ color space is commonly used in North American television systems. Note that if the chrominance is ignored, the result is a "black and white" picture.

- ✓ Y (luminance) is the CIE Y primary
  $Y = 0.299R + 0.587G + 0.114B$
- ✓ the other two vectors
  $I = 0.596R - 0.275G - 0.321B$
  $Q = 0.212R - 0.528G + 0.311B$

✓ I is red-orange axis, Q is roughly orthogonal to I.
✓ Eye is most sensitive to Y (luminance), next to I, next to Q.

YIQ is intended to take advantage of human color response characteristics. Eye is more sensitive to Orange-Blue range (I) than in Purple-Green range (Q). Therefore less bandwidth is required for Q than for I. NTSC limits I to 1.5 MHZ and Q to 0.6 MHZ. Y is assigned higher bandwidth, 4MHZ.

YUV Color Model

✓ Established in 1982 to build digital video standard
✓ Works in PAL (50 fields/sec) or NTSC (60 fields/sec)
✓ The luminance (brightness), Y, is retained separately from the chrominance (color)

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The Y component determines the brightness of the color (referred to as luminance or luma), while the U and V components determine the color itself (it is called chroma). U is the axis from blue to yellow and V is the axis from magenta to cyan. Y ranges from 0 to 1 (or 0 to 255 in digital formats), while U and V range from -0.5 to 0.5 (or -128 to 127 in signed digital form, or 0 to 255 in unsigned form).

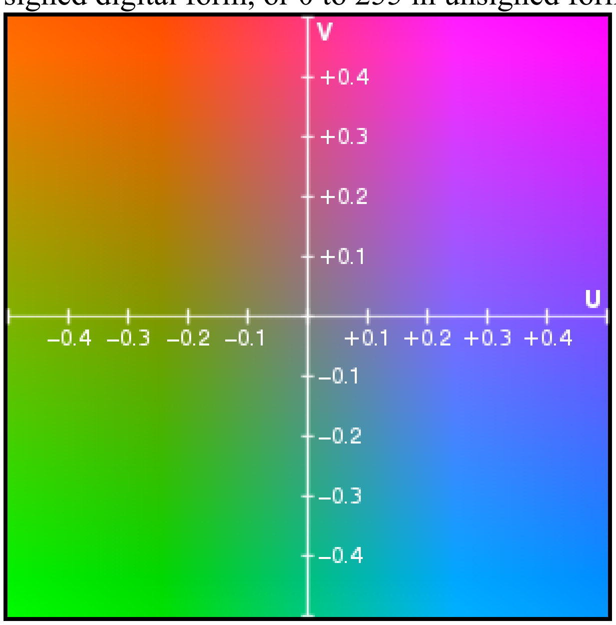
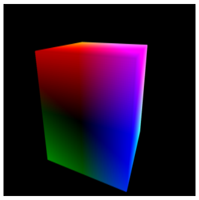


Fig YUV color model

One neat aspect of YUV is that you can throw out the U and V components and get a grey-scale image. Black and white TV receives only Y (luminanace) component ignoring the otheres. This makes it black-white TV compatible. Since the human eye is more responsive to brightness than it is to color, many lossy image compression formats throw away half or more of the samples in the chroma channels (color part) to reduce the amount of data to deal with, without severely destroying the image quality.



This image shows a slightly tilted representation of the YUV color cube, looking at the dark (Y = 0) side. Notice how in the middle it is completely black, which is where U and V are zero, and Y is as well. As U and V move towards their limits, you start to see their effect on the colors.

This image shows the same cube, from the bright side (Y = 1). Here we have bright white in the middle of the face, with very bright colors on the corners where U and V are also at their limits.

$Y = R * 0.299 + G * 0.587 + B * 0.114$
$U = R * -0.169 + G * -0.332 + B * 0.500$
$V = R * 0 .500 + G * -0.419 + B * -0.0813$

- ✓ The YUV color space is commonly used in European television.
- ✓ if the chrominance is ignored, the result is a "black and white" picture.

The CMY Color Model

- ✓ Cyan, Magenta, and Yellow (CMY) are complementary colors of RGB. They can be used as Subtractive Primaries.
- ✓ CMY model is mostly used in printing devices where the color pigments on the paper absorb certain colors (e.g., no red light reflected from cyan ink) and in painting.

Fig RGB and CMY cubes

Cyan, magenta, and yellow are used as subtractive primaries

Conversion between RGB and CMY
E.g., convert *White* from (1, 1, 1) in RGB to (0, 0, 0) in CMY

$C=1-R$
$M=1-G$
$Y=1-B$

CMYK color model

- ✓ Sometimes, an alternative CMYK model (K stands for Black) is used in color printing (e.g., to produce darker black than simply mixing CMY), where

  K = min(C, M, Y),
  C = C - K,
  M = M - K,
  Y = Y - K.

Colors on self-luminous devices, such as televisions and computer monitors, are produced by adding the three RGB primary colors in different proportions. However, color reproduction media, such as printed matter and paintings, produce colors by absorbing some wavelengths and reflecting others.

The three RGB primary colors, when mixed, produce white, but the three CMY primary colors produce black when they are mixed together. Since actual inks will not produce pure colors, black (K) is included as a separate color, and the model is called CMYK. With the CMYK model, the range of reproducible colors is narrower than with RGB, so when RGB data is converted to CMYK data, the colors seem dirtier.

HSL Color Space

HSL stands for Hue Saturation Lightness.
H-represents hue (color)
S-represents saturation. It goes from fully saturated color to equivalent gray.

Fig HSL color space

The **HSL** color space stands for Hue, Saturation, Lightness (also luminance or *luminosity*). HSL is drawn as a double cone or double hexcone. The two apexes of the HSL double hexcone correspond to black and white. The angular parameter corresponds to hue, distance from the axis corresponds to saturation, and distance along the black-white axis corresponds to lightness.

HSV Color Space (also called HSB**)**

Stands for Hue Saturation Value.
H-represents color type (red, blue, yellow). It ranges from 0-360 degrees.
Saturation-the vibrancy of color. It ranges from 0-100%.
Value-brightness of color. It ranges from 0-100%.

Fig HSV color model

Summary of Color

- ✓ Color images are encoded as (R,G,B) integer triplet values. These triplets encode how much the corresponding phosphor should be excited in devices such as a monitor.

- ✓ Three common systems of encoding in video are *RGB*, *YIQ*, and *YcrCb(YUV)*.
- ✓ Besides the hardware-oriented color models (i.e., RGB, CMY, YIQ, YUV), HSB (Hue, Saturation, and Brightness, e.g., used in Photoshop) and HLS (Hue, Lightness, and Saturation) are also commonly used.
- ✓ YIQ uses properties of the human eye to prioritize information. Y is the black and white (luminance) image; I and Q are the color (chrominance) images. YUV uses similar idea.
- ✓ YUV is a standard for digital video that specifies image size, and decimates the chrominance images (for 4:2:2 video)
- ✓ A black and white image is a 2-D array of integers.

Chapter 5
Fundamental Concepts in Video

- ✓ Video is a series of images. When this series of images are displayed on screen at fast speed (e.g 30 images per second), we see a perceived motion. It projects single images at a fast rate producing the illusion of continuous motion. These single images are called frames. The rate at which the frames are projected is generally between 24 and 30 frames per second (fps). The rate at which these images are presented is referred to as the Frame Rate
- ✓ This is fundamental to the way video is modeled in computers.
- ✓ A single image is called frame and video is a series of frames.
- ✓ An image just like conventional images is modeled as a matrix of pixels.
- ✓ To model smooth motion psychophysical studies have shown that a rate of 30 frames a second is good enough to simulate smooth motion.
    - Old Charlie Chaplin movies were taken at 12 frames a second and are visibly jerky in nature.

Each screen-full of video is made up of thousands of pixels. A pixel is the smallest unit of an image. A pixel can display only one color at a time. Your television has 720 vertical lines of pixels (from left to right) and 486 rows of pixels (top to bottom). A total of 349,920 pixels (720 x 486) for a single frame.

There are two types of video:

- ✓ Analog Video
- ✓ Digital Video

Analog Video

Analog technology requires information representing images and sound to be in a real-time continuous-scale electric signal between sources and receivers. It is used throughout the television industry. For television, images and sound are converted into electric signals by transducers. Distortion of images and noise are common problems for analog video.

In an analogue video signal, each frame is represented by a fluctuating voltage signal.

This is known as an analogue waveform. One of the earliest formats for this was composite video.

Analog formats are susceptible to loss due to transmission noise effects. Quality loss is also possible from one generation to another. This type of loss is like photocopying, in which a copy of a copy is never as good as the original.

Digital Video

Digital technology is based on images represented in the form of bits. A digital video signal is actually a pattern of 1's and 0's that represent the video image. With a digital video signal, there is no variation in the original signal once it is captured on to computer disc. Therefore, the image does not lose any of its original sharpness and clarity. The image is an exact copy of the original. A computer is the must common form of digital technology.

The limitations of analog video led to the birth of digital video. Digital video is just a digital representation of the analogue video signal. Unlike analogue video that degrades in quality from one generation to the next, digital video does not degrade. Each generation of digital video is identical to the parent.

Even though the data is digital, virtually all digital formats are still stored on sequential tapes. There are two significant advantages for using computers for digital video :

- ✓ the ability to randomly access the storage of video and
- ✓ compress the video stored.

Computer-based digital video is defined as a series of individual images and associated audio. These elements are stored in a format in which both elements (pixel and sound sample) are represented as a series of binary digits (bits).

Advantages:

- — Direct random access –> good for nonlinear video editing
- — No problem for repeated recording
- — No need for blanking and sync pulse

✓ Almost all digital video uses component video

Analog vs. Digital Video

An analog video can be very similar to the original video copied, but it is not identical. Digital copies will always be identical and will not loose their sharpness and clarity over time. However, digital video has the limitation of the amount of RAM available, whereas this is not a factor with analog video. Digital technology allows for easy editing and enhancing of videos. Storage of the analog video tapes is much more cumbersome than digital video CDs. Clearly, with new technology continuously emerging, this debate will always be changing.

Displaying Video

There are two ways of displaying video on screen:

- ✓ Progressive scan
- ✓ Interlaced scan

Interlaced Scanning

Interlaced scanning writes every second line of the picture during a scan, and writes the other half during the next sweep. Doing that we only need 25/30 pictures per second. This idea of splitting up the image into two parts became known as interlacing and the splitted up pictures as fields. Graphically seen a field is basically a picture with every 2nd line black/white. Here is an image that shows interlacing so that you can better imagine what happens.

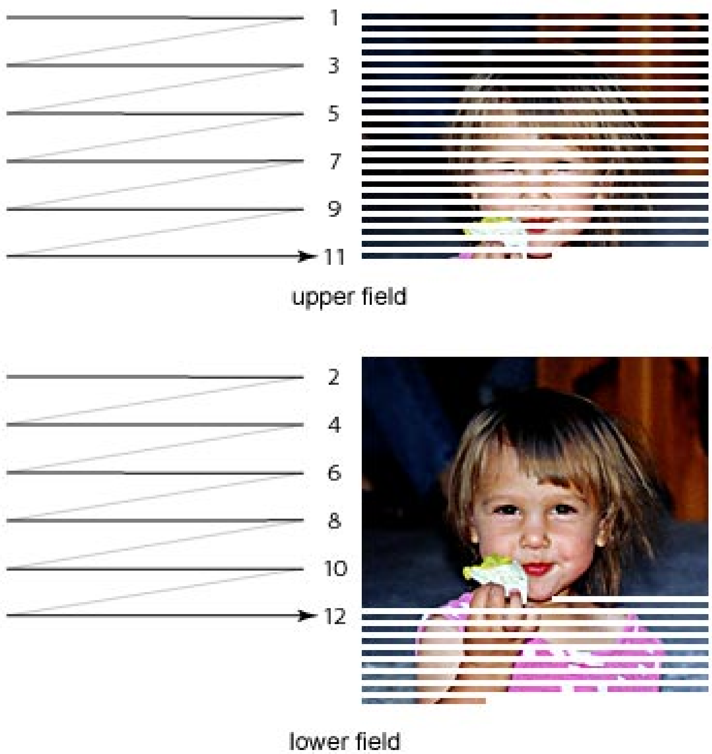


Fig interlaced scanning

During the first scan the upper field is written on screen. The first, 3rd, 5th, etc. line is written and after writing each line the electron beam moves to the left again before writing the next line.

Currently the picture exhibits a "combing" effect, it looks like you're watching it through a comb. When people refer to interlacing artifacts or say that their picture is interlaced this is what they commonly refer to.

Once all the odd lines have been written the electron beam travels back to the upper left of the screen and starts writing the even lines. As it takes a while before the phosphor stops emitting light and as the human brain is too slow instead of seeing two fields what we see is a combination of both fields - in other words the original picture.

Progressive Scanning

PC CRT displays are fundamentally different from TV screens. Monitor writes a whole picture per scan. Progressive scan updates all the lines on the screen at the same time, 60 times every second. This is known as progressive scanning. Today all PC screens write a picture like this.

Fig progressive scanning

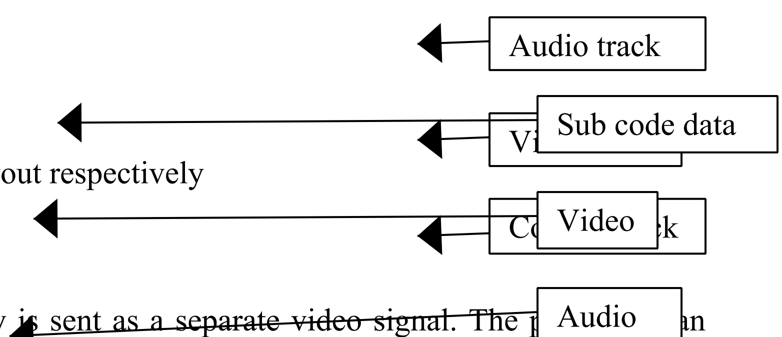
Here is a comparison of computer and television display.

Computer	Television
✓ Scans 480 horizontal lines from top to bottom	Scans 625, 525 horizontal lines
✓ Scan each line progressively	Scan line using interlacing system
✓ Scan full frame at a rate of typically 66.67 HZ or higher	Scan 25-30 HZ for full time
✓ Use RGB color model	Uses limited color palette and restricted luminance (lightness or darkness)

Recording Video

CCDs (Charge Coupled Devices) a chip containing a series of tiny, light-sensitive photosites. It forms the heart of all electronic and digital cameras. CCDs can be thought of as film for electronic cameras. CCDs consist of thousands or even millions of cells, each of which is light-sensitive and capable of producing varying amounts of charge in response to the amount of light they receive.

Digital camera uses lens which focuses the image onto a Charge Coupled Device (CCD), which then converts the image into electrical pulses. These pulses are then saved into memory. In short, Just as the film in a conventional camera records an image when light hits it, the CCD records the image electronically. The photosites convert light into electrons. The electrons pass through an analog-to-digital converter, which produces a file of encoded digital information in which bits represent the color and tonal values of a subject. The performance of a CCD is often measured by its output resolution, which in turn is a function of the number of photosites on the CCD's surface.



Analog and digital video frame layout respectively

Types of Color Video Signals

Component video – each primary is sent as a separate video signal. The p[illegible]an either be RGB or a luminance-chrominance transformation of them (e.g., YIQ, YUV).

— Best color reproduction

—Requires more bandwidth and good synchronization of the three components

Component video takes the different components of the video and breaks them into separate signals. Improvements to component video have led to many video formats, including S-Video, RGB etc.

Composite video – color (chrominance) and luminance signals are mixed into a single carrier wave. Some interference between the two signals is inevitable.

Composite analog video has all its components (brightness, color, synchronization information, etc.) combined into one signal. Due to the *compositing* (or combining) of the video components, the quality of composite video is marginal at best. The results are color bleeding, low clarity and high generational loss.

S-Video (Separated video) – a compromise between component analog video and the composite video. It uses two lines, one for luminance and another for composite chrominance signal.

Video Broadcasting Standards/ TV standards

There are three different video broadcasting standards: *PAL*, *NTSC*, and *SECAM*

PAL (Phase Alternate Line)

PAL uses 625 horizontal lines at a field rate of 50 fields per second (or 25 frames per second). Only 576 of these lines are used for picture information with the remaining 49 lines used for sync or holding additional information such as closed captioning. It is used in Australia, New Zealand, United Kingdom, and Europe.

- ✓ Scans 625 lines per frame, 25 frames per second (40 msec/frame)
- ✓ Interlaced, each frame is divided into 2 fields, 312.5 lines/field
- ✓ For color representation, PAL uses YUV (YCbCr) color model
- ✓ In PAL,
    - 5.5 MHz is allocated to Y,
    - 1.8 MHz each to U and V

SECAM (Sequential Color with Memory)

SECAM uses the same bandwidth as PAL but transmits the color information sequentially. It is used in France, East Europe, etc

SECAM (Systeme Electronic Pour Couleur Avec Memoire) is very similar to PAL. It specifies the same number of scan lines and frames per second. It is the broadcast standard for France, Russia, and parts of Africa and Eastern Europe.

NTSC (National Television Standards Committee)

NTSC is a black-and-white and color compatible 525-line system that scans a nominal 30 interlaced television picture frames per second. Used in USA, Canada, and Japan.

• 525 scan lines per frame, 30 frames per second (or be exact, 29.97 fps, 33.37 sec/frame)

• Interlaced, each frame is divided into 2 fields, 262.5 lines/field

• 20 lines reserved for control information at the beginning of each field (Fig. 38)

– So a maximum of 485 lines of visible data

NTSC Video

- ✓ 525 scan lines per frame, 30 frames per second
- ✓ Interlaced, each frame is divided into 2 fields i.e. 262.5 lines/field
- ✓ 20 lines reserved for control information at the beginning of each field
    - — So a maximum of 485 lines of visible data

NTSC Video Scan Line

- ✓ Each line takes 63.5 microseconds to scan. Horizontal retrace takes 10 microseconds (with 5 microseconds horizontal synch pulse embedded), so the active line time is 53.5 microseconds.

Figure Digital video raster

NTSC Video Color Representation/Compression

- ✓ For color representation, NTSC uses YIQ color model.
- ✓ Basic Compression Idea

  Eye is most sensitive to Y, next to I, next to Q.

  This is still Analog Compression:
- ✓ In NTSC,
    - — 4 MHz is allocated to Y,
    - — 1.5 MHz to I,
    - — 0.6 MHz to Q.

Fig Television standards used in different countries

HDTV (High Definition Television)

High-Definition television (HDTV) means broadcast of television signals with a higher resolution than traditional formats (NTSC, SECAM, PAL) allow. Except for early analog formats in Europe and Japan, HDTV is broadcasted digitally, and therefore its introduction sometimes coincides with the introduction of digital television (DTV).

- ✓ Modern plasma television uses this
- ✓ It consists of 720-1080 lines and higher number of pixels (as many as 1920 pixels).
- ✓ Having a choice in between progressive and interlaced is one advantage of HDTV. Many people have their preferences

HDTV vs Existing Signals (NTSC, PAL, or SECAM)

The HDTV signal is digital resulting in crystal clear, noise-free pictures and CD quality sound. It has many viewer benefits like choosing between interlaced or progressive scanning.

File Formats

File formats in the PC platform are indicated by the 3 letter filename extension.
.mov= QuickTime Movie Format

.avi= Windows movie format
.mpg =MPEG file format

Four Factors of Digital Video

With digital video, four factors have to be kept in mind. These are :

- Frame rate
- Spatial Resolution
- Color Resolution
- Image Quality

Frame Rate

The standard for displaying any type of non-film video is 30 frames per second (film is 24 frames per second). This means that the video is made up of 30 (or 24) pictures or *frames* for every second of video. Additionally these frames are split in half (odd lines and even lines), to form what are called *fields*.

Color Resolution

This second factor is a bit more complex. Color resolution refers to the number of colors displayed on the screen at one time. Computers deal with color in an *RGB* (red-green-blue) format, while video uses a variety of formats. One of the most common video formats is called *YUV*. Although there is no direct correlation between RGB and YUV, they are similar in that they both have varying levels of color depth (maximum number of colours).

Spatial Resolution

The third factor is spatial resolution - or in other words, *"How big is the picture?"*. Since PC and Macintosh computers generally have resolutions in excess of 640 by 480, most people assume that this resolution is the video standard.

A standard analogue video signal displays a full, over scanned image without the borders common to computer screens. The National Television Standards Committee (NTSC) standard used in North America and Japanese Television uses a 768 by 484 display. The Phase Alternative system (PAL) standard for European television is slightly larger at 768 by 576. Most countries endorse one or the other, but never both.

Since the resolution between analogue video and computers is different, conversion of analogue video to digital video at times must take this into account. This can often the result in the *down-sizing* of the video and the loss of some resolution.

4.4 Image Quality

The last, and most important factor is video quality. The final objective is video that looks acceptable for your application. For some this may be 1/4 screen, 15 frames per second (fps), at 8 bits per pixel. Other require a full screen (768 by 484), full frame rate video, at 24 bits per pixel (16.7 million colours).

Digital video basics

Analog composite signals, such as PAL, NTSC and SECAM, are subject to cumulative distortions and noise that affect the quality of the reproduced picture. Separate distortions of the luminance and chrominance components, as well as intermodulation between them, are likely to occur.

The cumulative analog video signal impairments and their effect on the reproduced picture can be reduced considerably by using a digital representation of the video signal and effecting the distribution, processing and recording in the digital domain. By a proper selection of two parameters, namely the sampling frequency and the quantizing accuracy, these impairments can be reduced to low, visually imperceptible values. As long as the digitized signals are distributed, processed and recorded in the digital domain, these impairments are limited.

Sampling

The sampling of the video signal is essentially a pulse amplitude modulation process. It consists of checking the signal amplitude at periodic intervals (T). The sampling frequency (F_S=1/T) has to meet two requirements:

- ✓ It has to be higher than twice the maximum baseband frequency of the analog video signal (F_B), as stipulated by Nyquist. This is required in order to avoid aliasing. Aliasing is visible as spurious picture elements associated with fine details (high frequencies) in the picture. The only way to avoid aliasing is to use an anti-aliasing filter ahead of the A/D converter. The task of this filter is to reduce the bandwidth of the sampled base band.
- ✓ It has to be coherent with and related to an easily identifiable and constant video frequency.

An early approach, $3F_{SC}$, sampled the composite video signal at three times the color subcarrier frequency. This resulted in F_S = 3 × 3.58MHz = 10.7MHz in NTSC and F_S = 3 × 4.43MHz = 13.29MHz in PAL. A later approach, $4F_{SC}$, sampled the composite video signal at four times the color subcarrier frequency, or 17.7MHz in PAL and 14.3MHz in NTSC.

While sampling at a multiple of F_{SC} works well in PAL and NTSC, it doesn't work at all in SECAM. This is due to the inherent nature of SECAM, which uses two separate line-sequential frequency-modulated color subcarriers carrying, respectively, the D_B and D_R color-difference signals.

It appeared evident in the 1970s that a digital video system in which the luminance and chrominance are individually coded would ease the program interchange between the PAL and SECAM countries. This resulted in the component digital concept, which is at the core of all contemporary digital video systems.

Quantizing

The pulse amplitude modulation results in a sequence of pulses, spaced at T=1/F_S intervals, whose amplitude is proportional to the amplitude of the sampled analog signal at the sampling instant. There are an infinite number of shades of gray — ranging from black (lowest video signal amplitude) to white (highest video signal amplitude) — that the analog video signal can represent.

The instantaneous sampling pulse amplitudes can be represented in the digital domain by only a limited number of binary values, resulting in quantizing errors. The possible number of shades of gray is equal to 2^n, where n is the number of bits per sample. Experiments have shown that when less than eight bits per sample are used, the quantizing errors appear as contouring. With eight bits per sample or more, the quantizing errors appear, in general, as random noise (quantizing noise) in the picture. In practical applications, in order to avoid clipping, the signal occupies less than 2^n steps, resulting in a specified quantizing range.

Advantages and disadvantages

- ✓ The advantages of digital video are:
- ✓ Single-pass, analog-type impairments are non-cumulative if the signal stays digital.
- ✓ There is a reduced sensitivity to noise and interference.
- ✓ Digital equipment efficiently and economically performs tasks that are difficult or impossible to perform using analog technology.
- ✓ It is amenable to the application of techniques for efficient retention of essential information such as compression.

The disadvantages of digital video are:
- ✓ Analog-type of distortions, as well unique digital distortions related to sampling and quantizing, result in a variety of visible impairments.
- ✓ Wide bandwidth requirements for recording, distribution and transmission necessitate sophisticated bit-rate reduction and compression schemes to achieve manageable bandwidths.
- ✓ Unlike analog signals, the digital signals do not degrade gracefully and are subjected to a cliff effect.

Chapter 6
Basics of Digital Audio

Digitizing Sound
- ✓ Microphone produces analog signal
- ✓ Computer deals with digital signal

Sampling Audio

Analog Audio
Most natural phenomena around us are continuous; they are continuous transitions between two different states. Sound is not exception to this rule i.e. sound also constantly varies. Continuously varying signals are represented by analog signal.

Signal is a continuous function f in the time domain. For value *y=f(t)*, the argument *t* of the function *f* represents time. If we graph *f,* it is called wave. (see the following diagram)

Fig 1 analog signal

A wave has three characteristics:

- ✓ Amplitude
- ✓ Frequency, and
- ✓ Phase

Amplitude: is the intensity of signal. This is can be determined by looking at the height of signal. If amplitude increases, the sound becomes louder. Amplitude measures the how high or low the voltage of the signal is at a given point of time.
Frequency: is the number of times the wave cycle is repeated. This can be determined by counting the number of cycles in given time interval. Frequency is related with pitchness of the sound. Increased frequency→high pitch.
Phase: related to the wave's appearance.

Fig 2 recording sound and the need for digitization

When sound is recorded using microphone, the microphone changes the sound into analog representation of the sound. In computer, we can't deal with analog things. This makes it necessary to change analog audio into digital audio. How? Read the next topic.

Analog to Digital Conversion

Converting an analog audio to digital audio requires that the analog signal is *sampled*. Sampling is the process of taking periodic measurements of the continuous signal. Samples are taken at regular time interval, i.e. every T seconds. This is called *sampling frequency/sampling rate*. Digitized audio is sampled audio. Many times each second, the analog signal is sampled. How often these samples are taken is referred to as *sampling rate*. The amount of information stored about each sample is referred to as *sample size*.

Analog signal is represented by *amplitude* and *frequency*. Converting these waves to digital information is referred to as digitizing**.** The challenge is to convert the analog waves to numbers (digital information).

In digital form, the measure of amplitude (the 7 point scale - vertically) is represented with binary numbers (bottom of graph). The more numbers on the scale the better the quality of the sample, but more bits will be needed to represent that sample. The graph below only shows 3-bits being used for each sample, but in reality either 8 or 16-bits will be used to create all the levels of amplitude on a scale. (Music CDs use 16-bits for each sample).

Fig 3 quantization of samples

In digital form, the measure of *frequency* is referred to as how often the sample is taken. In the graph below the sample has been taken 7 times (reading across). Frequency is talked about in terms of Kilohertz (KHz).

Hertz (Hz) = number of cycles per second
KHz = 1000Hz
MHz = 1000 KHz

Music CDs use a frequency of 44.1 KHz. A frequency of 22 KHz for example, would mean that the sample was taken less often.

Sampling means measuring the value of the signal at a given time period. The samples are then quantized. **Quantization** is rounding the value of each sample to the nearest amplitude number in the graph. For example, if amplitude of a specific sample is 5.6, this should be rounded either up to 6 or down to 5. This is called quantization. Quantization is assigning a value (from a set) to a sample. The quantized values are changed to binary pattern. The binary patterns are stored in computer.

Fig 4 digitization process (sampling, quantization, and coding)

Fig 5 Sampling and quantization

Example:
The sampling points in the above diagram are A, B, C, D, E, F, H, and I.
The value of sample at point A falls between 2 and 3, may be 2.6. This value should be represented by the nearest number. We will round the sample value to 3. Then this three is converted into binary and stored inside computer.

Similarly, the values of other sampling points are:
B=1
C=3
D=1
E=3
F=1
G=2
H=3
I=1
The values of most sample points are quantized. After quantization, we convert sample values into binary digits.

Sample Rate

A sample is a single measurement of amplitude. The sample rate is the number of these measurements taken every second. In order to accurately represent all of the frequencies in a recording that fall within the range of human perception, generally accepted as

20Hz–20KHz, we must choose a sample rate high enough to represent all of these frequencies. At first consideration, one might choose a sample rate of 20 KHz since this is identical to the highest frequency. This will not work, however, because every cycle of a waveform has both a positive and negative amplitude and it is the rate of alternation between positive and negative amplitudes that determines frequency. Therefore, we need at least two samples for every cycle resulting in a sample rate of at least 40 KHz.

Sampling Theorem

Sampling frequency/rate is very important in order to accurately reproduce a digital version of an analog waveform.

Nyquist's Theorem:

The Sampling frequency for a signal must *be* at least twice the highest frequency component in the signal.

Sample rate = 2 x highest frequency

Fig 5 Sampling at signal frequency and at twice Nyquist frequency

When the sampling rate is lower than or equal to the Nyquist rate, the condition is defined as under sampling. It is impossible to rebuild the original signal according to the sampling theorem when such sampling rate is used.

Aliasing

What exactly happens to frequencies that lie above the Nyquist frequency? First, we'll look at a frequency that was sampled accurately:

In this case, there are more than two samples for every cycle, and the measurement is a good approximation of the original wave. we will get back the same signal we put in later on when converting it into analog.

Remember: speakers can play only analog sound. You have to convert back digital audio to analog when you play it.

If we undersample the signal, though, we will get a very different result:

In this diagram, the blue wave (the one with short cycles) is the original frequency. The red wave (the one with lower frequency) is the aliased frequency produced from an insufficient number of samples. This frequency, which was in all likelihood a high partial in a complex timbre, has folded over and is now below the Nyquist frequency. For example, a 11KHz frequency sampled at 18KHz would produce an alias frequency of 7KHz. This will alter the timbre of the recording in an unacceptable way.

Under sampling causes frequency components that are higher than half of the sampling frequency to overlap with the lower frequency components. As a result, the higher frequency components roll into the reconstructed signal and cause distortion of the signal. This type of signal distortion is called aliasing.

Common Sampling Rates

- 8KHz: used for telephone
- 11.025 KHz: Speech audio
- 22.05 KHz: Low Grade Audio (WWW Audio, AM Radio)
- 44.1 KHz: CD Quality audio

Sample Resolution/Sample Size

Each sample can only be measured to a certain degree of accuracy. The accuracy is dependent on the number of bits used to represent the amplitude, which is also known as the sample resolution.

How do we store each sample value (*quantized value*)?

- ***8 Bit Value** (0-255)*
- ***16 Bit Value*** (Integer) (0-65535)

The amount of memory required to store t seconds long sample is as follows:

- If we use 8 bit resolution, mono recording
  memory = f*t*8*1
- If we use 8 bit resolution, stereo recording
  memory = f*t*8*2

- If we use 16 bit resolution, and mono recording
  memory = f*t*16*1
- If we use 16 bit resolution, and stereo recording
  memory =f* t*16*2

where f is sampling frequency, and
t is time duration in seconds

<u>Examples:</u>
Abebe sampled audio for 10 seconds. How much storage space is required if
a) 22.05 KHz sampling rate is used, and 8 bit resolution with mono recording?
b) 44.1 KHz sampling rate is used, and 8 bit resolution with mono recording?
c) 44.1 KHz sampling rate is used, 16 bit resolution with stereo recording?
d) 11.025 KHz sampling rate, 16 bit resolution with stereo recording?

<u>Solution:</u>
a) m=22050*8*10*1
m= 1764000bits=220500bytes=220.5KB
b) m=44100*8*10*1

m= 3528000 bits=441000butes=441KB
c) m=44100*16*10*2
m= 14112000 bits= 1764000 bytes= 1764KB
d) m=11025*16*10*2
m= 3528000 bits= 441000 bytes= 441KB

Implications of Sample Rate and Bit Size

- ✓ Affects Quality of Audio
- ✓ Affects Size of Data

File Type	44.1 KHz	22.05 KHz	11.025 KHz
16 Bit Stereo	10.1 Mb	5.05 Mb	2.52 Mb
16 Bit Mono	5.05 Mb	2.52 Mb	1.26 Mb
8 Bit Mono	2.52 Mb	1.26 Mb	630 Kb

Table Memory required for 1 minute of digital audio

Clipping

Both analog and digital media have an upper limit beyond which they can no longer accurately represent amplitude. Analog clipping varies in quality depending on the medium. The upper amplitudes are being altered, distorting the waveform and changing the timbre, but the alterations are slightly different. Digital clipping, in contrast, is always the same. Once an amplitude of 1111111111111111 (the maximum value in a 16 bit resolution) is reached, no higher amplitudes can be represented. The result is not the smooth, rounded flattening of analog clipping, but a harsh slicing of off the top of the waveform, and an unpleasant timbral result.

An Ideal Recording

We should all strive for an ideal recording. First, don't ignore the analog stage of the process. Use a good microphone, careful microphone placement, high quality cables, and a reliable analog-to-digital converter. Strive for a hot (high levels), clean signal.

Second, when you sample, try to get the maximum signal level as close to zero as possible without clipping. That way you maximize the inherent signal-to-noise ratio of the medium. Third, avoid conversions to analog and back if possible. You may need to convert the signal to run it through an analog mixer or through the analog inputs of a digital effects processor. Each time you do this, though, you add the noise in the analog signal to the subsequent digital re-conversion.

Chapter 7
Data Compression

Introduction

Data compression is often referred to as coding, where coding is a very general term encompassing any special representation of data which satisfies a given need.

Definition: Data compression is the process of encoding information using fewer number of bits so that it takes less memory area (storage) or bandwidth during transmission.
Two types of compression:

- ✓ Lossy data compression
- ✓ Lossless data compression

Lossless Data Compression: in lossless data compression, the original content of the data is not lost/changed when it is compressed (encoded).

Examples:
RLE (Run Length Encoding)
Dictionary Based Coding
Arithmetic Coding

Lossy data compression: the original content of the data is lost to certain degree when compressed. Part of the data that is not much important is discarded/lost. The *loss* factor determines whether there is a loss of quality between the original image and the image after it has been compressed and played back (decompressed). The more compression, the more likely that quality will be affected. Even if the quality difference is not noticeable, these are considered *lossy* compression methods.

Examples
JPEG (Joint Photographic Experts Group)
MPEG (Moving Pictures Expert Group)
ADPCM

Information Theory

Information theory is defined to be the study of efficient coding and its consequences. It is the field of study concerned about the storage and transmission of data. It is concerned with source coding and channel coding.
Source coding: involves compression
Channel coding: how to transmit data, how to overcame noise, etc

Data compression may be viewed as a branch of information theory in which the primary objective is to minimize the amount of data to be transmitted.

Fig Information coding and transmission

Need for Compression

With more colors, higher resolution, and faster frame rates, you produce better quality video, but you need more computer power and more storage space for your video. Doing some simple calculations (see below) it can be shown that with 24-bit color video, with 640 by 480 resolutions, at 30 fps, requires an astonishing 26 megabytes of data per second! Not only does this surpass the capabilities of the many home computer systems, but also overburdens existing storage systems.

640 horizontal resolution

X 480 vertical resolution

= 307, 200 total pixels per frame

X 3 bytes per pixel

= 921, 600 total bytes per frame

X 30 frames per second

= 27, 648, 000 total bytes per second

/ 1, 048 576 to convert to megabytes

= 26.36 megabytes per second!

The calculation shows space required for video is excessive. For video, the way to reduce this amount of data down to a manageable level is to compromise on the quality of video to some extent. This is done by lossy compression which forgets some of the original data.

Compression Algorithms

Compression methods use mathematical algorithms to reduce (or compress) data by eliminating, grouping and/or averaging similar data found in the signal. Different Although there are various compression methods, including Motion JPEG, only MPEG-1 and MPEG-2 are internationally recognized standards for the compression of moving pictures (video).

A simple characterization of data compression is that it involves transforming a string of characters in some representation (such as ASCII) into a new string (of bits, for example) which contains the same information but whose length is as small as possible. Data compression has important application in the areas of data transmission and data storage.

The proliferation of computer communication networks is resulting in massive transfer of data over communication links. Compressing data to be stored or transmitted reduces storage and/or communication costs. When the amount of data to be transmitted is reduced, the effect is that of increasing the capacity of the communication channel.

Lossless compression is a method of reducing the size of computer files without losing any information. That means when you compress a file, it will take up less space, but when you decompress it, it will still have the exact same information. The idea is to get rid of any redundancy in the information, this is exactly what happens is used in ZIP and GIF files. This differs from lossy compression, such as in JPEG files, which loses some information that isn't very noticeable. Why use lossless compression?

You can use lossless compression whenever space is a concern, but the information must be the same. An example is when sending text files over a modem or the Internet. If the files are smaller, they will get there faster. However, they must be the same as that you sent at destination. Modem uses LZW compression automatically to speed up transfers.

There are several popular algorithms for lossless compression. There are also variations of most of them, and each has many implementations. Here is a list of the families, their variations, and the file types where they are implemented:

Variable Length Encoding
Claude Shannon and R.M. Fano created the first compression algorithm in the 1950's. This algorithm assigns variable number of bits to letters/symbols.

Shannon-Fano Coding
Let us assume the source alphabet S={X1,X2,X3,…,Xn} and
Associated probability P={P1,P2,P3,…,Pn}
The steps to encode data using Shannon-Fano coding algorithm is as follows:
Order the source letter into a sequence according to the probability of occurrence in non-increasing order i.e. decreasing order.

ShannonFano(sequence s)
If s has two letters
  Attach 0 to the codeword of one letter and 1 to the codeword of another;
Else if s has more than two letter
  Divide s into two subsequences S1, and S2 with the minimal difference between probabilities of each subsequence;
  extend the codeword for each letter in S1 by attaching 0, and by attaching 1 to each codeword for letters in S2;
ShannonFano(S1);
ShannonFano(S2);

Example: Suppose the following source and with related probabilities
S={A,B,C,D,E}
P={0.35,0.17,0.17,0.16,0.15}

Message to be encoded="ABCDE"

The probability is already arranged in non-increasing order. First we divide the message into AB and CDE. Why? This gives the smallest difference between the total probabilities of the two groups.
S1={A,B}   P={0.35,0.17}=0.52
S2={C,D,E} P={0.17,0.17,0.16}=0.46
The difference is only 0.52-0.46=0.06. This is the smallest possible difference when we divide the message.
Attach 0 to S1 and 1 to S2.
Subdivide S1 into sub groups.
S11={A} attach 0 to this
S12={B} attach 1 to this

Again subdivide S2 into subgroups considering the probability again.
S21={C} P={0.17}=0.17
S22={D,E} P={0.16,0.15}=0.31
Attach 0 to S21 and 1 to S22. Since S22 has more than one letter in it, we have to subdivide it.
S221={D} attach 0
S222={E} attach 1

Fig Shannon-Fano coding tree

The message is transmitted using the following code (by traversing the tree)
A=00   B=01
C=10   D=110
E=111

Instead of transmitting ABCDE, we transmit 000110110111.

Dictionary Encoding

Dictionary coding uses groups of symbols, words, and phrases with corresponding abbreviation. It transmits the index of the symbol/word instead of the word itself. There are different variations of dictionary based coding:
LZ77 (printed in 1977)
LZ78 (printed in 1978)
LZSS
LZW (Lempel-Ziv-Welch)

LZW Compression
LZW compression has its roots in the work of Jacob Ziv and Abraham Lempel. In 1977, they published a paper on "sliding-window" compression, and followed it with another paper in 1978 on "dictionary" based compression. These algorithms were named LZ77

and LZ78, respectively. Then in 1984, Terry Welch made a modification to LZ78 which became very popular and was called LZW.

The Concept
Many files, especially text files, have certain strings that repeat very often, for example " the ". With the spaces, the string takes 5 bytes, or 40 bits to encode. But what if we were to add the whole string to the list of characters? Then every time we came across " the ", we could send the code instead of 32,116,104,101,32. This would take less no of bits.

This is exactly the approach that LZW compression takes. It starts with a dictionary of all the single character with indexes 0-255. It then starts to expand the dictionary as information gets sent through. Then, redundant strings will be coded, and compression has occurred.

The Algorithm:

```
LZWEncoding()
Enter all letters to the dictionary;
Initialize string s to the first letter from the input;
While any input is left
  read symbol c;
  if s+c exists in the dictionary
      s = s+c;
  else
      output codeword(s); //codeword for s
      enter s+c to dictionary;
      s =c;
end loop
output codeword(s);
```

Example: encode the ff string "*aababacbaacbaadaa*"

The program reads one character at a time. If the code is in the dictionary, then it adds the character to the current work string, and waits for the next one. This occurs on the first character as well. If the work string is not in the dictionary, (such as when the second character comes across), it adds the work string to the dictionary and sends over the wire the works string without the new character. It then sets the work string to the new character.

Example:
Encode the message *aababacbaacbaadaaa* using the above algorithm

Encoding
Create dictionary of letters found in the message

Encoder		**Dictionary**	
Input	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d

S is initialized to the first letter of message *a (s=a)*
Read symbol to c, and the next symbol is *a (c=a)*
Check if *s+c (s+c=aa)* is found in the dictionary (the one created above in step 1). It is not found. So add s+c(s+c=aa) to dictionary and output codeword for s(s=a). The code for a is 1 from the dictionary.
Then initialize s to c (s=c=a).

Encoder		**Dictionary**	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
aa	1	5	aa

Read the next letter from message to *c (c=b)*
Check if s+c (ab) is found in the dictionary. It is not found. Then, add s+c (s+c=ab) into dictionary and output code for c (c=b). The codeword is 2. Then initialize s to c (s=c=b).

Encoder		Dictionary	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
aa	1	5	aa
ab	1	6	ab

Read the next letter to c (c=a).
Check if s+c (s+c=ba) is found in the dictionary. It is not found. Then add s+c (s+c=ba) to the dictionary. Then output the codeword for s (s=b). It is 2. Then initialize s to c (s=c=b).

Encoder		Dictionary	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
aa	1	5	aa
ab	1	6	ab
ba	2	7	ba

Read the next message to c (c=a). Then check if s+c (s+c=ab) is found in the dictionary. It is there. Then initialize s to s+c (s=s+c=ab).

Read again the next letter to c (c=a). Then check if s+c (s+c=aba) is found in the dicitionary. It is not there. Then transmit codeword for s (s=ab). The code is 6. Initialize s to c(s=c=a).

-

Encoder		**Dictionary**	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
aa	1	5	aa
ab	1	6	ab
ba	2	7	ba
aba	6	8	aba

Again read the next letter to c and continue the same way till the end of message. At last you will have the following encoding table.

Encoder		**Dictionary**	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
aa	1	5	aa
ab	1	6	ab
ba	2	7	ba
aba	6	8	aba
ac	1	9	ac
cb	3	10	cb
baa	7	11	baa
acb	9	12	acb
baad	11	13	baad
da	4	14	da
aaa	5	15	aa

Table encoding string

Now instead of the original message, you transmit their indexes in the dictionary. The code for the message is *112613791145.*

Decompression

The algorithm:

LZWDecoding()
Enter all the source letters into the dictionary;
Read priorCodeword and output one symbol corresponding to it;
While codeword is still left
read Codeword;
PriorString = string (PriorCodeword);
If codeword is in the dictionary
Enter in dictionary PriorString + firstsymbol(string(codeword));
output string(codeword);
else
Enter in the dictionary priorString +firstsymbol(priorString);
Output priorString+firstsymbol(priorstring);
priorCodeword=codeword;
end loop

The nice thing is that the decompressor builds its own dictionary on its side, that matches exactly the compressor's dictionary, so that only the codes need to be sent.

Example:
Let us decode the message *112613791145.*
We will start with the following table.

Encoder		**Dictionary**	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d

Read the first code. It is 1. Output the corresponding letter→a

Encoder		**Dictionary**	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
1	a		

Read the next code. It is 1 and it is found in the dictionary. So add aa to the dictionary and output a again.

Encoder		**Dictionary**	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
1	a		
1	a	5	aa

Read the next code which is 2. It is found in the dictionary. We add ab to dictionary and output b.

Encoder		**Dictionary**	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
1	a		
1	a	5	aa
2	b	6	ab

Read the next code which is 6. It is found in the dictionary. Add ba to dictionary and output ab

Encoder		Dictionary	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
1	a		
1	a	5	aa
2	b	6	ab
6	ab	7	ba

Read the next code. It is 1. 1 is found in the dictionary. Add aba to the dictionary and output a.

Encoder		Dictionary	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
1	a		
1	a	5	aa
2	b	6	ab
6	ab	7	ba
1	a	8	aba

Read the next code. It is 3 and it is found in the dictionary. Add ac to dictionary and output c.

Continue like this till the end of code is reached. You will get the following table:

Encoder		Dictionary	
Input(s+c)	Output	Index	Entry
		1	a
		2	b
		3	c
		4	d
1	a		
1	a	5	aa
2	b	6	ab
6	ab	7	ba
1	a	8	aba
3	c	9	ac
7	ba	10	cb
9	ac	11	baa
11	baa	12	acb
4	d	13	baad
5	aa	14	da

The decoded message is aababacbaacbaadaa

Huffman Compression

When we encode characters in computers, we assign each an 8-bit code based on an ASCII chart. But in most files, some characters appear more often than others. So wouldn't it make more sense to assign shorter codes for characters that appear more often and longer codes for characters that appear less often? D.A. Huffman published a paper in 1952 that improved the algorithm slightly and it soon superceded Shannon-Fano coding with the appropriately named Huffman coding.

Huffman coding has the following properties:
- ✓ Codes for more probable characters are shorter than ones for less probable characters.
- ✓ Each code can be uniquely decoded

To accomplish this, Huffman coding creates what is called a *Huffman tree*, which is a binary tree.

First count the amount of times each character appears, and assign this as a *weight/ probability* to each character, or node. Add all the nodes to a list.
Then, repeat these steps until there is only one node left:

- ✓ Find the two nodes with the lowest weights.
- ✓ Create a parent node for these two nodes. Give this parent node a weight of the sum of the two nodes.
- ✓ Remove the two nodes from the list, and add the parent node.

This way, the nodes with the highest weight will be near the top of the tree, and have shorter codes.

<u>Algorithm to create the tree</u>
Assume the source alphabet S={X1, X2, X3, …,Xn} and
Associated Probabilities P={P1, P2, P3,…, Pn}

Huffman()
For each letter create a tree with single root node and order all trees according to the probability of letter of occurrence;
while more than one tree is left
take two trees t1, and t2 with the lowest probabilities p1, p2 and create a tree with probability in its root equal to p1+p2 and with t1 and t2 as its subtrees;
associate 0 with each left branch and 1 with each right branch;
create unique codeword for each letter by traversing the tree the root to the leaf containing the probability corresponding to this letter and putting all encountered 0s and 1s together;

Example: Suppose the following source and related probability
S={A,B,C,D,E}
P={0.15,0.16,0.17,0.17,0.35}
Message="abcde"

Fig Huffman tree

To read the codes from a Huffman tree, start from the root and add a *0* every time you go left to a child, and add a *1* every time you go right. So in this example, the code for the character *b* is 01 and the code for *d* is 110.

As you can see, *a* has a shorter code than *d*. Notice that since all the characters are at the leafs of the tree, there is never a chance that one code will be the prefix of another one (eg. *a* is *01* and *b* is *011*). Hence, this unique prefix property assures that each code can be uniquely decoded.

The code for each letter is:
a=000 b=001
c=010 d=011
e=1

The original message will be encoded to:
abcde=0000010100111

To decode the message coded by Huffman coding, a conversion table had to be known by the receiver. Using this table, a tree can be constructed with the same path as the tree used for coding. Leaves store the same path as the tree used for coding. Leaves store letters instead of probabilities for efficiency purpose.

The decoder then can use the Huffman tree to decode the string by following the paths according to the string and adding a character every time it comes to one.

Fig Huffman tree

<u>*The Algorithm*</u>

Move left if you get 0
Move right if you get 1
If you get letter (reach leaf node) output that letter.
Go back and start from root again with the remaining code.

Using this algotihm and the above decoding tree, let us decode the encoded message 0000010100111 at destination.
0-move left
0-move left again
0-move left again, and we have reached leaf. Output the letter on the leaf node which is a.

Go back to root.
0-move left
0-move left
1-move right, and we have reached the leaf. Output letter on the leaf and it is b.

Go back to root.
0-move left
1-move right
0-move left, and we reach leaf. Output letter found on the leaf which is c.

Go back to root.
0-move left
1-move right
1-move right, and we reach leaf. Output letter on leaf which is d.

Go back to root.
1-move right, and we reach leaf node. Output the letter on the node which is e. Now we have finished i.e. no more code remains. Display the letters output as message. *Abcde*

How can the encoder let the decoder know which particular coding tree has been used? Two ways:

i) Both agree on particular Hufmann tree and both use it for sending any message
ii) The encoder constructs Huffman tree afresh every time a new message is sent and sends the conversion table along with the message. This is more versatile, but has additional overload—sending conversion table. But for large data, there is the advantage.

It is also possible to create tree for pairs of letters. This improves performance.

Example:
S={x, y, z}
P={0.1, 0.2, 0.7}
To get the probability of pairs, multiply the probability of each letter.
xx=0.1*0.1=0.01
xy=0.1*0.2=0.02
xz=0.1*0.3=0.07
yx=0.2*0.1=0.02
yy=0.2*0.2=0.04
yz=0.2*0.7=0.14
zx=0.7*0.1=0.07
zz=0.7*0.7=0.49
zy=0.7*0.2=0.14
Using these probabilities, you can create Huffman tree of pairs the same way as we did previously.

Arithmetic Coding

The entire data set is represented by a single rational number, whose value is between 0 and 1. This range is divided into sub-intervals each representing a certain symbol. The number of sub-intervals is identical to the number of symbols in the current set of symbols and the size is proportional to their probability of appearance. For each symbol in the original data a new interval division takes place, on the basis of the last sub-interval.

Algorithm:

```
ArithmeticEncoding(message)
    CurrentInterval=[0,1);  //includes 0 but not 1
    while the end of message is not reached
           read letter Xi from message;
           divide the CurrentInterval into SubInterval IR_cureentInterval;
           CurrentInterval=SubInterval_i in CurrentInterval;
    Output bits uniquely identifying CurrentInterval;
```

Assume the source alphabet s={X1, X2, X3,…, Xn} and associated probability of P={p1, p2, p3,…, pn}

To calculate sub interval of current interval [L,R], use the following formula

$IR_{[L,R]}$={[L, L+(R-L)*P1],[L+(R-L)*P1, L+(R-L)*P2],[L+(R-L)*P2, L+(R-L)*P3],…,
[L+(R-L)*P_{n-1}, L+(R-L)*P1)}

where P_i= , and
[L,R]=current interval for which sub interval is calculated

Cumulative probabilities are indicated using capital P and single probabilities are indicated using small p.

Example:
Encode the message *abbc#* using arithmetic encoding.
s={a,b,c,#}
p={0.4,0.3,0.1,0.2}
At the beginning CurentInterval is set to [0,1). Let us calculate subintervals of [0,1).

First let us get cumulative probability P_i
P1=0.4
P2=0.4+0.3=0.7
P3=0.4+0.3+0.1=0.8
P4=0.4+0.3+0.1+0.2=1

Next calculate subintervals of [0,1) using the formula given above.
IR[0,1]={[0,0+(1-0)*0.4),[0+(1-0)*0.4, 0+(1-0)*0.7), [0+(1-0)*0.7, 0+(1-0)*0.8),
[0+(1-0)*0.8, 0+(1-0)*1)}
IR[0,1]={[0,0.4),[0.4,0.7),[0.7,0.8),[0.8,1)}-- four subintervals

Now the question is, which one of the SubIntervals will be the CurrentInterval? To determine this, read the first letter of the message. It is a. Look where a is found in the source alphabet. It is found at the beginning. So the next CurrentInterval will be [0,4) which is also found at the beginning in the SubIntervals.

Again let us calculate the SubIntervals of CurrentInterval [0,0.4). The cumulative probability does not change i.e the same as previous.
IR[0,0.4]={[0,0+(0.4-0)*0.4),[0+(0.4-0)*0.4, 0+(0.4-0)*0.7),[0+(0.4-0)*0.7, 0+(0.4-0) *0.8),
[0+(0.4-0)*0.8, 0+(0.4-0)*1)}
IR[0,0.4]={[0,0.16),[0.16,0.28),[0.28,0.32),[0.32,0.4)}.
Which interval will be the next CurrentInterval? Read the next letter from message. It is b. B is found in the second place in the source alphabet list. The next CurrentInterval will be the second SubInterval i.e [0.16,0.28).

Continue like this till there is letter left in the message. You will get the following result:
IR[0.16,0.28]={[0.16,0.208),[0.208,0.244),[0.244,0.256),[0.256,0.28)}. Next
IR[0.208,0.244]={[0.208,0.2224),[0.2224,0.2332),[0.2332,0.2368),[0.2368,0.242). Next
IR[0.2332,0.2368]={[0.2332,0.23464),[0.23464,0.23572),[0.23572,0.23608),[0.23608, 0.2368)}.

We are done because no more letter remained in the message. The last letter read was #. It is the fourth letter in source alphabet. So take the fourth SubInterval as CurrentInterval i.e [0.23608, 0.2368]. Now any number between the last CurrentInterval is sent as the message. So you can send 0.23608 as the encoded message or any number between 0.23608, and 0.2368.

Diagramatically, calculating SubIntervals look like this:

Fig sub interval and current interval

Decoding

Algorithm:

ArithmeticDecoding(codeword)
CurrentInterval=[0,1];
While (1)
Divide CurrentInterval into SubIntervals $IR_{currentInterval}$;
Determine the $SubInterval_i$ of CurrentInterval to which the codeword belongs;
Output letter X_i corresponding to this SubInterval;
If end of file
Return;
CurrentInterval=$SubInterval_i$ in $IR_{currentInterval}$;
End of while

Example:
Decode 0.23608 which we previously encoded.
To decode the source alphabet and related probability should be known by destination.
Let us use the above source and probability.
s={a,b,c,#}
p={0.4,0.3,0.1,0.2}

First set CurrentInterval to [0,1], and then calculate SubInterval for it. The formula to calculate the SubInterval is the same to encoding. The cumulative probabilities are:
P1=0.4
P2=0.4+0.3=0.7
P3=0.4+0.3+0.1=0.8
P4=0.4+0.3+0.1+0.2=1

IR[0,1]={[0,0+[1-0]*0.4),[0+[1-0]*0.4, 0+[1-0]*0.7),[0+[1-0]*0.7, 0+[1-0]*0.8), [0+[1-0]*0.8, 0+[1-0]*1)}
IR[0,1]={[0,0.4),[0.4,0.7),[0.7,0.8),[0.8,1)}. Now check in which SubInterval the encode message falls. It falls in the first SubInterval i.e [0,0.4]. Output the first letter from source alphabet. It is a. Set CurrentInterval to [0,0.4]

IR[0,0.4]={[0,0+(0.4-0)*0.4),[0+(0.4-0)*0.4, 0+(0.4-0)*0.7),[0+(0.4-0)*0.7, 0+(0.4-0)*0.8),
[0+(0.4-0)*0.8, 0+(0.4-0)*1)}
IR[0,0.4]={[0,0.16),[0.16,0.28),[0.28,0.32),[0.32,0.4)}. Again check where 0.23608 falls. It falls in the second SubInterval i.e [0.16,0.28]. Set CurrentInterval to this SubInterval. Output the second letter from source alphabet. It is b.

IR[0.16,0.28]={[0.16,0.208),[0.208,0.244),[0.244,0.256),[0.256,0.28)}. 0.23608 falls in the second SubInterval. Output the second letter from source alphabet. It is b.

IR[0.208,0.244]={[0.208,0.2224),[0.2224,0.2332),[0.2332,0.2368),[0.2368,0.242). falls in the third SubInterval. Output the third letter from source alphabet. It is c.

IR[0.2332,0.2368]={[0.2332,0.23464),[0.23464,0.23572),[0.23572,0.23608),[0.23608, 0.2368)}. 0.23608 falls in the fourth SubInterval. Output fourth letter which is #. Now end of the message has been reached.

Disadvantage: arithmetic precision of computer is soon suppressed and hence large message can't be encoded.

Implementation of Arithmetic Coding
To solve the above disadvantage, arithmetic coding is implemented as follows:

Algorithm:

```
OutputBits()
{
    While(1)
        If CurrentInterval ⊂ [0,0.5)
            Output 0 and bitcount 1s; //and here shows concatenation
            Bitcount=0;
        Else if CurrentInterval ⊂ [0.5,1)
            Output 1 and bitcount 0s;
            Bitcount=0;
            Subtract 0.5 from left and right bounds of CurrentInterval;
        Else if CurrentInterval ⊂ [0.25,0.75)
            Bitcount++;
            Subtract 0.25 both left and right bounds of CurrentInterval;
        Else
            Break;
        Double left and right bounds of CurreentInterval;
}

FinishArithmeticCoding()
{
```

```
        bitcount++;
        if lowerbound of CurrentInterval <0.25
            output 0 and bitcount 1s;
        else
            output 1 and bitcount 0s;
}

ArithmeticEncoding(message)
{
        CurrentInterval=[0,1];
        Bitcount=0;
        While the end of message is not reached
        {
            read letter X_i from message;
            divide CurrentInterval into SubInterval IR_CurrentInterval;
            CurrentInterval=SubInterval_i in IR_CurrentInterval;
            OutputBits();
        }
        FinishArithmeticEncoding();
}
```

Example:

Encode the message abbc#.

s={a,b,c,#}

p={0.4,0.3,0.1,0.2}

CurrentInterval	input	output	bitcount	Subintervals
[0,1)	a		0	[0, 0.4) [0.4,0.7) [0.7,0.8)

				[0.8, 1)
[0,0.4)		0		
[0,0.8)	b			[0,0.32) [0.32, 0.56) [0.56,0.64) [0.64,0.8)
[0.32,0.56)		-	1	
[0.14,0.62)	b			[0.14, 0.332) [0.332, 0.476) [0.476, 0.524) [0.524,0.62)
[0.332,0.476)		01	0	
[0.664,0.952)		1		
[0.328,0.904)	c			[0.328,0.5584) [0.5

			584, 0.7312) [.7312, .7888) [.7888,.904)
[.7312,.788)	1		
[.4624,.5776)	-	1	
[.4248,.6552)	-	2	
[.3496,.8104)	#		[.3496,.53392) [.53392,.67216) [.67216,.71824) [.71824,.8104)
[.71824,.8104)	100	0	
[.43648,.6208)	-	1	

[.37296, .7416)	-	2
[.24592, .9832)	0111	

The final code will be 001111000111 from the output column of table.