LECTURE NOTES

ON

MULTIMEDIA & RICH INTERNET APPLICATIONS

IV B. Tech II semester (JNTUH-R15)

Ms. Y Harika Devi
Assistant Professor

INFORMATION TECHNOLOGY

INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomus)
DUNDIGAL, HYDERABAD - 500 043

UNIT- I
Fundamental Concepts in Text and Image

What is Multimedia?

When different people mention the term multimedia, they often have quite different, or even opposing, viewpoints.

- **A PC vendor:** a PC that has sound capability, a DVD-ROM drive, and perhaps the superiority of multimedia-enabled microprocessors that understand additional multimedia instructions.
- **A consumer entertainment vendor:** interactive cable TV with hundreds of digital channels available, or a cable TV-like service delivered over a high-speed Internet connection.
- **A Computer Science (CS) student:** applications that use multiple modalities including text, images, drawings (graphics), animation, video, sound including speech, and interactivity.
- **Multimedia and Computer Science:** Graphics, HCI, visualization, computer vision, data compression, graph theory, networking, database systems.

Components of Multimedia:

Multimedia involves multiple modalities of text, audio, images, drawings, animation, and video. Examples of how these modalities are put to use:

- Video teleconferencing.
- Distributed lectures for higher education.
- Tele-medicine.
- Co-operative work environments.
- Searching in (very) large video and image databases for target visual objects.
- Augmented reality: placing real-appearing computer graphics and video objects into scenes.
- Including audio cues for where video-conference participants are located.
- Building searchable features into new video, and enabling very high- to very low-bit-rate use of new, scalable multimedia products.
- Making multimedia components editable.
- Building inverse-Hollywood applications that can recreate the process by which a video was made.
- Using voice-recognition to build an interactive environment, say a kitchen-wall web browser.

Multimedia Research Topics and Projects:

To the computer science researcher, multimedia consists of a wide variety of topics:

1. **Multimedia processing and coding**: multimedia content analysis, content-based multimedia retrieval, multimedia security, audio/image/video processing, compression, etc.
2. **Multimedia system support and networking**: network protocols, Internet, operating systems, servers and clients, quality of service (QoS), and databases.
3. **Multimedia tools, end-systems and applications**: hypermedia systems, user interfaces, authoring systems.
4. **Multi-model interaction and integration**: web-everywhere devices, multimedia education including Computer Supported Collaborative Learning, and design and applications of virtual environments.

Current Multimedia Projects:

Many exciting research projects are currently underway. Here are a few of them:

1. **Camera-based object tracking technology**: tracking of the control objects provides user control of the process.
2. **3D motion capture**: used for multiple actor capture so that multiple real actors in a virtual studio can be used to automatically produce realistic animated models with natural movement.
3. **Multiple views**: allowing photo-realistic (video-quality) synthesis of virtual actors from several cameras or from a single camera under differing lighting.
4. **3D capture technology**: allow synthesis of highly realistic facial animation from speech.
5. **Specific multimedia applications**: aimed at handicapped persons with low vision capability and the elderly a rich of endeavor.
6. **Digital fashion**: aims to develop smart clothing that can communicate with other such enhanced clothing using wireless communication, so as to artificially enhance human interaction in a social setting
7. **Electronic House call system**: an initiative for providing interactive health monitoring services to patients in their homes.
8. **Augmented Interaction applications**: used to develop interfaces between real and virtual humans for tasks such as augmented story telling.

Multimedia and Hypermedia:

History of Multimedia:

Brief history of use of Multimedia:

- **Newspaper:** the first mass communication medium that uses text, graphics, and images
- **Motion Pictures:** conceived of in 1830's in order to observe motion too rapid for reception by the human eye. Thomas Alva Edision invented motion picture camera in 1887
- **Wireless Radio:** 1895, Guglielmo Marconi sent first radio transmission at Pontecchio, Italy
- **Television:** the new medium for the 20th century, established video as a commonly available medium and has since changed the world of mass communications.

The connection between computers and ideas about multimedia covers what is actually only a short period:

1945: Vannevar Bush wrote a landmark article describing hypermedia system called Memex.

1960: Ted Nelson coined the term hypertext.

1967: Nicholas Negroponte formed the Architecture Machine Group.

1968: Douglas Engelbart demonstrated the On-Line System (NLS), very early hypertext program.

1969: Nelson and van Dam at Brown University created an early hypertext editor called FRESS.

1976: MIT Architecture Machine Group proposed a Multiple Media project resulted in Aspen Movie Map

1978: First hypermedia videodisk

1985: Negroponte and Wiesner co-founded the MIT Media Lab.

1989: Tim Berners-Lee proposed the World Wide Web.

1990: Kristina Hooper Woolsey headed the Apple Multimedia Lab.

1991: MPEG-1 was approved as an international standard for digital video later MPEG-2, MPEG-4 The introduction of PDAs in 1991 began a new period in the use of computers in multimedia.

1992: JPEG was accepted as international standard for digital image compression later JPEG2000 The first MBone audio multicast on the Net was made.

1993: The University of Illinois National Center for Supercomputing Applications produced NCSA. Mosaic-the first full-edged browser.

1994: Jim Clark and Marc Andreessen created the Netscape program.

1995: The JAVA language was created for platform-independent application development.

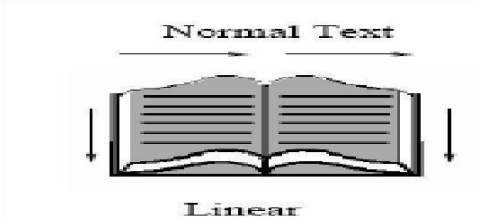
1996: DVD video was introduced; high quality full-length movies were distributed on a single disk.

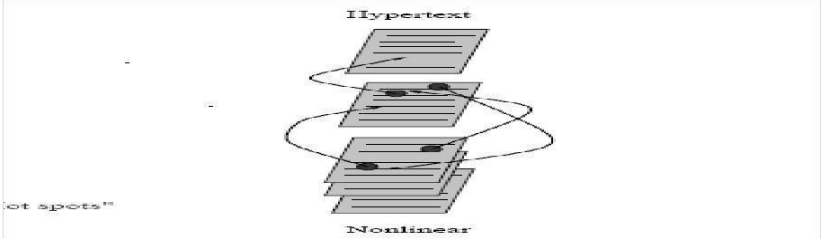
1998: XML 1.0 was announced as a W3C Recommendation. Hand-held MP3 devices first made with devices holding 32MB of flash memory.

2000: WWW size was estimated at over 1 billion pages.

Hypermedia and Multimedia
Ted Nelson invented the term "Hyper Text" around 1965 Types of media:

Linear media: Meant to read from **N** beginning to end. Ex: Text books	**on-linear media:** Meant to be read non-linearly, by following links that point to other parts of the document or to other documents Ex: Hyper Text system





Hypermedia: not constrained to be text-based, can include other media, e.g., graphics, images, and especially the continuous media - sound and video. Examples of Multimedia applications includes: Digital Video edition, E-magazines, WWW, Online reference books, games, home shopping, interactive TV, video conferencing, Interactive Movies. The World Wide Web (WWW) is best example of a hypermedia application.

World Wide Web:
WWW is maintained & developed by World Wide Web Consortium (W3C) and standardized by Internet Engineering Task Force (IETF). The W3C has listed the following goals for the WWW:

Universal access of web resources (by everyone every-where).
Effectiveness of navigating available information.
Responsible use of posted material.

1) History of the WWW:

1960: Charles Goldfarb et al. developed the Generalized Markup Language (GML) for IBM.
1986: The ISO released a final version of the Standard Generalized Markup Language (SGML).
1990: Tim Berners-Lee invented the Hyper Text Markup Language (HTML) & Hyper Text Transfer Protocol (HTTP).
1993: NCSA released an alpha version of Mosaic based on the version by Marc Andreessen for X-Windows the first popular browser.
1994: Marc Andreessen et al. formed Mosaic Communications Corporation later named as Netscape Communications Corporation.
1998: The W3C accepted XML version 1.0 specifications as a Recommendation. It is the main focus of W3C and supersedes HTML.

2) HTTP (Hyper Text Transfer Protocol):

HTTP is a protocol that was originally designed for transmitting hypermedia, but can also support the transmission of any file type. HTTP is a stateless request/response protocol: no information carried over for the next request.
The basic request format:
Method URI Version Additional-Headers Message-body
The URI (Uniform Resource Identifier): an identifier for the resource accessed, e.g. the host name, always preceded by the token "http://". URL could be Universal Resource Locator, if URI is included with Query strings. Method is a way of exchanging information or performing task on URI. Two popular methods:
GET method that the information requested is in the request string itself

POST method specifies that the resource pointed to URI should consider Message body Additional header specifies additional parameters about the client.

The basic response format:

Version Status-Code Status- Phrase Additional-Headers Message-body

Status code is number that identifies response type, Status Phrase is textual description of it. Two commonly seen status codes: 200 OK - the request was processed successfully, 404 Not Found - the URI does not exist.

3) HTML (Hyper Text Markup Language):

HTML is a language for publishing Hypermedia on the World Wide Web - defined using SGML. HTML uses ASCII, it is portable to all different computer hardware. The current version of HTML is version

4.01 in 1999. The next generation of HTML is XHTML - a reformulation of HTML using XML. HTML uses tags to describe document elements:

<token params> - defining a starting point,

</token> - the ending point of the element.

Some elements have no ending tags. A very simple HTML page is as follows:

```
<HTML>
<HEAD>
<TITLE>                A sample web page.      </TITLE>
<META NAME = "Author" CONTENT = "Cranky Professor">
</HEAD>
<BODY>
<P>We can put any text we like here, since this is a paragraph element.</P>
</BODY>
</HTML>
```

Naturally, HTML has more complex structures and can be mixed in with other standards. It allow integration with script languages, dynamic manipulation, modular customization with Cascading Style Sheets (CSS)

4) XML (Extensible Markup Language):

XML is a markup language for the WWW in which there is modularity of data, structure and view so that user or application can be able to define the tags (structure). Example of using XML to retrieve stock information from a database according to a user query,

First use a global Document Type Definition (DTD) that is already defined.

The server side script will abide by the DTD rules to generate an XML document according to the query using data from the database.

Finally send user the XML Style Sheet (XSL) depending on the type of device used to display the information.

The current XML version is XML 1.0, approved by the W3C in Feb. 1998. XML syntax looks like HTML syntax.

All tags are in lower case, and a tag that has only inline data has to terminate itself, i.e.<token params/>

Uses Name spaces so that multiple DTDs declaring different elements but with similar tag names can have their elements distinguished.

DTDs can be imported from URIs as well. An example of an XML document structure - the definition for a small XHTML document:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1 transition.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"> [html that follows the above mentioned XML rules] </html>
```

The following XML related specifications are also standardized:

XML Protocol: used to exchange XML information between processes.

XML Schema: a more structured and powerful language for defining XML data types (tags).

XSL: basically CSS for XML. it has three parts: XSLT, XPath, XSL Formatting Objects

SMIL: synchronized Multimedia Integration Language, pronounced “smile"-a particular application of XML (globally predefined DTD) that allows for specification of interaction among any media types and user input, in a temporally scripted manner.

5) SMIL (Synchronized Multimedia Integration Language):

Purpose of SMIL: it is also desirable to be able to publish multimedia presentations using a markup language. A multimedia markup language needs to enable scheduling and synchronization of different multimedia elements, and define their interactivity with the user. The W3C established a Working Group in 1997 to come up with specifications for a multimedia synchronization language. SMIL 2.0 was accepted in August 2001. SMIL 2.0 is specified in XML using a modularization approach similar to the one used in XHTML. All SMIL elements are divided into modules - sets of XML elements, attributes and values that define one conceptual functionality.

In the interest of modularization, not all available modules need to be included for all applications. Language Profiles: specifies a particular grouping of modules, and particular modules may have integration requirements that a profile must follow. SMIL 2.0 has a main language profile that includes almost all SMIL modules. Basic elements of SMIL as shown in the following example:

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<xlmns="http://www.w3.org/2001/SMIL20/Language"> <head>
<meta name="Author" content="Some Professor" />
</head>
<body>
<par id="MakingOfABook">
<seq>
        <video src="authorview.mpg" />
        <img src="onagoodday.jpg" />
</seq>
<audio src="authorview.wav" />
<text src="http://www.cs.sfu.ca/mmbook/" />
</par>
</body>
</smil>
```

Overview of Multimedia Software Tools:

The categories of software tools briefly examined here are:

- Music Sequencing and Notation
- Digital Audio
- Graphics and Image Editing
- Video Editing
- Animation
- Multimedia Authoring

1) Music Sequencing and Notation:

- **Cakewalk:** now called Pro Audio. The term sequencer comes from older devices that stored sequences of notes (“events", in MIDI). It is also possible to insert WAV files and Windows MCI commands (for animation and video) into music tracks
- **Cubase:** another sequencing/editing program, with capabilities similar to those of Cakewalk. It includes some digital audio editing tools.
- **Macromedia Soundedit:** mature program for creating audio for multimedia projects and the web that integrates well with other Macromedia products such as Flash and Director.

2) Digital Audio:

Digital Audio tools deal with accessing and editing the actual sampled sounds that make up audio:

- **Cool Edit:** a very powerful and popular digital audio toolkit; emulates a professional audio studio – multi track productions and sound file editing including digital signal processing effects.
- **Sound Forge:** a sophisticated PC-based program for editing audio WAV files.
- **Pro Tools:** a high-end integrated audio production and editing environment - MIDI creation and manipulation; powerful audio mixing, recording, and editing software.

3) Graphics and Image Editing:

- **Adobe Illustrator:** a powerful publishing tool from Adobe. Uses vector graphics; graphics can be exported to Web.
- **Adobe Photoshop:** the standard in a graphics, image processing and manipulation tool. Allows layers of images, graphics, and text that can be separately manipulated for maximum flexibility.
- Filter factory permits creation of sophisticated lighting-effects filters.
- **Macromedia Fireworks:** software for making graphics specifically for the web.
- **Macromedia Freehand:** a text and web graphics editing tool that supports many bitmap formats such as GIF, PNG, and JPEG.

4) Video Editing:

- **Adobe Premiere:** an intuitive, simple video editing tool for nonlinear editing, i.e., putting video clips into any order: Video and audio are arranged in "tracks". Provides a large number of video and audio tracks, superimpositions and virtual clips. A large library of built-in transitions, filters and motions for clips) effective multimedia productions with little effort.
- **Adobe After Effects:** a powerful video editing tool that enables users to add and change existing movies. Can add many effects: lighting, shadows, motion blurring; layers.
- **Final Cut Pro:** a video editing tool by Apple; Macintosh only.

5) Animation:

- **Multimedia APIs:**

  Java3D: API used by Java to construct and render 3D graphics, similar to the way in which the Java Media Framework is used for handling media files. Provides a basic set of object primitives (cube, splines, etc.) for building scenes. It is an abstraction layer built on top of OpenGL or DirectX (the user can select which).

  DirectX : Windows API that supports video, images, audio and 3-D animation

  OpenGL: the highly portable, most popular 3-D API.

- **Rendering Tools:**

  3D Studio Max: rendering tool that includes a number of very high-end professional tools for character animation, game development, and visual effects production.

  Softimage XSI: a powerful modeling, animation, and rendering package used for animation and special effects in films and games.

  Maya: competing product to Softimage; as well, it is a complete modeling package.

  Render Man: rendering package created byPixar.

  GIF Animation Packages: a simpler approach to animation, allows very quick development of effective small animations for the web.

6) Multimedia Authoring:

- **Macromedia Flash:** allows users to create interactive movies by using the score metaphor, i.e., a timeline arranged in parallel event sequences.
- **Macromedia Director:** uses a movie metaphor to create interactive presentations very powerful and includes a built-in scripting language, Lingo, which allows creation of complex interactive movies.
- **Authorware:** a mature, well-supported authoring product based on the conic/Flow-control metaphor.
- **Quest:** similar to Authorware in many ways, uses a type of owcharting metaphor. However, the owchart nodes can encapsulate information in a more abstract way (called frames) than simply subroutine levels.

Image/Graphics Data representation:

Image/Graphics Data Types:

There are number of file formats used in multimedia to represent image or graphics data. In general, image or graphics data can be represented as follows:

1) 1-bit Images:

Image Consist of pixels or pels – picture elements in digital images. It contains On(1) or Off(0) bits stored in single bit. So they are also known as **Binary image**. It is also called as Mono chrome image because it contains no color. 640x480 image Requires 38.4 KB of storage.

2. 8-bit gray Level Images:

Consider 8-bit image, One for which each pixel has Gray value between 0 to 255 stored in single byte. Image is a Two dimensional array known Bitmap. **Image resolution** refers to number of pixels in digital image like 1600x1200 is high resolution where as 640x480 is low resolution with aspect ration of 4:3. **Frame buffer** is a

hardware used to store array of pixels of image. Special hardware is used for this purpose known as Video/ Graphics card. 8-bit image is a collection of 1-bit bit planes. 640x480 image requires 300 KB of storage.

Dithering:

Printing images is a complex task, 600 Dot per Inch (dpi) laser printer can usually print a dot or not print it. However, 600x600 image will be printed in 1-inch space. Basic strategy of dithering is to trade Intensity resolution for spatial resolution. For printing 1-bit printer, dithering is used to calculate larger patterns of dots. Replace a pixel value by a larger pattern say 2x2, 4x4.

Halftone printing: Number of printed dots approximates varying sized disks of ink, which is a analog process that uses smaller or larger filled circles of black ink to represent shading. Use NxN matrix of on-off 1- bit dots, 2^2 represents n +1 levels of intensity resolution. Dot patterns are created heuristically known as Dither matrix.

Rules:

- If intensity > dither matrix entry, print ON dot at that entry location
- Replace each pixel by NxN matrix of dots

Number of levels is small for this type of printing, if number of effective intensity levels are increased by increasing dither matrix size, size of output image also increases

Ordered Dither: consists of turning on printer output bit for pixel if intensity level is greater than particular matrix element just at that pixel position.

Dithering Algorithm: for NxN dither matrix

3) 24-bit color image:

In color 24-bit images, Each pixel is represented by three bytes, usually representing components R, G, B. Each value is in range 0-255, this format supports 256x256x256 possible combined colors. 640x480 size image takes 921.6 KB of storage. Actually it is stored in 32 bits, extra byte of data for each pixel storing alpha value for representing special effect information. Alpha channel is used for overlapping graphical objects known as Transparency.

4) 8-bit color image:

Accurate color images can be obtained by quantizing color information to 8-bit, Called as 256 color image. 24 bit image is obtained from 8-bit using a concept of Lookup Table.

Color Lookup table: which stores color information. Image stores just set of bytes which is an index into table with 3 byte values that specify color for pixel. It is useful to choose which colors to represent best in image.

Color histogram: all colors in 24 bit image are collected in 256x256x256 set of cells, along with count of how many pixels belong to each of these colors stored in that cell. Then we can get a three dimensional array structure known as Color Histogram. Few important clusters of color information, corresponding to R, G, and B allows us to pick most important 256 groups of colors for construction of Color look up table.

Color Lookup Tables (LUT):

Idea used in 8-bit color images is to store only index or code value for each pixel. While images are displayed as two dimensional arrays of values, they are usually stored in row-column order as simply a long series of values. LUT is often called a Palette.

Color picker: consists of an array of fairly large blocks of color, such that mouse click will select color indicated. A simple animation process is possible via simply changing color table Known as Color Cycling or Palette Animation. Dithering can also be carried out for color printers, using 1 bit per color channel and spacing out color with R, G, and B dots.

How to devise a Color Lookup Table

It gives idea of clustering to generate most important 256 colors from 24-bit color image. In general, clustering is expensive & slow process. This can be done in two ways:

Method 1: it is straight forward way to make 8-bit lookup color out of 24-bit colors by dividing RGB cube into equal slices in each dimension.

- Divide 24 bit color RGB cube into equal slices in each dimension
- Center of each of resulting cube would serve as entries in color LUT
- This will scale RGB of 24 bit into 8 bit code
- Shrinking R range from 0 to 255 0 to 7 which takes 3 bits only
- Similarly, G range from 0 to 255 0 to 7 which takes 3 bits only

- Finally, B range from 0 to 255 0 to 3 which takes 2 bits only
- Combining all the resulting bits give 8 bit color value

Method 2: Median Cut Algorithm

This approach derives from computer graphics. The idea is as follows:

- Sort R byte values & find their median. Then values smaller than median are labeled with 0 bit & values larger than median are labeled with 1 bit.
- Next consider only pixels with 0 label from first step & sort their G values. Again label image pixels with another bit 0 for less than median in greens & 1 for greater
- Carrying on to blue channel, we have 3-bit scheme
- Repeating all steps, R, G, B results 6-bit & cycling through R & G once more results 8-bits

These bits form out 8-bit color index value of pixels & 24-bit colors can be centers of resulting small color cubes. Accurate version of Median Cut algorithm

- Find smallest box that contains all colors
- Sort enclosed colors along longest dimension of box
- Split box into two regions at median of sorted list
- Repeat 2& 3 until original color space divided to 256 regions
- For every box, call mean of R, G, B in that box representative color for box
- Based on Euclidean distance between pixel RGB & box center assign every pixel to one of representative colors,
- Repeat pixel by code in lookup table that indexes representative colors

Popular File Formats:

1) Graphics Interchange Format(GIF):

GIF was devised by UNISYS for transmitting images over phone lines. It uses Lempel-Ziv-Welch algorithm. It is limited to 8-bit color image only. It produces acceptable color with few distinctive colors. Support It supports Interlacing - successive display of pixels by 4 pass display.GIF comes in two Versions: GIF87a, GIF 89a. it supports simple animation with Graphics Control Extension Block. This provides simple control over delay time, transparency index. GIF file format includes: GIF Signature (6 Bytes), Screen Descriptor (7 Bytes), Global Color Map, Image Information, GIF Terminator. Each image can contain its own color lookup table known as Local color map.

GIF File format	**GIF screen descriptor**	**GIF color Map**	**GIF image descriptor**

Screen descriptor comprises set of attributes that belong to every image in file that specified Screen width(2 bytes), Height(2 Bytes), m in byte 5 is 0 if no global color map is given, Color resolution(cr) is 3 bits, Pixel is another 3 bits indicating number of bits per pixel in image.

Color map actual length is $2^{pixel+1}$.

Each image file has its own Image descriptor. If Interlace bit is set in local Image Descriptor, rows of image are displayed in 4 pass sequence. JPEG uses Progressive Mode display. GIF Four pass interlace display. Actual raster data is first compressed using LZW compression before being stored.

2) Joint Photographic Experts Group(JPEG):

Most important current standard for image compression is JPEG. It was created by ISO. Eye brain system cannot see excrementally fine detail. If many changes occur within few pixels, we refer to that image segment as having High Spatial Frequency i.e. great change in (x,y) space. Color information is partially dropped or averaged & then Small blocks of image are represented in spatial frequency domain (u,v). values are divided by some large integer & truncated. In this way, small values are zeroed out. This compression scheme is lossy. It is straightforward to allow user to choose how large denominator to use & hence how much information to discard. This will allows to choose desired quality of image. Usual default quality factor is Q=75%.

3) Portable Network Graphics(PNG):

It is System independent image format. Motivated by UNISYS on LZW compression method. Special features of PNG files include support 48-bit color information. Files may contain gamma correction, alpha channel information such as channel of transparency. Supports progressive display pixels in two dimensional fashion few at time over seven passes through each 8x8 block of image

4) Tagged Image File Format(TIFF):

It is developed by Aldus Corporation, support Microsoft. It ssupports attachments of additional information known Tags provides flexibility. Most tags are format signifiers. Different types of images: 1-bit, gray scale, 8-bit, 24-bit are supported.

5) Exchange Image File(EXIF):

It is image format for Digital cameras, published in 2002 by JEITA. Compressed EXIF files are use baseline JPEG format. Variety of tags available for higher quality printing. Picture taking conditions: light source, white balance. It also includes specification of file format for audio that accompanies digital images.

6) Graphics Animation Files:

Few formats are aimed at storing graphics animations. FLC is important animation or moving picture file format. It was originally created by Animation Pro. GL produces some what better quality moving pictures. GL animations can also usually handle larger file sizes.

7) PS & PDF:

Post Script is language for typesetting & many high end printers have PS interpreter built into them. PS is vector based, picture language. Page elements are essentially defined in terms of vectors. It includes Text as well as vector/structured graphics, bit mapped images. PS does not provide compression it self, are just stored as ASCII. Portable Document Format(PDF) includes Text and Figure language with LZW compressing method. Provide higher compression with JPEG compression.

8) Windows WMF:

Windows Meta File is native vector file format. It is collection of Graphics Device Interface(GDI) function calls. Device independent & unlimited in size.

9) Windows BMP:

Bitmap is system standard for Microsoft windows. It uses Run length encoding compression & can fairly Store 24 bit bitmap image. BMP has many different modes including uncompressed 24 bit images.

10) Macintosh PAINT & PICT:

PAINT used in MacPaint program only for 1 bit monochrome images. PICT used in MacDraw structured graphics.

11) X Windows PPM:

For X Window system, Portable Pix Map support 24 bit color bitmap & can be manipulated using many public domain graphic editors.

Color Science

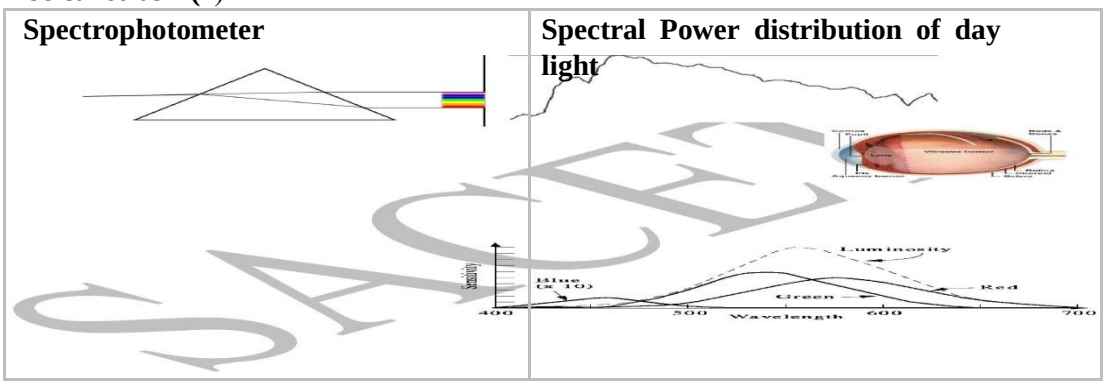
Light & Spectra:

Light is Electro magnetic wave and Color is characterized by wave length of wave. Laser light has single wave length, in contrast most light sources produce contributions over many wave lengths. Humans cannot detect all lights, just contributions that fall in visible wave length.

Spectrophotometer: is used to measure visible light, by reflecting light from diffraction grating that spreads out the different wave lengths. Diffraction & Dispersion will generates rainbow effect. If we look light through

prism, we will get this rainbow effect due to Dispersion. Visible light Ranges from 400 nm – 700 nm

Spectral Power Distribution (SPD) or Spectrum: shows relative amount of light energy at each wave length & this type of cure can be called as **E(**λ).



Human vision:

Eye works like a camera with lens focusing image on to Retina. Retina contains array of Rods & three Kinds of Cones. Rods come into play when light level are low & produce shades of gray. Cones are activated to sense high levels of light by differing pigments to produce color images. Three kinds of cones are more sensitive to Red(R), Green(G) & Blue(B) light.

Spectral Sensitivity of Eye:

Eye is most sensitive to light in middle of visible spectrum. Like PSD profile for light source, for receptors we have relative sensitivity as function of wave length. Blue receptor sensitivity is not shown to scale, because it is much smaller than curves for red or green. Overall sensitivity of R, G, B is known as Luminous Efficiency Function **V(**λ**)**. Rods are sensitive to broad range of wave lengths, but produce signal that generates perception of black white-scale only. Eye has about 6 million cones, but proportions of R,G & B cones are different. They likely are present in rations 40:20:1. These Spectral sensitivity functions are usually denoted by vector function **q(**λ) with components. $\mathbf{q}(\lambda)=[q_R(\lambda),q_G(\lambda),q_B(\lambda)]^T$

Response in each color channel in eye is proportional to number of neurons firing. Think that sensitivities are continuous functions, then three colors form three dimensional vector space.

Image Formation:

Above equations are useful for self luminous object. But in most situations, image is formed by light reflected from surface. Surfaces reflect different amounts of light at different wave lengths & dark surfaces reflect less energy than light surfaces. Image formation system can be seen as follows:

Light from illuminant with SPD **E(**λ) falls on surface with surface reflectance function **S(**λ) is reflected & then filtered by eye's cone functions **q(**λ). The function **C(**λ) is called Color Signal & product of illuminant **E(**λ) &.

Camera Systems: A good camera has three signals produces at each pixel location. Analog signals are converted to digital, truncated to integers & stored. If precision used is 8-bit maximum value for any R, G, B is 255 & minimum is 0.

Gamma Correction:

RGB numbers in image file are converted back to analog & drive electron guns in cathode ray tube(CRT). Electrons are emitted proportional to driving voltage. CRT system produce light linearly related to voltage. Light emitted is roughly proportional to voltage raised to power. This power is called Gamma (γ). If the file γ value in Red channel is R, screen emits light proportional to R . Value of Gamma is around 2.2. Gamma iscorrected by raising to power (1/γ) before transmission. Thus we have

Why gamma is 2.2?

Power law for color receivers (like TV) may be actuality be closer to 2.8. However, if we compensate for only about 2.2 of this power law, we arrive at overall value about 1.25 instead of 1.0. Darker colors are made even darker. Camera may or may not insert gamma correction, Software may write image file using gamma, may decode expecting some gamma. Most common machines, might make sense to gamma correct images at average of about 2.1. Most practitioners might use value of 2.4 adopted by sRGB group. A new standard RGB for WWW applications called sRGB. Eye is most sensitive to ratios of intensity levels rather than absolute intensities. This means that brighter the light, greater must be change in light level for change to be perceived.

Color Matching Functions:

Recreating desired colors can be done by psychology for matching combination of basic R, G, B lights to given shade. Particular set of three basic lights was available, called set of Color primaries. A device for carrying out such experiment is called Colorimeter**.** It uses set of three colors brightness is adjusted to match desired color. International standards body for color, Commission International deLEclairage(CIE) pooled all

such data in 1931, in set of curves called Color Matching Functions. Color matching experiments are summarized by statement of proportion of color primaries needed for each individual narrow band wavelength of light.

CIE color matching curves, denoted r(λ), g(λ), b(λ). This curve indicates that some colors can not be reproduced by linear combination of primaries. That why r,g,b color matching functions produce some parts of negative curve.

For such colors, one or more of primary lights

has to be shifted from one side of black partition to other so they illuminate sample to matched instead of white screen. Set of virtual primaries are devised that led to color matching functions with only positive values.

CIE chromaticity diagram:

Result from liner (3x3 matrix)Transform from r, g, b curves, are denoted x(λ), y(λ), z(λ). Matrix is chosen such that middle standard color matching function y(λ) exactly equals luminous efficiency curve V(λ). Essential colorimetric information required to characterize set of Tri-stimulus values X, Y, Z defined as

Middle value Y is called Luminance. All color information & transforms are tied to these special values. 3D diagrams are difficult to visualize & CIE devised 2D diagram based on values of (X, Y,Z). It makes sense to devicse 2D diagram by some how factoring out magnitude of vectors (X, Y, Z). in CIE system this is accomplished like X+Y+Z:

This effectively means that one value out of set (x,y,z) is redundant, since we have:

Values x, y are called chromaticities. Locus of points of monochromatic light drawn on this CIE chromaticity diagram. The straight line along the bottom of Horseshoe joints points at extremities of visible spectrum 400 & 700 nanometers. The straight diagonal line in the diagram is called Line of purples. Middle of the diagram displays white point. All possible chromaticity values must necessarily lie below line of purple. Horseshoe itself is called Spectrum locus. Colors with chromaticities on spectrum locus represent pure colors. These are most saturated. Colors closer to white point are more unsaturated.

Color Monitor Specifications:

Color monitors are specified in part by White point chromaticity desired if the RGB electron guns are all activated at their highest power. If we normalize voltage range 0 to 1, then we wish to specify monitor such that it displays desired white point when R'=G'=B'=1. Phosphorescent paint on the monitor screen have their own chromaticities. There are several monitor specifications are in use, some of them are NTSC, SMPTE, EBU

System	Red		Green		Blue		White Point	
	xr	yr	xg	yg	xb	yb	xw	yw
NT	0.6	0.3	0.2	0.7	0.1	0.0	0.3	0.3
SC	7	3	1	1	4	8	10 1	16 2
S	0.6	0.3	0.3	0.5	0.1	0.0	0.3	0.3
M	30	40	10	95	55	70	12	29
PT							7	1
E								
EB	0.6	0.3	0.2	0.6	0.1	0.0	0.3	0.3
U	4	3	9	0	5	6	12 7	29 1

Out of Gamut Colors:

Basic problem for displaying color is how to generate Device-independent color, by (x,y) chromaticity values, using Device-dependent color values RGB. For any (x,y) pair (x,y,z) must be specified by forming z values for phosphors via z=1-x-y & solve for RGB from manufacturer specified chromaticities, we combine non zero values of RGB. If any of the RGB numbers is Negative, it is not represent able on the device being used. This case is known as Out of Gamut, since set of all possible displayable colors consist gamut of device. One method to deal with this is to

simply use closest in gamut color available. Another common approach is to select closest complementary color. For monitor, every displayable color within Triangle of CIE chromaticity diagram, Known as Grassman's Law describing human vision. If we compose colors from linear combination of three lights available from the three phosphors, we can create colors only from convex set derived from lights.

White Point Correction:

The difference is that XYZ values includes magnitude of color. We also need to be able to alter matters such that when each R, G, B is at maximum value, we obtain white point.

X= 0.630+0.310+0.155 =1.095
Y= 0.340+0.595+0.070 =1.005
Z= 0.03+0.095+0.775 =0.9

Transformation with Gamma Correction:

Instead of liner R, G, B, non linear gamma corrected R',G',B' values are used. Best way of carrying out XYZ-to-RGB transform to calculate liner RGB required then create non linear signals via gamma correction. Only concession to accuracy is to give new name Y' is used to indicated accurate Y from R', G', B'. Original NTSC system, require following transform:

X=0.607.R+0.174.G+0.200.B Y=0.299.R+0.587.G+0.114.B Z=0.000.R+0.066.G+1.116.B

Coding for Non linear signals begins with encoding non-linear signal correlate of luminance:
Y'=0.299.R'+0.587.G'+0.114.B'

L*a*b* (CIELAB) color model:

Weber's Law: more there is quality, more change there must be perceive difference. Rule of thumb for this phenomenon states that equally perceived changes must be relative. Changes are about equally perceived if ratio of changes is same, whether for dark or bright lights & so on. This idea leads to logarithmic approximation to perceptually equally spaces units. For human vision, CIE arrived at some what more involved version of this kind of rule called CIELAB.

More Color Coordinate Schemes:

Several other co-ordinates schemes in use to describe color as humans to perceive its gamma correction.

Other schemes include:

- CMY (Cyan, Magenta, Yellow)
- HSL(Hue, Saturation, Light)
- HSV(Hue, Saturation, Value)
- HIS(Hue, Saturation, Intensity)
- HCI(Hue, Chroma, Intensity)
- HVC(Hue, Value, Chroma)
- HSD(Hue, Saturation, Darkness)

Munsell Color Naming System:

Accurate Naming of colors is also important consideration. Munsell re-notation: Munsell devised one time tested standard system in 1900's & revised may times. Idea is to set up approximately perceptually uniform system of three axes to discuss & specify color. The axes are Value(black-white), Hue & Chroma, Value is divides into 9 steps. Hue is in 40 steps around circle. Chroma has maximum of 16 levels.

Color Models in Images:

RGB color model for CRT display:

Usually color information is stored directly in RGB form**.** 8 bits per color channel for color is accurate enough, in fact about 12 bits per channel need to be used to avoid aliasing effect in dark images**.** We have gamma correction LUT between frame buffer and CRT

Subtractive Color: CMY color model

Additive color, when two light beams impose on target,

their colors add Ex: red + green = yellow. For ink deposited on paper, opposite situation occurs: yellow ink subtracts blue from white illumination but reflects red and green. Subtractive color primaries are Cyan(C), Magenta (M), Yellow(Y) inks. In additive RGB system, black is no light i.e. R=G=B=0; in subtractive CMY system, black arises from subtracting all light by laying down inks with C=M=Y=1.

Color Models in Video:

Video Color Transforms:

Methods for dealing with color in digital video derive largely from older analog methods of coding color for TV. Some version of luminance is combined with color information in the signal. Matrix transform method called YIQ is used to transmit TV signals in North America and Japan. YUV matrix transform signals with PAL or SECAM coding are used in Europe. Digital video mostly uses matrix transform called YCbCr that is closely related to YUV

YUV color model:

It was used for PAL analog video, it version is CCIR 601 standard for digital video. First it codes gamma corrected Luminance signal equal to Y'. Chrominance refers to difference between color and reference white at the same luminance. It can be represented by color differences U, V: U=B'-Y' & V=R'-Y' We can get the equations as follows:

For gray pixel with R'=G'=B', luminance Y' is equal to that same gray value. For gray image, chrominance (U, V) is zero, since sum of coefficients in each of lower two equations is zero. Black & White TV only uses Y' signal to display no color information. For analog video, U & V are scaled to the range between ±0.5 times the maximum of Y'. this reflects dealing of component video with three separate signals. Component

Then chrominance signal is composed from U & V as composite signal

$C=U.\cos(\omega t) + V.\sin(\omega t)$ where ω represents NTSC color frequency

U is approximately from blue (U>0) to yellow(U<0) in RGB, V is from red (V>0) to cyan (V<0)

YIQ color model:

Used in NTSC color TV broadcasting, gray pixels generate zero chrominance signals. Original names came from combinations of analog signals – I for in-phase chrominance, and Q for Quadrature Chrominance. In YUV model U, V do not best correspond to actual human perceptual color senilities, but NTSC uses I and Q instead. YIQ is just version of YUV with Same Y' but U and V rotated by 33^o

$I=0.877283(R'-Y') \cos 33^o - 0.492111(B'-Y') \sin 33^o$

$Q=0.877283(R'-Y') \sin 33^o - 0.492111(B'-Y') \cos 33^o$

I corresponds orange-blue direction, Q corresponds to purple-green direction. YIQ decomposition does better job of forming hierarchical sequence of images for 8-bit Y', Root-Mean-Square(RMS) value is 137. U, V components have RMS values 43, 44; I, Q components have 42, 14 that prioritize color values.

YCbCr color model:

International standard for component digital video is officially recommended ITU-R known as Rec. 601, which uses another color space YCbCr. YCbCr transform is used in JPEG & MPEG compression and is closely related to YUV transform. YUV is changed by scaling such that Cb is U with coefficient of 0.5 multiplying B'. In some cases Cb & Cr are shifted between 0 to 1with following equations:

$Cb=((B'-Y')/1.772)+0.5$

$Cr=((R'-Y')/1.402)+0.5$

Then the YCbCr transform 8-bit coding, with maximum Y' value of only 219 and minimum of 16. Values below 16 and above 235, denoted as Head room and Foot room. We obtain Y', Cb, Cr as follows:

UNIT - II
Fundamental Concepts in Video

Component Video

Higher-end video systems, such as for studios, make use of three separate video signals for the red, green, and blue image planes. This is referred to as componellt video. This kind of system has three wires (and connectors) connecting the camera or other devices to a TV or monitor.
Color signals are not restricted to always being RGB separations. Instead, as we saw in. Chapter 4 on color models for images and video, we can form three signals via a luminance chrominance transformation of the RGB signals - for example, YIQ or YUV. In contrast, most computer systems use component video, with separate signals for R, G, and B signals. For any color separation scheme, component video gives the best color reproduction, since there is no "crosstalk" between the three different channels, unlike composite video or S-video. Component video, however, requires more bandwidth and good synchronization of the three components.

Composite Video

In composite video, color ("chrominance") and intensity ("luminance") signals are mixedinto a single carrier wave. Chrominance is a composite of two color components (I andQ, or U and V). This is the type of signat'used by broadcast color TVs; it is downward compatible with black-and-white TV.
In NTSC TV, for example [1], [and Q are combined into a chroma signal, and a color subcanier then puts the chroma signal at the higher frequency end of the channel shared with the luminance signal. The chrominance and luminance components can be separated at the receiver end, and the two color components can be further recovered.
When connecting to TVs or VCRs, composite video uses only one wire (and hence one connector, such as a'BNe connector at each end of a coaxial cable or an RCA plug at each end of an ordinary wire), and video color signals are mixed, not sent separately. The audiosignal is another addition to this one signal. Since color information is mixed and both color and intensity are wrapped into the same signal, some interference between the luminance and chrominance signals is inevitable.

S-Video

As a compromise, S-video (separated video, or super-video, e.g" in S-VHS) uses two wires: one for luminance and another for a composite chrominance signal. As a result, there is less crosstalk between the color information and the crucial gray- scale information. The reason for placing luminance into its own part of the signal is that black-and-white information is cLUcial for visual perception. As noted in the previous chapter, humans are able to differentiate spatial resolution in grayscale images much better than for the color part of color images (as opposed to the "black-and-white" part). Therefore, color information sent can be much less accurate than intensity information. We can see only fairly large blobs of color, so it makes sense to send less color detail.

ANALOG VIDEO

Most TV is still sent and received as an analog signal. Once the electrical signal is received, we may assume that brightness is at least a monotonic function of voltage, if not necessarily linear, because of gamma correction.
An analog signal !(t) samples a time-varying image. So-called progressive scanning traces through a complete picture (a frame) row-wise for each time interval. A high resolution computer monitor typically uses a time interval of 1n2 second.
In TV and in some monitors and multimedia standards, another system, interlaced scanning, is used. Here, the odd-numbered lines are traced first, then the even-numbered lines. These results in "odd" and "even" fields - two fields make up one frame.
In fact, the odd lines (starting from 1) end up at the middle of a line at the end of the odd field, and the even scan starts at a half-way point. Figure 5.1 shows the scheme used. First the solid (odd) lines are traced - P to Q, then R to S, and so on, ending at T – then the even field starts at U and ends at V. The scan lines are not horizontal because a small Voltage is applied, moving the electron beam down over time.

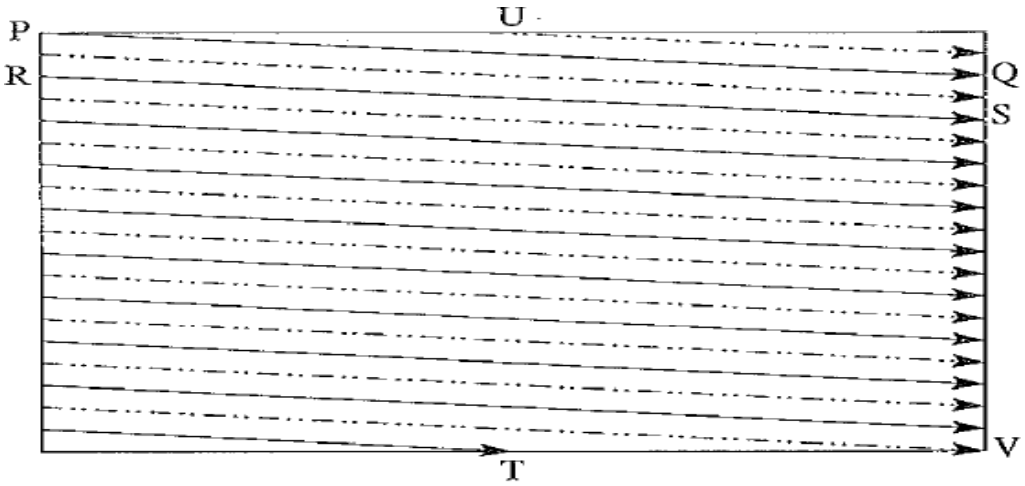


FIGURE 5.1: Interlaced raster scan.

Interlacing was invented because, when standards were being defined, it was difficult to transmit the amount of information in a full frame quickly enough to avoid flicker. The double number of fields presented to the eye reduces perceived flicker. Because of interlacing, the odd and even lines are displaced in time from each other. This is generally not noticeable except when fast action is taking place onscreen, when blurring may occur. For example, in the video in Figure 5.2, the moving helicopter is blurred more than the still background. Since it is sometimes necessary to change the frame rate, resize, or even produce stills from an interlaced source video, various schemes are used to de-interlace it. The simplest de-interlacing method consists of discarding one field and duplicating the scan lines of the other field, which results in the information in one field being lost completely. Other, more complicated methods retain information from both fields.

CRT displays are built like fluorescent lights and must flash 50 to 70 times per second to appear smooth. In Europe, this fact is conveniently tied to their 50 Hz electrical system, and they use video digitized at 25 frames per second (fps); in North America, the 60 Hz electric system dictates 30 fps. The jump from Q to R and so on in Figure 5.1 is called the horizontall"etrace, during which the electronic beam in the CRT is blanked. The jump from T to U or V to P is called the vertical retrace. Since voltage is one-dimensional - it is simply a signal that varies with time - how do we know when a new video line begins? That is, what part of an electrical signal tells us that we have to restart at the left side of the screen? The solution used in analog video is a small voltage offset from zero to indicate black and another value, 'Such as zero, to indicate the start of a line. Namely, we could use a “blacker-than-black" zero signal to indicate the beginning of a line.

Figure 5.3 shows a typical electronic signal for one scan line of NTSC composite video. 'White' has a peak value of 0.714 V; 'Black' is slightly above zero at 0.055 V; whereas

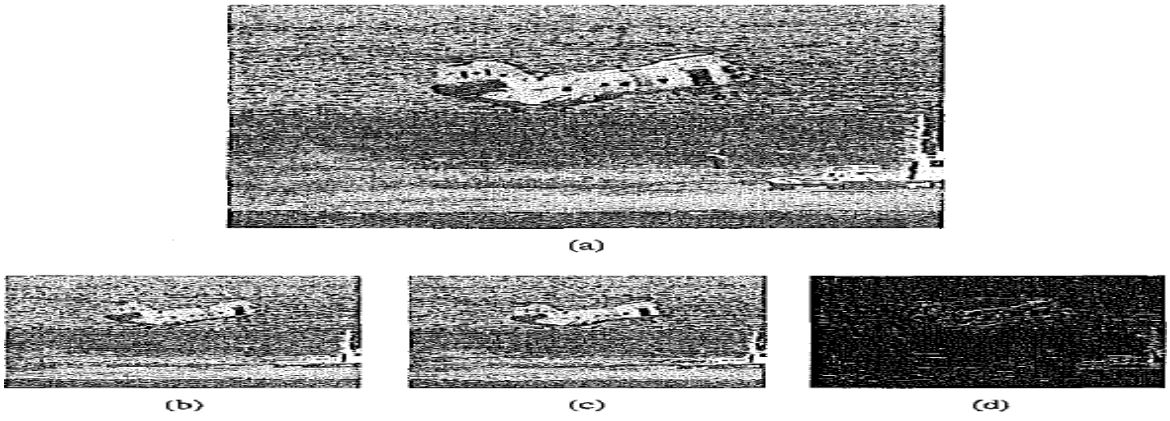


FIGURE 5.3: Interlaced scan produces two fields for each frame: (a) the video frame;(b) Field 1; (c) Field 2; (d) difference of Fields. Blank is at zero volts. As shown, the time duration for blanking pulses in the signal is used for synchronization as well, with the tip of the Sync signal at approximately -0.286 V. In fact, the problem of reliable synchronization is so important that special signals to control sync take up about 30% of the signal.

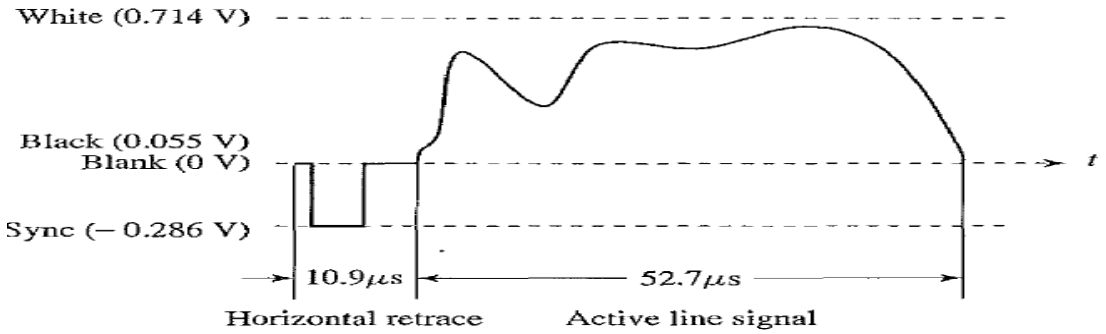


FIGURE 5.3: Electronic signal for one NTSC scan line.

The vertical retrace and sync ideas are similar to the horizontal one, except that they happen only once per field. Tekalp [2] presents a good discussion of the details of analog(and digital) video. The handbook [3] considers many fundamental problems in video processing in great depth.

NTSC Video

The NTSC TV standard is mostly used in North America and Japan. It uses a familiar 4:3aspect ratio (i.e., the ratio of picture width to height) and 525 scan lines per frame at 30frames per second.

More exactly, for historical reasons NTSC uses 29.97 fps -or, in other words, 33.37 msec per frame. NTSC follows the interlaced scanning system, and each frame is divided into two fields, with 262.5 lines/field. Thus the horizontal sweep frequency is 525 x 29.97 :::::::15, 734 lines/sec, so that each line is swept out in 1/15,734 sec::::::: 63.6 /Lsec. Since the horizontal retrace takes 10.9 /Lsec, this leaves 52.7 /Lsec for the active line signal, during which image data is displayed (see Figure 5.3).

Figure 5.4 shows the effect of "vertical retrace and sync" and "horizontal retrace andsync" on the NTSC video raster. Blanking information is placed into 20 lines reserved for control information at the beginning of each field. Hence, the number of active video linesper frame is only 485. Similarly, almost 1/6 of the raster at the left side is blanked for horizontal retrace and sync. The non blanking pixels are called active pixels.

Pixels often fall between scan lines. Therefore, even with non interlaced scan, NTSC TVis capable of showing only about 340 (visually distinct) lines, - about 70% of the 485specified active lines. With interlaced scan, it could be as low as 50%.

Image data is not encoded in the blanking regions, but other information can be placed there, such as V-chip information, stereo audio channel data, and subtitles in many languages.

NTSC video is an analog signal with no fixed horizontal resolution. Therefore, we must decide how many times to sample the signal for display. Each sample cOlTesponds to one pixel output. A pixel clock divides each horizontal line of video into samples. The higher the frequency of the pixel clock, the more samples per line.

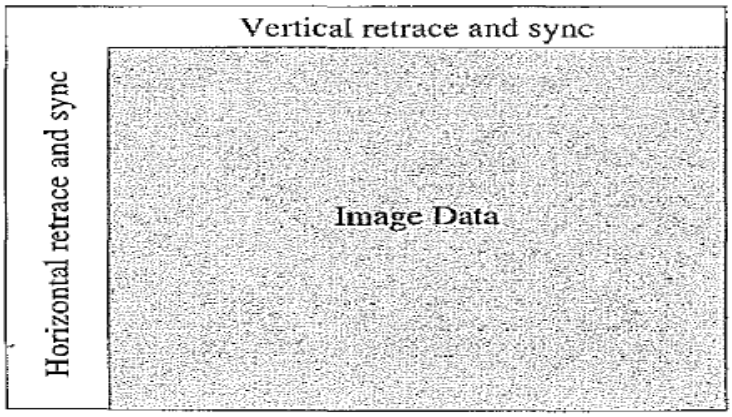


FIGURE 5.4: Video raster, including retrace and sync data.

TABLE 5.1: Samples per line for various analog video formats.

Format	Samples per line
VHS	240
S-VHS	400–425
Beta-SP	500
Standard 8 mm	300
Hi-8 mm	425

Different video formats provide different numbers of samples per line, as listed in Table 5.1. Laser disks have about the same resolution as Hi-8. (In comparison, miniDV 114-inch tapes for digital video are 480 lines by 720 samples per line.)NTSC uses the YIQ color model. We employ the technique of quadrature modulation to combine (the spectrally overlapped part of) / (in-phase) and Q (quadrature) signals into a single chroma signal C [1, 2]: C = I cos(Fsct) + Q sin(Fsct) (5.1)This modulated chroma signal is also known as the color subcarrier, whose magnitude is J/2 + Q2 and phase is tan- 1(Q/I). The frequency of Cis Fsc ~ 3.58 MHz. The / and Qsignals are multiplied in the time domain by cosine and sine functions with the frequency Fsc [Equation (5.1)]. This is equivalent to convolving their Fourier transforms in the frequency domain with two impulse functions at Fsc and - Fsc. As a result, a copy of J and Q frequency spectra are made which are centered at Fsc and - Fsc , respectively. The NTSC composite signal is a further composition of the luminance signal Y and the chroma signal, as defined below:

$$\text{Composite} = Y + C = Y + 1 \cos(\text{Fsct}) + Q \sin(\text{Fsct}) \quad (5.2)$$

NTSC assigned a bandwidth of 4.2 MHz to Y but only 1.6 MHz to I and 0.6 MHz to Q, due to humans' insensitivity to color details (high-frequency color changes). As Figure 5.5 shows, the picture carrier is at 1.25 MHz in the NTSC video channel, which has a total bandwidth of 6 MHz. The chroma signal is being "carried" by Fsc ~ 3.58 MHz towards the higher end of the channel and is thus centered at 1.25+3.58 = 4.83 MHz. This greatly reduces the potential interference between the Y (luminance) and C (chrominance) signals, since the magnitudes of higher-frequency components of Y are significantly smaller than their lower frequency counterparts. Moreover, as Blinn[l] explains, great care is taken to interleave the discrete Y and C spectra so as to further reduce tlie interference between them. The "interleaving" is illustrated in Figure 5.5, where the frequency components for Y (from the discrete Fourier transform) are shown as solid lines, and those for I and Q are shown as dashed lines. As

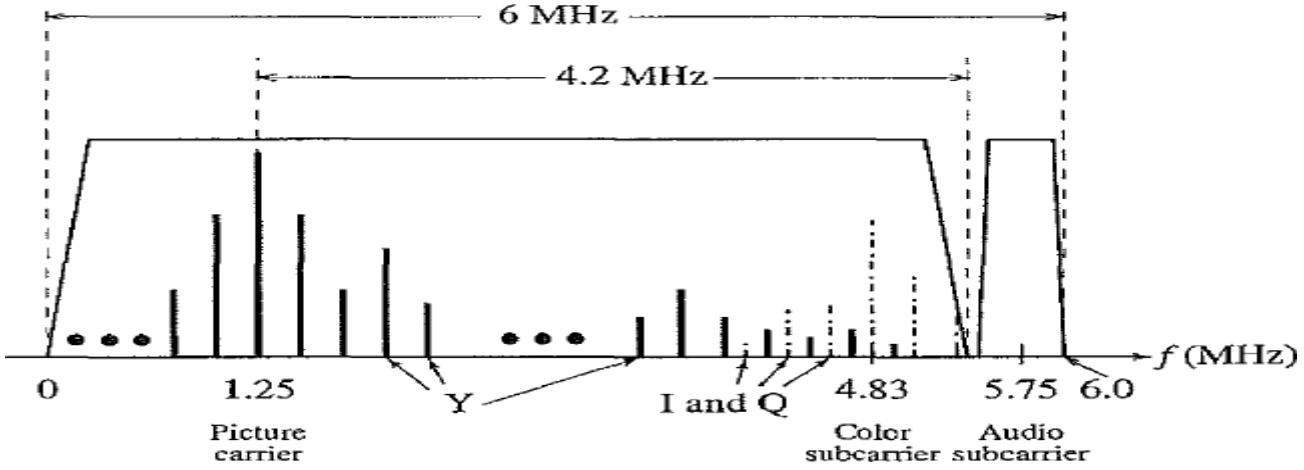


FIGURE 5.5: Interleaving Y and C signals in the NTSC spectrum.

PAL Video

PAL (Phase Alternating Line) is a TV standard originally invented by German scientists. It uses 625 scan lines per frame, at 25 frames per second (or 40 msec!frame), with a 4:3 aspect ratio and interlaced fields. Its broadcast TV signals are also used in composite video. This Important standard is widely used in Western Europe, China, India and many other parts of the world. PAL uses the YUV color model with an 8 MHz channel, allocating a bandwidth of 5.5 MHz to Y and 1.8 MHz each to U and V. The color subcarrier frequency is fsc ~4.43 MHz. To improve picture quality, chroma signals have alternate signs (e.g., +U and - U) in successive scan lines; hence the name "Phase Alternating Line.,,2 This facilitates the use of a (line-rate) comb filter at the receiver - the signals in consecutive lines are averaged so as to cancel the chroma signals (which always CalT)' opposite signs) for separating Y andC and obtain high-quality Y signals.

SECAM Video

SECAM, which was invented by the French, is the third major broadcast TV standard. SECAM stands for System Electronique COlllell" Avec Memoire. SECAM also uses 625 scan lines per frame, at 25 frames per second, with a 4:3 aspect ratio and interlaced fields.

The Oliginal design called for a higher number of scan lines (over 800), but the final version settled for 625. SECAM and PAL are similar, differing slightly in their color coding scheme. In SECAM, U and V signals are modulated using separate color subcarriers at 4.25 MHz and 4.41 MHz, respectively. They al'e sent in alternate lines - that is, only one of the U or V signals will be sent on each scan line. Table 5.2 gives a comparison of the three major analog broadcast TV systems.

DIGITAL VIDEO

- The advantages of digital representation for video are many.
- Storing video on digital devices or in memory, ready to be processed (noiseremoval, cut and paste, and so on) and integrated into various multimedia applications.
- Direct access, which makes nonlinear video editing simple
- Repeated recording without degradation of image quality
- Ease of encryption and better tolerance to channel noise

TABLE 5.2: Comparison of analog broadcast TV systems.

TV system	Frame rate (fps)	Number of scan lines	Total channel width (MHz)	Bandwidth allocation (MHz)		
				Y	*I* or *U*	*Q* or *V*
NTSC	29.97	525	6.0	4.2	1.6	0.6
PAL	25	625	8.0	5.5	1.8	1.8
SECAM	25	625	8.0	6.0	2.0	2.0

In earlier Sony or Panasonic recorders, digital video was in the form of composite video. Modem digital video generally uses component video, although RGB signals are first converted into a certain type of color opponent space, such as YUV. The usual color space is YCbCr.

Chroma Subsampling:

Since humans see color with much less spatial resolution than black and white, it make ssense to decimate the chrominance signal. Interesting but not necessarily infinitive name shave arisen to label the different schemes used. To begin with, numbers are given stating how many pixel values, per four original pixels, are actually sent. Thus the chroma sub samplings cheme "4:4:4" indicates that no chroma subsampling is used. Each pixel's Y, Cb, and Crvalues are transmitted, fourfor each of Y, Cb, and (Cr1,Y2)(Cr2, Y3)(Cb4, Y4), and so on.The scheme "4: 1: I" subsamples horizontally by a factor of 4. The scheme "4:2:0"subsamples in both the horizontal and vertical dimensions by a factor of 2. Theoretically, anaverage chroma pixel is positioned between the rows and columns, as shown.in Figure 5.6.We can see that the scheme 4:2:0 is in fact another kind of 4: 1: 1 sampling, in the sense thatwe send 4, 1, and 1 values per 4 pixels. Therefore, the labeling scheme is not a very reliablenmemomc!

CCIR Standards for Digital Video:

The CCIR is the Consultative Committee for Intemational Radio. One of the most important standards it has produced is CCIR-601, for component digital video (introduced in Section 4.3.4). This standard has since become standard ITU-R-601, an international standard for professional video applications. It is adopted by certain digital video fonnats, including the popular DV video.
The NTSC version has 525 scan lines, each having 858 pixels (with 720 of them visible, not in the blanking period). Because the NTSC version uses 4:2:2, each pixel can be represented with two bytes (S bits for Y and 8 bits alternating between Cb and Cr). The CCIR 601. (NTSC) data rate (including blanking and sync but excluding audio) is thus approximately 216 Mbps (megabits per second).
bits 525 x 858 x 30 x 2 bytes x 8-$\frac{bits}{byte}$ ~ 216 Mbps byte During blanking, digital video systems may make use of the extra data capacity to carry audio signals translations into foreign languages, or error-correction information. Table 5.3 shows some of the digital video specifications, all with an aspect ratio of 4:3. The CCIR 60 I standard uses an interlaced scan, so each field has only half as much vertical resolution (e.g., 240 lines in NTSC).

TABLE 5.3: Digital video specifications.

	CCIR 601 525/60 NTSC	CCIR 601 625/50 PAL/SECAM	CIF	QCIF
Luminance resolution	720 × 480	720 × 576	352 × 288	176 × 144
Chrominance resolution	360 × 480	360 × 576	176 × 144	88 × 72
Color subsampling	4:2:2	4:2:2	4:2:0	4:2:0
Aspect ratio	4:3	4:3	4:3	4:3
Fields/sec	60	50	30	30
Interlaced	Yes	Yes	No	No

CIF stands for Common Imermediate Format, specified by the International Telegraph and Telephone Consultative Committee (CCITT), now superseded by the International Telecommunication Union, which oversees both telecommunications (ITU-T) and radio: ',) r \\ frequency matters (ITU-R) under one United Nations body. The idea of CIF, which is about\\,." t,! .,/') the same as VHS quality, is to specify a format for lower bit rate. CIF uses a progressive or;:?
,\,~,-,Y.Y' (noninterlaced) scan. QCIF stands for Quarter-CIF, and is for even lower bit rate. All the J~(,s"J CIF/QCIF resolutions are evenly divisible by 8, and all except 88 are divisible by 16; this is convenient for block-based video coding in H.261 and H.263, discussed in Chapter 10.CIF is a compromise between NTSC and PAL, in that it adopts the NTSC frame rate and half the number of active lines in PAL. When played on existing TV sets, NTSC TV will first need to convert the number of lines, whereas PAL TV will require frame-rate conversion.

High Definition TV (HDTV)

The introduction of wide-screen movies brought the discovery that viewers seated near the screen enjoyed a level of participation (sensation of immersion) not experienced with conventional movies. Apparently the exposure to a greater field of view, especially the involvement of peripheral vision, contributes to the sense of "being there". The main thrust of High Definition TV (HDTV) is not to increase the "definition" in each unit area, but rather to increase the visual field, especially its width. First-generation HDTV was based on an analog technology developed by Sony and NHK in Japan in the late 1970s. HDTV successfully broadcast the 1984 Los Angeles Olympic Games in Japan. Multiple sub-Nyquist Sampling Encoding (MUSE) was an improved NHK HDTV with hybrid analog/digital technologies that was put in use in the 1990s. It has 1,125 scan lines, interlaced (60 fields per second), and a 16:9 aspect ratio. It uses satellite to broadcast ~ quite appropriate for Japan, which can be covered with one or two satellites. The Direct Broadcast Satellite (DBS) channels used have a bandwidth of 24 :MHz. In general, terrestrial broadcast, satellite broadcast, cable, and broadband networks are all feasible means for transmitting HDTV as well as conventional TV.

TABLE 5.4: Advanced Digital TV Formats Supported by ATSC.

Number of active pixels per line	Number of active lines	Aspect ratio	Picture rate
1,920	1,080	16:9	60I 30P 24P
1,280	720	16:9	60P 30P 24P
704	480	16:9 and 4:3	60I 60P 30P 24P
640	480	4:3	60I 60P 30P 24P

HDTV will easily demand more than 20 MHz bandwidth, which will not fit in the current6 MHz or 8 MHz channels, various compression techniques, are being investigated. It is also anticipated that high-quality HDTV signals will be transmitted using more than one channel, even after compression. In 1987, the FCC decided that HDTV standards must be compatible with the existing NTSC standard and must be confined to the existing Very High Frequency (VHF) and Ultrahigh Frequency (UHF) bands. This prompted a number of proposals in North America by the end of 1988, all of them analog or mixed analog/digital. In 1990, the FCC announced a different initiative ~ its preference for full-resolution HDTV. They decided that HDTV would be simultaneously broadcast with existing NTSC TV and eventually replace it. The development of digital HDTV immediately took off in North America.

Witnessing a boom of proposals for digital HDTV, the FCC made a key decision to go all digital in 1993. A "grand alliance" was formed that included four main proposals, by General Instruments, NUT, Zenith, and AT&T, and by Thomson, Philips, Sarnoff and others. This eventually led to the formation of the Advanced Television Systems Committee (ATSC), which was responsible for the standard for TV broadcasting of HDTV. In 1995,

The U.S. FCC Advisory Committee on Advanced Television Service recommended that the ATSC digital television standard be adopted.

The standard supports video scanning formats shown in Table 5.4. In the table, "1" means interlaced scan and "P" means progressive (noninterlaced) scan. The frame rates supported are both integer rates and the NTSC rates - that is, 60.00 or 59.94, 30.00 or 29.97,24.00 or 23.98 fps. For video, MPEG-2 is chosen as the compression standard. As will be seen in Chapter 11, it uses Main Level to High Level of the Main Profile of MPEG-2. For audio, AC-3 is the standard. It supports the so-called 5.1 channel Dolby surround sound - five surround Channels plus a subwoofer channel. The salient difference between conventional TV and HDTV [4, 6] is that the latter has a much wider aspect ratio of 16:9 instead of 4:3. (Actually, it works out to be exactly one-third wider than current TV.) The FCC has planned to replace all analog broadcast services with digital TV broadcasting by the year 2006. Consumers with analog TV sets will still be able to receive signals via an 8-VSB (8-level vestigial sideband) demodulation box. The services provided will include

- Standard Definition TV (SDTV) ~ the current NTSC TV or higher Enhanced Definition TV (EDTV) - 480 active lines or higher ~ the third and fourth rows in Table 5.4
- High Definition TV (HDTV) - 720 active lines or higher. So far, the popular choices are nop (720 lines, progressive, 30 fps) and 108m (1,080 lines, interlaced, 30 fps or 60 fields per second). The latter provides slightly better picture quality but requires much higher bandwidth.

DIGITIZATION OF SOUND

Sound is a wave phenomenon like light, but it is macroscopic and involves molecules of air being compressed and expanded under the action of some physical device..For example, a speaker in an audio system vibrates back and forth and produces a longitudinal pressure Wave that we perceive as sound. (As an example, we get a longitudinal wave by vibrating a Slinky along its length; in contrast, we get a transverse wave by waving the Slinky back and forth perpendicular to its length.)

Without air there is no sound - for example, in space. Since sound is a pressure wave, it takes on continuous values, as opposed to digitized ones with a finite range. Nevertheless, if we wish to use a digital version of sound waves, we must form digitized representations. of audio information.

Even though such pressure waves are longitudinal, they still have ordinary wave properties and behaviors, such as reflection (bouncing), refraction (change of angle when entering a medium with a different density), and diffraction (bending around an obstacle). This makes the design of "surround sound" possible.

Since sound consists of measurable pressures at any 3D point, we can detect it by measuring the pressure level at a location, using a transducer to convert pressure to voltage levels.

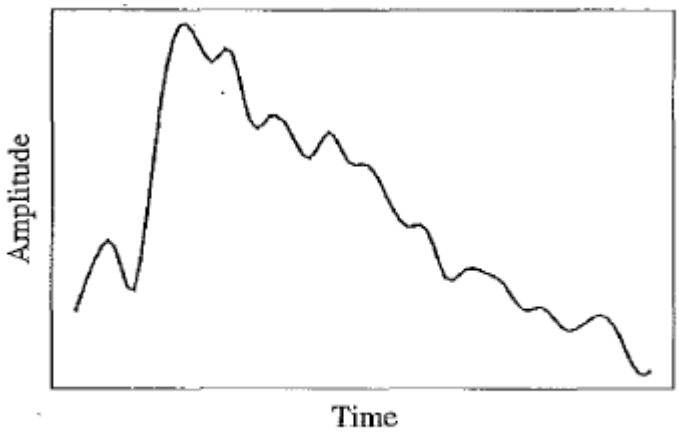


FIGURE 6.1: An analog signal: continuous measurement of pressure wave.

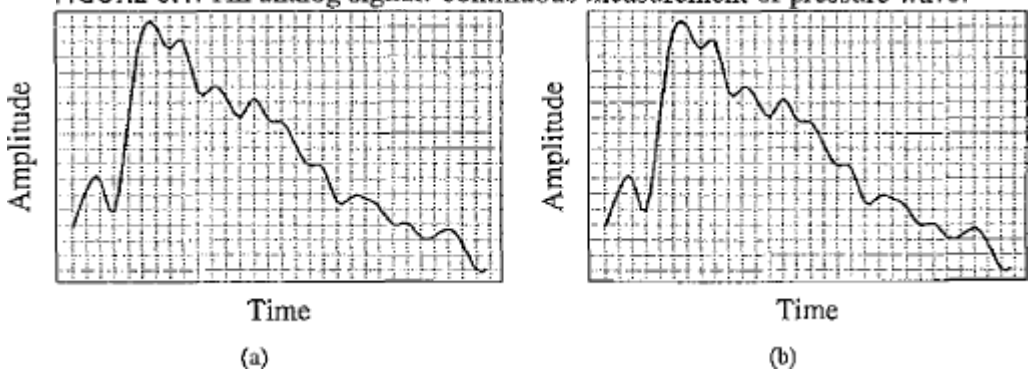


FIGURE 6.2: Sampling and quantization: (a) sampling the analog signal in the time dimension;
Quantization is sampling the analog signal in the amplitude dimension.

For audio, typical sampling rates are from 8 kHz (8,000 samples per second) to 48 kHz. The human ear can hear from about 20 Hz (a very deep rumble) to as much as 20 kHz; above this level, we enter the range of ultrasound. The human voice can reach approximately 4 kHz and we need to bound our sampling rate from below by at least double this frequency (see the discussion of the Nyquist sampling rate, below). Thus we arrive at the useful range about 8 to 40 or so kHz.

Sampling in the amplitude or voltage dimension is called quantization, shown in Figure 6.2(b). While we have discussed only unifoffil sampling, with equally spaced sampling intervals, nonuniform sampling is possible. This is not used for sampling in time but is used for quantization (see the ,u-Iaw rule, below). Typical uniform quantization rates are 8-bit and 16-bit; 8-bit quantization divides the vertical axis into 256 levels, and 16-bit divides it into 65,536 levels.

MIDI: MUSICAL INSTRUMENT DIGITAL INTERFACE:

Wave-table files provide an accurate rendering of real instrument sounds but are quite large. For simple music, we might be satisfied with PM synthesis versions of audio signals that could easily be generated by a sound card. A sound card is added to a PC expansion board and is capable of manipulating and outputting sounds through speakers connected to the board, recording sound input from a microphone connected to the computer, and manipulating sound stored on a disk. If we are willing to be satisfied with the sound card's defaults for many of the sounds we wish to include in a multimedia project, we can use a simple sclipting language and hardware setup called MIDI.

MIDI Overview:

MIDI, which dates from the early 1980s, is an acronym that stands for Musical Instrument Digital Interface. It forms a protocol adopted by the electronic music industry that enables computers, synthesizers, keyboards, and other musical devices to communicate with each other. A synthesizer produces synthetic music and is included on sound cards, using one of the two methods discussed above. The MIDI standard is supported by most synthesizers, so sounds created on one can be played and manipulated on another and sound reason ably close. Computers must have a special MIDI interface, but this is incorporated into most sound cards. The sound card must also have both DA and AD converters.
MIDI is a scripting language - it codes "events" that stand for the production of certain sounds. Therefore, MIDI files are generally very small. For example, a MIDI event might include values for the pitch of a single note, its duration, and its volume.
Terminology. A synthesizer was, and still may be, a stand-alone sound generator that can vary pitch, loudness, and tone color. (The pitch is the musical note the instrument plays - a C, as opposed to a G, say.) It can also change additional music characteristics, such as attack and delay time. A good (musician's) synthesizer often has a microprocessor, keyboard, control panels, memory, and so on. However, inexpensive synthesizers are now included on PC sound cards. Units that generate sound are referred to as tone modules or sound modules.
A sequencer started off as a special hardware device for storing and editing a sequence of musical events, in the form of MIDI data. Now it is more often a software music editor on the computer.
A MIDI keyboard produces no sound, instead generating sequences of MIDI instmctions, called MIDI messages. These are rather like assembler code and usually consist of just a few bytes. You might have 3 minutes of music, say, stored in only 3 kB. In comparison, a wave-table file (WAV) stores 1 minute of music in about 10MB. In MIDI parlance, the keyboard is referred to as a keyboard controller.
MIDI Concepts. Music is organized into tracks in a sequencer. Each track can be turned on or off on recording or playing back. Usually, a particular instrument is associated with a MIDI channel. MIDI channels are used to separate messages. There are 16 channels, numbered from 0 to 15. The channel forms the last four bits (the least significant bits) of the message. The idea is that each channel is associated with a particular instrument – for example, channel 1 is the piano, channel 10 is the drums. Nevertheless, you can switch instruments midstream, if desired, and associate another instrument with any charmel. The channel can also be used as a placeholder in a message. If the first four bits are all ones, the message is interpreted as a system common message. Along with channel messages (which include a channel number), several other types of messages are sent, such as a general message for all instruments indicating a change in tuning or timing; these are called system messages. It is also possible to send a special message to an instrument's channel that allows sending many notes without a channel specified. We will describe these messages in detail later. The way a synthetic musical instrument responds to a MIDI message is usually by simply ignoring any "plays sound" message that is not for its channel. If several messages are for its channel, say several simultaneous notes being played on a piano, then the instrument responds, provided it is multi-voice - that is, can play more than a single note at once.
It is easy to confuse the tenn voice with the term timbre. The latter is MIDI terminology for just what instrument we are trying to emulate - for example, a piano as opposed to a violin. It is the quality of the sound. An instrument (or sound card) that is multi-timbral is capable of playing many different sounds at the same time, (e.g., piano, brass, drums) On the other hand, the term "voice", while sometimes used by musicians to mean the same thing as timbre, is used in MIDI to mean every different timbre and pitch that the tone module can produce at the same time. Synthesizers can have many (typically 16,32, 64, 256, etc.) voices. Each voice works independently and simultaneously to produce sounds of different timbre and pitch. The tenn polyphony refers to the number of voices that can be produced at the same time. So a typical tone module may be able to produce "64 voices of polyphony" (64 different notes at once) and be "16-part multi- timbral" (can produce sounds like 16 different instruments at once).
FIGURE 6.8: Stream of IO-bit bytes; for typical MIDI messages, these consist of {status byte, data byte, data byte} = {Note On, Note Number, Note Velocity}. by a Note Off message (key release), which also has a pitch (which note to

turn off) and for consistency, one supposes - a velocity (often set to zero and ignored).
The data in a MIDI status byte is between 128 and 255; each of the data bytes is between o and 127. Actual MIDI bytes are 8 bit, plus a 0 start and stop bit, making them lO-bit "bytes". Figure 6.8 shows the MIDI data stream.

Hardware Aspects of MIDI:

The MIDI hardware setup consists of a 31.25 kbps (kilobits per second) serial connection, with the lO-bit bytes including a 0 starts and stop bit. Usually, MIDI-capable units are either input devices or output devices, not both.
Figure 6.10 shows a traditional synthesizer. The modulation wheel adds vibrato. Pitch bend alters the frequency, much like pulling a guitar string over slightly. There are often other controls, such as foots pedals, sliders, and so on.
The physical MIDI ports consist of 5-pin connectors labeled IN and OUT and a third connector, THRU. This last data channel simply copies data entering the IN channel. MIDI communication is half-duplex. MIDI IN is the connector via which the device receives all MIDI data. MIDI OUT is the connector through which the device transmits all the MIDI data it generates itself. MIDI THRU is the connector by which the device echoes the data it receives from MIDI IN (and only that - all the data generated by the device itself is sent via wilDI OUT).

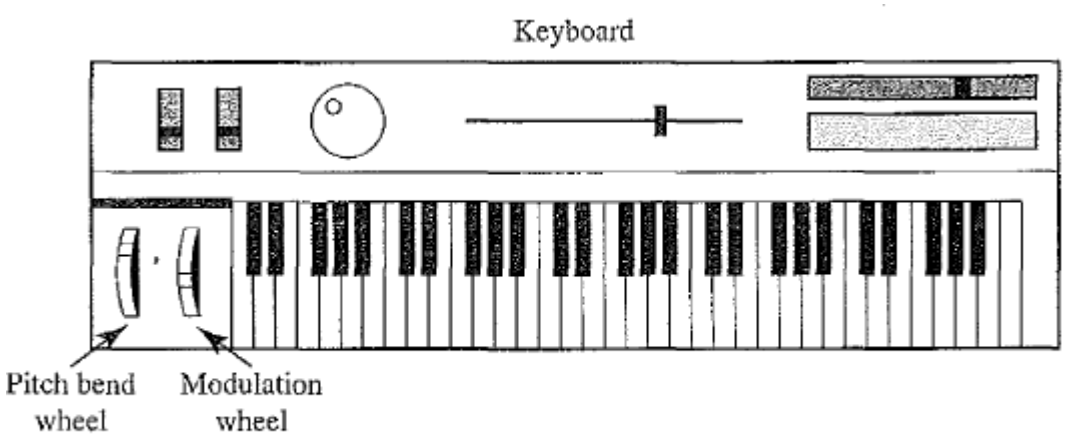


FIGURE 6.10: A MIDI synthesizer.

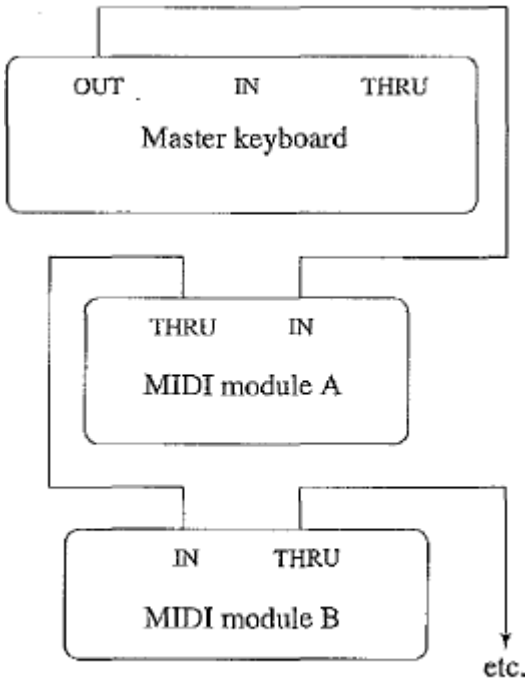


FIGURE 6.11: A typical MIDI setup.

Figure 6.11 shows a typical MIDI sequencer setup. Here, the MIDI OUT of the keyboard is connected to the 1vllDI IN of a synthesizer and then THRU to each of the additional sound modules. During recording, a keyboard-equipped synthesizer sends MIDI messages to a sequencer, which records them. During playback, messages are sent from the

sequencer to all the sound modules and the synthesizer, which play the music.

Structure of MIDI Messages:

MIDI messages can be classified into two types, as in Figure 6.12 ~ channel messages and system messages and further classified as shown. Each type of message will be examined below.

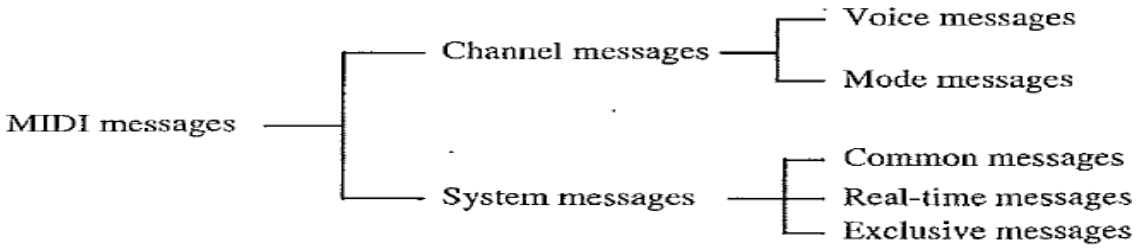


FIGURE 6.12: MIDI message taxonomy.

TABLE 6.3: MIDI voice messages

Voice message	Status byte	Data byte1	Data byte2
Note Off	&H8n	Key number	Note Off velocity
Note On	&H9n	Key number	Note On velocity
Polyphonic Key Pressure	&HAn	Key number	Amount
Control Change	&HBn	Controller number	Controller value
Program Change	&HCn	Program number	None
Channel Pressure	&HDn	Pressure value	None
Pitch Bend	&HEn	MSB	LSB

&H indicates hexadecimal, and *n* in the Status byte hex value stands for a channel number. All values are in 0 .. 127 except Controller number, which is in 0 .. 120.

Channel Messages:

A channel message can have up to 3 bytes; the first is the status byte (the opcode, as it were), and has its most significant bit set to 1. The four low-order bits identify which of the 16 possible channels this message belongs to, with the three remaining bits holding the message. For a data byte, the most significant bit is set to zero. Voice Messages. This type of channel message controls a voice - that is, sends information specifying which note to play or to tun off - and encodes key pressure. Voice messages are also used to specify controller effects, such as sustain, vibrato, tremolo, and the pitch wheel. Table 6.3 lists these operations.
For Note On and Note Off messages, the velocity is how quickly the key is played. Typically, a synthesizer responds to a higher velocity by making the note louder or brighter. Note On makes a note occur, and the synthesizer also attempts to make the note sound like the real instrument while the note is playing. Pressure messages can be used to alter the sound of notes while they are playing. The Channel Pressure message is a force measure for the keys on a specific channel (instrument) and has an identical effect on all notes playing on that channel. The other pressure message, Polyphonic Key Pressure (also called Key Pressure), specifies how much volume keys played together are to have and can be different for each note in a chord. Pressure is also called after touch.

The Control Change instmction sets various controllers (faders, vibrato, etc.). Each manufacturer may make use of different controller numbers for different tasks. However, controller 1 is likely the modulation wheel (for vibrato).
For example, a Note On message is followed by two bytes, one to identify the note and one to specify the velocity. Therefore, to play note number 80 with maximum velocity on channell3, the MIDI device would send the following three hex byte values: &H9C &H50 &H7F. (A hexadecimal number has a range 0 ..15. Since it is used to denote channels 1 to 16, "&HC" refers to channel 13). Notes are numbered such that middle C has number 60.
To play two notes simultaneously (effectively), first we would send a Program Change message for each of two channels. Recall that Program Change means to load a particular.

TABLE 6.4: MIDI mode messages

1st data byte	Description	Meaning of 2nd data byte
&H79	Reset all controllers	None; set to 0
&H7A	Local control	0 = off; 127 = on
&H7B	All notes off	None; set to 0
&H7C	Omni mode off	None; set to 0
&H7D	Omni mode on	None; set to 0
&H7E	Mono mode on (Poly mode off)	Controller number
&H7F	Poly mode on (Mono mode off)	None; set to 0

Patch for that channel. So far, we have attached two timbres to two different channels. Then sending two Note On messages (in serial) would turn on both channels. Alternatively, we could also send a Note On message for a particular channel and then another Note On message, with another pitch, before sending the Note Off message for the first note. Then we would be playing two notes effectively at the same time on the same instrument.
Polyphonic Pressure refers to how much force simultaneous notes have on several instruments. Channel Pressure refers to how much force a single note has on one instrument.

Channel Mode Messages:

Channel mode messages form a special case of the Control Change message, and therefore all mode messages have opcode B (so the message is "&HBn," or 101lnnnn). However, a Channel Mode message has its first data byte in 121 through 127 (&H79-7F).
Channel mode messages determine how an instrument processes MIDI voice messages. Some examples include respond to all messages, respond just to the correct channel, don't respond at all, or go over to local control of the instrument.
Recall that the status byte is "&HBn," where n is the channel. The data bytes have meanings as shown in Table 6.4. Local Control Off means that the keyboard should be disconnected from the synthesizer (and another, external, device will be used to control the sound). All Notes Off is a handy command, especially if, as sometimes happens, a bug arises such that a note is left playing inadvertently. Omlli means that devices respond to messages from all channels. The usual mode is OMNI OFF - pay attention to your own messages only, and do not respond to every message regardless of what channel it is on. Poly means a device will play back several notes at once if requested to do so. The usual mode is POLY ON.
In POLY OFF - monophonic mode - the argument that represents the number of monophonic channels can have a value of zero, in which case it defaults to the number of voices the receiver can play; or it may set to a specific number of channels. However, the exact meaning of the combination of OMNI ON/OFF and Mono/Poly depends on the specific combination, with four possibilities. Suffice it to say that the usual combination is ONINI OFF, POLY ON.

TABLE 6.5: MIDI System Common messages

System common message	Status byte	Number of data bytes
MIDI Timing Code	&HF1	1
Song Position Pointer	&HF2	2
Song Select	&HF3	1
Tune Request	&HF6	None
EOX (terminator)	&HF7	None

System Messages. System messages have no channel number and are meant for commands that are not channel-specific, such as timing signals for synchronization, positioning information in prerecorded MIDI sequences, and detailed setup information for the destination device. Opcodes for all system messages start with "&HF." System messages are divided into three classifications, according to their use.

System Common Messages. Table 6.5 sets out these messages, which relate to timing or positioning. Song Position is measured in beats. The messages determine what is to be played upon receipt of a "start" real-lime message (see below).

System Real-Time Messages. Table 6.6 sets out system real-time messages, which are related to synchronization.

System Exclusive Message. The final type of system message, System Exclusive messages, is included so that manufacturers can extend the IvIIDI standard. After the initial code, they can insert a stream of any specific messages that apply to their own product. A System Exclusive message is supposed to be terminated by a terminator byte "&HF7," as Specified in Table 6.5. However, the terminator is optional, and the data stream may simply be ended by sending the status byte of the next message.

TABLE 6.6: MIDI System Real-Time messages

System real-time message	Status byte
Timing Clock	&HF8
Start Sequence	&HFA
Continue Sequence	&HFB
Stop Sequence	&HFC
Active Sensing	&HFE
System Reset	&HFF

General MIDI

For MIDI music to sound more or less the same on every machine, we would at least like to have the same patch numbers associated with the same instruments - for example, patch I should always be a piano, not a ftugelhol1l. To this end, General MIDI [5] is a scheme for assigning instruments to patch numbers. A standard percussion map also specifies 47 percussion sounds. Where a "note" appears on a musical score detentions just what percussion element is being struck. This book's web site includes both the General MIDI Instrument Path Map and the Percussion Key map.

Other requirements for General MIDI compatibility are that a MIDI device must support all 16 channels; must be muIti- timbral (i.e., each channel can play a different instrument/ program); must be polyphonic (i.e., each channel is able to play many voices); and must have a minimum of 24 dynamically allocated voices.

General MIDI Leve12. An extended General MIDI has recently been defined, with a standard SMF Standard MIDI File format defined. A nice extension is the inclusion of extra character information, such as karaoke lyrics, which can be displayed on a good sequencer.

MIDI-to-WAV Conversion:

Some programs, such as early versions of Premiere, cannot include MIDI files - instead, they insist on WAY format files. Various shareware programs can approximate a reasonable conversion between these formats. The programs essentially consist of large lookup files that try to do a reasonable j~b of substituting predefined or shifted WAV output for some MIDI messages, with inconsistent success.

QUANTIZATION AND TRANSMISSION OF AUDIO:

To be transmitted, sampled audio information must be digitized, and here we look at some of the details of this process. Once the information has been quantized, it can then be transmitted or stored. We go through a few examples in complete detail, which helps in understanding what is being discussed.

Coding of Audio

Quantization and transformation of data are collectively known as coding of the data. For audio, the J.l-law technique for commanding audio signals is usually combined with a simple algorithm that exploits the temporal redundancy present in audio signals.

Pulse Code Modulation:

PCM in General. Audio is analog - the waves we hear travel through the air to reach our eardrums. We know that the basic techniques for creating digital signals from analog ones consist of sampling and quantization. Sampling is invariably done uniformly – we select a sampling rate and produce one value for each sampling time.
In the magnitude direction, we digitize by quantization, selecting breakpoints in magnitude and remapping any value within an interval to one representative output level. The set of interval boundaries is sometimes called decision boundaries, and the representative values are called reconstruction levels.
We say that the boundaries for quantizes input intervals that will all be mapped into the same output level form a coder mapping, and the representative values that are the output values from a quantizer are a decoder mapping. Since we quantize, we may choose to create either an accurate or less accurate representation of sound magnitude values. Finally, we may wish to compress the data, by assigning a bit stream that uses fewer bits for the most prevalent signal values.
Every compression scheme has three stages:
1. **Transformation**. The input data is transformed to a new representation that is easier or more efficient to compress. For example, in Predictive Coding, (discussed later in the chapter) we predict the next signal from previous ones and transmit the prediction error.
2, **Loss**. we may introduce loss of information. Quantization is the main lossy step. Here we use a limited number of reconstruction levels, fewer than in the original signal. Therefore, quantization necessitates some loss of information.
3. **Coding**. Here, we assign a codeword (thus forming a binary bit stream) to each output level or symbol. This could be a fixed-length code or a variable-length code, such as Huffman coding (discussed in Chapter 7).
For audio signals, we first consider PCM, the digitization method. That enables us to consider Lossless Predictive Coding as well as the DPCM scheme; these 'methods use differential coding. We also look at the adaptive version, ADPCM, which is meant to provide better compression. Pulse Code Modulation, is a formal term for the sampling and quantization we have already been using. Pulse comes from an engineer's point of view that the resulting digital signals can be thought of as infinitely narrow vertical "pulses". As an example of PCM, audio samples on a CD might be sampled at a rate of 44.1 kHz, with 16 bits per sample. For stereo sound, with two channels, this amount to a data rate of about 1,400 kbps.

PCM in Speech Compression.

Recall that in Section 6.1.6 we considered companding:the so-called compressor and expander stages for speech signal processing, for telephony. For this application, ‘signals are first transformed using the j.t-law (or A-law for Europe) rule into what is essentially a logarithmic scale. Only then is PCM, using uniform quantization, applied. The result is that finer increments in sound volume are used at the low-volume end of speech rather than at the high-volume end, where we can't discern small changes in any event.
Assuming a bandwidth for speech from about 50 Hz to about 10 kHz, the Nyquist rate would dictate a sampling rate of20 kHz. Using uniform quantization without companding, the minimum sample size we could get away with would likely be about 12 bits. Hence, for mono speech transmission the bit rate w9uld be 240 kbps. With companding, we can safely reduce the sample size to 8 bits with the same perceived level of quality and thus reduce the bit rate to 160 kbps. However, the standard approach to telephony assumes that the highest-frequency audio signal we want to reproduce is about 4 kHz. Therefore, the sampling rate is only 8 kHz, and the companded bit rate thus reduces to only 64 kbps.
We must also address two small wrinkles to get this comparatively simple form of speech compression right. Filst because only sounds up to 4 kHz are to be considered, all other frequency content must be noise. Therefore, we should remove this high-frequency content from the analog input signal This is done using a band-limiting filter that blocks out high frequencies as well as very low ones. The "band" of not-removed ("passed") frequencies are what we wish to keep. This type of filter is therefore also called a band-pass filter.
Second, once we arrive at a pulse signal, such as the one in Figure 6.13(a), we must still perform l digital-ta-analog conversion and then construct an output analog signal. But

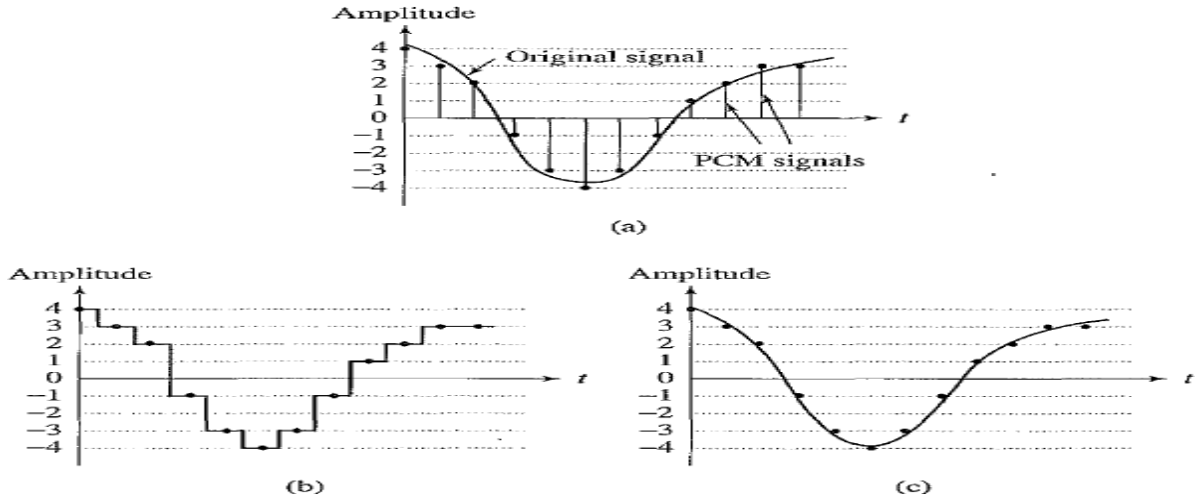


FIGURE 6.13: Pulse code modulation (PCM): (a) original analog signal and its corresponding PCM signals; (b) decoded staircase signal; (c) reconstructed signal after low-pass filtering.

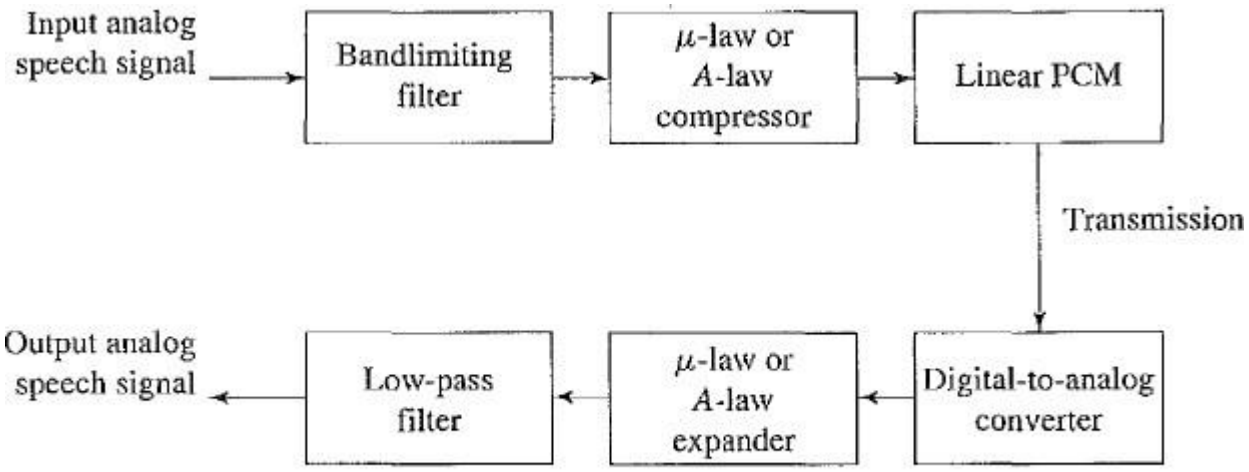


FIGURE 6.14: PCM signal encoding and decoding.

The signal we arrive at is effectively the staircase shown in Figure 6.13(b). This type of discontinuous signal contains not just frequency components due to the original signal but, because of the sharp comers, also a theoretically infinite set of higher-frequency components (From the theory of Fourier analysis, in signal processing). We know these higher frequencies are extraneous. Therefore, the output of the digital-to-analog converter is in turn passed to a low-pass filter, which allows only frequencies up to the original maximum to be retained. Figure 6.14 shows the complete scheme for encoding and decoding telephony signals as a schematic. As a result of the low-pass filtering, the output becomes smoothed, as Figure 6.13(c) shows. For simplicity, Figure 6.13 does not show the effect of companding.
A-law or jL-law PCM coding is used in the older International Telegraph and Telephone Consultative Committee (CCITT) standard G.711, for digital telephony. This CCITT standard is now subsumed into standards promulgated by a newer organization, the International Telecommunication Union (lTD).

Differential Coding of Audio

Audio is often stored not in simple PCM but in a form that exploits differences. For a start, differences will generally be smaller numbers and hence offer the possibility of using fewer bits to store.
An advantage of forming differences is that the histogram of a difference signal is usually considerably more peaked than the histogram for the original signal. For example, as an extreme case, the histogram for a linear ramp signal that has constant slope is uniform, whereas the histogram for the derivative of the signal (i.e., the differences, from sampling point to sampling point) consists of a spike at the slope value. Generally, if a time-dependent signal has some consistency over time (temporal redundancy), the different signal - subtracting the current sample from the previous one will have a more

peaked histogram, with a maximum around zero. Consequently, if we then go on to assign bit string code words to differences; we can assign short codes to prevalent values and long code words to rarely occurring ones.
To begin with, consider a lossless version of this scheme. Loss arises when we quantize. If we apply no quantization, we can still have compression - via the decrease in the variance of values that occurs in differences, compared to the original signal. Chapter 7 introduces more sophisticated versions of lossless cOffip, ression methods, but it helps to see a simple version here as well. With quantization, Predictive Coding becomes DPCM, a lossy method; we'll also tryout that scheme.

Lossless Predictive Coding

Predictive coding simply means transmitting differences - we predict the next sample as being equal to the current sample and send not the sample itself but the error involved in making this assumption. That is, if we predict that the next sample equals the previous one, then the error is just the difference between previous and next. Our prediction scheme could also be more complex.
However, we do note one problem. Suppose our integer sample values are in the range 0 .. 255. Then differences could be as much as -255 .. 255. So we have unfortunately increased our dynamic range (ratio of maximum to minimum) by a factor of two: we may well need more bits than we needed before to transmit some differences. Fortunately, we can use a trick to get around this problem, as we shall see.
So, basically, predictive coding consists of finding differences and transmitting them, using a PCM system such as the one introduced in Section 6.3.2. First, note that differences of integers will at least be integers. Let's formalize our statement of what we are doing by defining the integer signal as the set of values 1''. Then we predict values ill as simply the previous value, and we define the error ell as the difference between the actual and predicted signals:

$$\hat{f}_n = f_{n-1}$$
$$e_n = f_n - \hat{f}_n$$

We certainly would like our error value en to be as small as possible. Therefore, we would wish our prediction ill to be as close as possible to the actual signal 1,1' But for a particular sequence of signal values, some fUllction of a few of the previous values, In-1, 1,1-2, 1,1-3, etc., may provide a better prediction of 1,1' Typically, a linear predictor function is used:

$$\hat{f}_n = \sum_{k=1}^{2 \text{ to } 4} a_{n-k} f_{n-k}$$

Such a predictor can be followed by a truncating or rounding operation to result in integer values. In fact, since now we have such coefficients Gn-k available, we can even change them adaptively (see Section 6.3.7 below).
The idea of forming differences is t9 make the histogram of sample values more peaked. For example, Figure 6.15(a) plots 1 second of sampled speech at 8 kHz, with magnitude resolution of 8 bits per sample.
A histogram of these values is centered around zero, as in Figure 6.I5(b). Figure 6.l5(c) shows the histogram for corresponding speech signal differences: difference values are much more clustered around zero than are sample values themselves. As a result, a method that

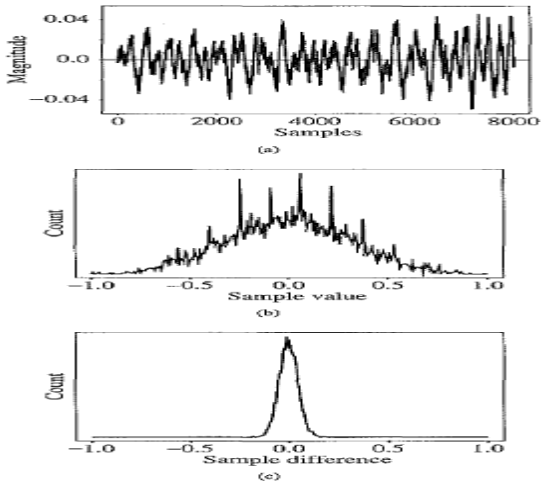


FIGURE 6.15: Differencing concentrates the histogram: (a) digital speech signal; (b) histogram of digital speech signal values; (c) histogram of digital speech signal differences.

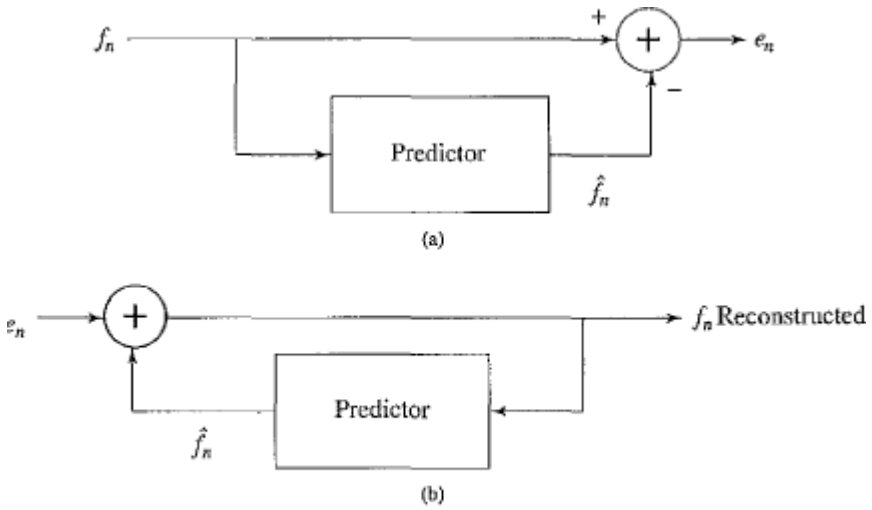


FIGURE 6.16: Schematic diagram for Predictive Coding: (a) encoder; (b) decoder.

DPCM

Differential Pulse Code Modulation is exactly the same as Predictive Coding, except that it incorporates a quantizer step. Quantization is as in PCM and can be uniform or nonuniform.One scheme for analytically determining the best set of nonuniform quantizer steps is the Lloyd-Max quantizer, named for Stuart Lloyd and Joel Max, which is based on a leasts queries minimization of the error term.

Here we should adopt some nomenclature for signal values. We shall call the original signal In, the predicted signal In, and the quantized, reconstmcted signal h. How DPCM operates is to form the prediction, form an error ell by subtracting the prediction from the actual signal, then quantize the error to a quantized version, en, The equations that describe DPCM are as follows:

$$
\begin{aligned}
\hat{f}_n &= \textit{function\_of}\,(\tilde{f}_{n-1}, \tilde{f}_{n-2}, \tilde{f}_{n-3}, \ldots) \\
e_n &= f_n - \hat{f}_n \\
\tilde{e}_n &= Q[e_n] \\
&\text{transmit } codeword(\tilde{e}_n) \\
&\text{reconstruct } \tilde{f}_n = \hat{f}_n + \tilde{e}_n
\end{aligned}
$$

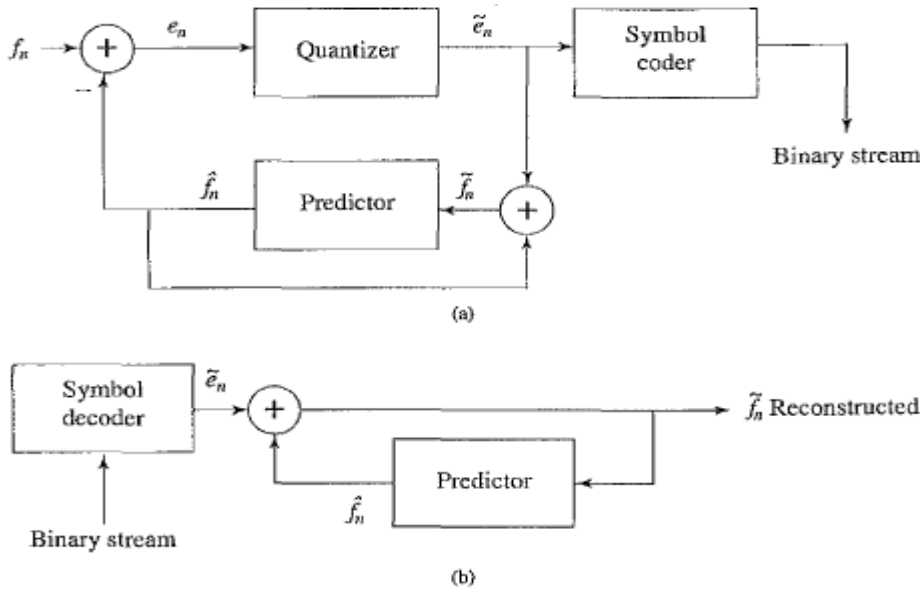


FIGURE 6.17: Schematic diagram for DPCM: (a) encoder; (b) decoder.

Multimedia Data Compression

The emergence of multimedia technologies has made digital libraries a reality. Nowadays, libraries, museums, film studios, and governments are converting more and more data and archives into digital form. Some of the data (e.g., precious books and paintings) indeed need to be stored without any loss.

As a start, suppose we want to encode the call numbers of the 120 million or so items in the Library of Congress (a mere 20 million, if we consider just books). Why don't we just transmit each item as a 27-bit number, giving each item a unique binary code (since227 > 120,000, OOO)?

The main problem is that this "great idea" requires too many bits. And in fact there exist many coding techniques that will effectively reduce the total number of bits needed to represent the above information. The process involved is generally referred to as compression [1,2J].

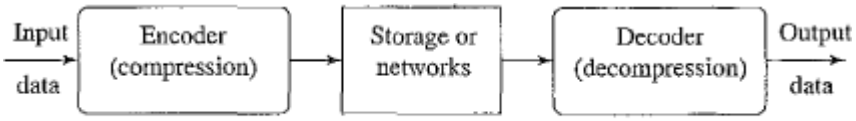


FIGURE 7.1: A general data compression scheme.

If the total number of bits required to represent the data before compression is Eo and the total number of bits require

$$compression\ ratio = \frac{B_0}{B_1}$$

represent the data before compression is Eo and the total number of bits required to represent the data after compression is 81, then we define the compression ratio as (7.1)In general, we would desire any codec (encoder/decoder scheme) to have a compression ratio much larger than 1.0. The higher the compression ratio, the better the lossless compression scheme, as long as it is computationally feasible.

BASICS OF INFORMATION THEORY

According to the famous scientist Claude E. Shannon, of Bell Labs [3, 4J, the entropy 1) of an information source with alphabet S = {SI, S2, ••. ,511 } is defined as:

$$\eta = H(S) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i}$$

$$= -\sum_{i=1}^{n} p_i \log_2 p_i$$

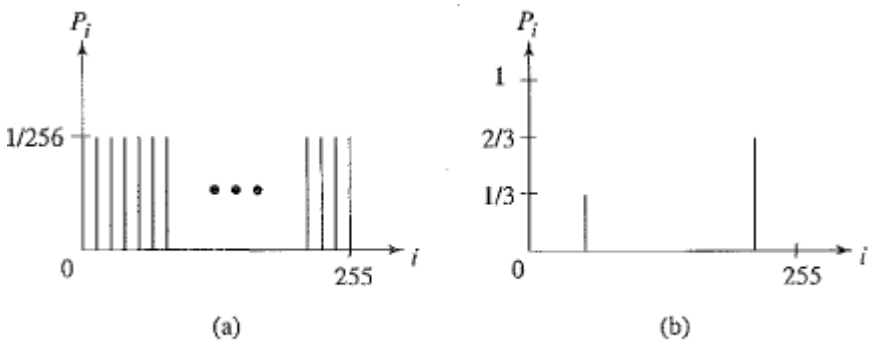


FIGURE 7.2: Histograms for two gray-level images.

Figure 7.2(a) shows the histogram of an image with Ill1iform distribution of gray-level intensities, - that is, Vi Pi = 1/256. Hence, the entropy of this image is

$$\eta = \sum_{i=0}^{255} \frac{1}{256} \cdot \log_2 256 = 8$$

As can be seen in Eq. (7.3), the entropy 17 is a weighted sum of tenns log2 ~i ; hence it represents the average amount of information contained per symbol in the source S. For a memory less source2 S, the entropy 1) represents the minimum average number of bits required to represent each symbol in S. In other words, it specifies the lower bound for the average number of bits to code each symbol in S.
If we use I to denote the average length (measured in bits) of the code words produced by the encoder, the Shannon Coding Theorem states that the entropy is the best we can do (under certain conditions):
Coding schemes aim to get as close as possible to this theoretical lower bound. It is interesting to observe that in the above uniform-distribution example we found that 1) = 8 - the minimum average number of bits to represent each gray-level intensity is at least 8. No compression is possible for this image! In the context of imaging, this will correspond to the "worst case," where neighboring pixel values have no similarity. Figure 7.2(b) shows the histogram of another image, in which 1/3 of the pixels are rather dark and 2/3 of them are rather bright. The entropy of this image is

$$\begin{aligned} \eta &= \frac{1}{3}\cdot\log_2 3 + \frac{2}{3}\cdot\log_2\frac{3}{2} \\ &= 0.33\times 1.59 + 0.67\times 0.59 = 0.52 + 0.40 = 0.92 \end{aligned}$$

In general, the entropy is greater when the probability distribution is flat and smaller when it is more peaked.

RUN-LENGTH CODING

Instead of assuming a memory less source, run-length coding (RLC) exploits memory present in the information sour<;e. It is one of the simplest fonns of data compression. The basic idea is that if the information source we wish to compress has the property that symbols tend to form continuous groups, instead of coding each symbol in the group individually, we can code one such symbol and the length of the group.
As an example, consider a bilevel image (one with only I-bit black and white pixels)with monotone regions. This information source can be efficiently coded using run-length coding. In fact, since there are only two symbols, we do not even need to code any symbol at the start of eachrun. Instead, we can assume that the starting run is always of a particular color (either black or white) and simply code the length of each run.
The above description is the one-dimensional run-length coding algorithm. A twodimensionalvariant of it is usually used to code bilevel images. This algorithm uses thecoded run information in the previous row of the image to code the run in the CUiTent row.A full description of this algorithm can be found in [5].

VARIABLE·LENGTH CODING (VLC)

Since the entropy indicates the information content in an information source S, it leads to a family of coding methods commonly known as entropy coding methods. As described earlier, variable-length coding (VLC) is one of the best-known such methods. Here, we will study the Shannon~Fano algorithm, Huffman coding, and adaptive Huffman coding.

Shannon-Fano Algorithm

The Shannon-Fano algorithm was independently developed by Shannon at Bell Labs andRobert Fana at MIT [6]. To illustrate the algorithm, let's suppose the symbols to be codedare the characters in the word HELLO. The frequency count of the symbols is

Symbol	H	E	L	O
Count	1	1	2	1

The encoding steps of the Shannon-Fano algorithm can be presented in the following top-down manner:
1. Sort the symbols according to t~e frequency count of their occurrences.
2. Recursively divides the symbols into two parts, each with approximately the same number of counts, until an parts contain only one symbol.
A natural way of implementing the above procedure is to build a binary tree. As a convention, let's assign bit 0 to its left branches and 1 to the right branches.

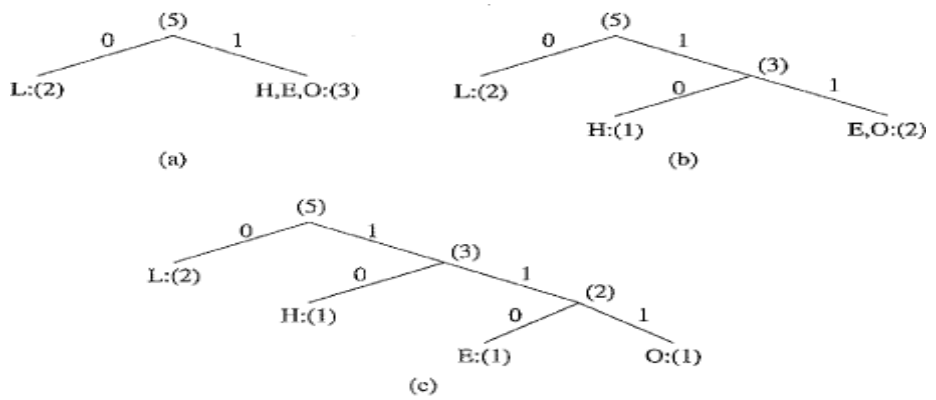


FIGURE 7.3: Coding tree for HELLO by the Shannon–Fano algorithm.

Initially, the symbols are sorted as LHEO. As Figure 7.3 shows, the first division yields two parts: (a) L with a count of2, denoted as L:(2); and (b) H, E and 0 with a total count of 3, denoted as H,E,0:(3). The second division yields H:(l) and E,O:(2). The last division is E:(1) and 0:(1).

Table 7.1 summarizes the result, showing each symbol, its frequency count, information content (log2 .;;), resulting codeword, and the number ofbits needed to encode each symbol in the word HELLO. The total number of bits used is shown at the bottom. To revisit the previous discussion on entropy, in this case,

$$\begin{aligned}\eta &= p_L \cdot \log_2 \frac{1}{p_L} + p_H \cdot \log_2 \frac{1}{p_H} + p_E \cdot \log_2 \frac{1}{p_E} + p_O \cdot \log_2 \frac{1}{p_O} \\ &= 0.4 \times 1.32 + 0.2 \times 2.32 + 0.2 \times 2.32 + 0.2 \times 2.32 = 1.92\end{aligned}$$

TABLE 7.1: One result of performing the Shannon–Fano algorithm on HELLO.

Symbol	Count	$\log_2 \frac{1}{p_i}$	Code	Number of bits used
L	2	1.32	0	2
H	1	2.32	10	2
E	1	2.32	110	3
O	1	2.32	111	3
TOTAL number of bits:				10

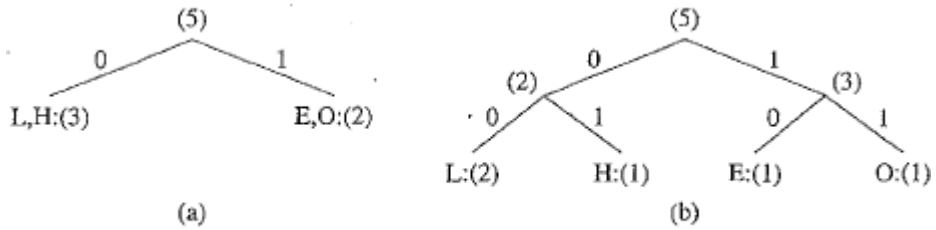


FIGURE 7.4: Another coding tree for HELLO by the Shannon–Fano algorithm.

This suggests that the minimum average number of bits to code each character in the word HELLO would be at least 1.92. In this example,. the Shannon-Fano algorithm uses an average of 10/5 = 2 bits to code each symbol, which is fairly close to the lower bound of1.92. Apparently, the result is satisfactory.

It should be pointed out that the outcome of the Shannon-Fano algorithm is not necessarily unique. For instance, at the first division in the above example, it would be equally valid to divide into the two parts L,H:(3) and E,O:(2). This would result in the coding in Figure 7.4. Table 7.2 shows the code words are different now. Also, these two sets of code words may behave differently when errors are present. Coincidentally, the total number of bits required to encode the world HELLO remains at 10.

The Shannon~Fano algorithm delivers satisfactory coding results for data compression, but it was soon outperformed and overtaken by the Huffman coding method.

Huffman Coding

First presented by David A. Huffman in a 1952 paper [7], this method attracted an overwhelming amount of research and has been adopted in many important and/or commercial applications, such as fax machines, JPEG, and MPEG.
In contradistinction to Shannon-Fano, which is top-down, the encoding steps of the Huffman algorithm are described in the following bottom-up manner. Let's use the same example word, HELLO. A similar binary coding tree will be used as above, in which the left branches are coded 0 and right branches 1. A simple list data structure is also used.

TABLE 7.2: Another result of performing the Shannon–Fano algorithm on HELLO.

Symbol	Count	$\log_2 \frac{1}{p_i}$	Code	Number of bits used
L	2	1.32	00	4
H	1	2.32	01	2
E	1	2.32	10	2
O	1	2.32	11	2
TOTAL number of bits:				10

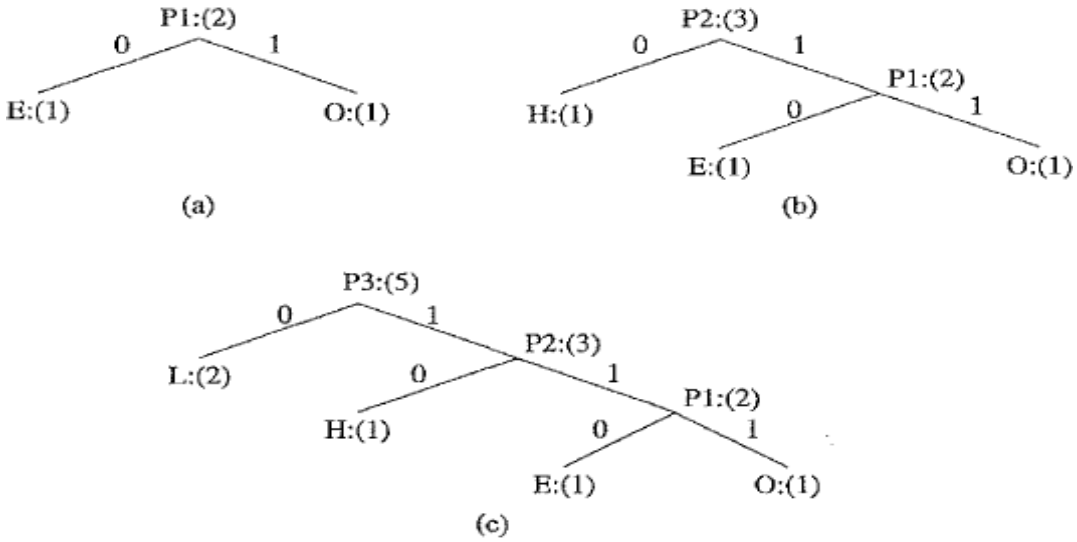


FIGURE 7.5: Coding tree for HELLO using the Huffman algorithm.

ALGORITHM 7.1 HUFFMAN CODING

1. Initialization; put all symbols on the list sorted according to their frequency counts.
2. Repeat until the list has only one symbol left.
(a) From the list, pick two symbols with the lowest frequency counts. Form a Huffman subtree that has these two symbols as child nodes and create a parent node for them.
(b) Assign the sum of the children's frequency counts to the parent and insert it into the list, such that the order is maintained.
(c) Delete the children from the list.
3. Assign a codeword for each leaf based on the path from the root.

In the above figure, new symbols PI, Pl, P3 are created to refer to the parent nodes in the Huffman coding tree. The contents in the list are illustrated below:
After initialization: LHEO After iteration (a): LPI H After iteration (b): LP2 After iteration (c): P3

The following are important properties of Huffman coding:

Unique prefix property.

No Huffman code is a prefix of any other Huffman code. For instance, the code 0 assigned to L in Figure 7.5(c) is not a prefix of the code 10 for H or 110 for E or 111 for 0; nor is the code 10 for H a prefix of the code 110 for E or 111 for O. It turns out that the unique prefix property is guaranteed by the above Huffman algorithm, since it always places all input symbols at the leaf nodes of the Huffman tree. The Huffman code is one of the prefix codes for which the unique prefix property holds. The code generated by the Shannon-Fano algorithm is another such example.
This property is essential and also makes for an efficient decoder, since it precludes any ambiguity in decoding. In the above example, if a bit 0 is received, the decoder can immediately produce a symbol L without waiting for any more bits to be transmitted.

Optimality.

The Huffman code is a minimum-redundancy code, as shown in Huffman's 1952 paper [7]. It has been proven 18, 2] that the Huffman code is optimal for a given data model (i.e., a given, accurate, probability distribution):
- The two least frequent symbols will have the same length for their Huffman codes, differing only at the last bit. This should be obvious from the above algorithm.

Extended Huffman Coding.

The discussion of Huffman coding so far assigns each symbol a codeword that has an integer bit length. As stated earlier, log2 ~ indicates the amount of information contained in the information source Si, which cori'e spoons to the number of bits needed to represent it. When a particular symbol Si has a large probability (close to 1.0), log2 -
/;; will be close to 0, and assigning one bit to represent that symbol will be costly. Only when the probabilities of all symbols can be expressed as 2-k, where is a positive ~integer, would the average length of code words be truly optimal- that is, I ~ ry. Clearly, I > ry in most cases One way to address the problem of integral codeword length is to group several symbols and assign a single codeword to the group. Huffman coding of this type is called Extended Huffman Coding [2]. Assume an information source has alphabet S = {S!, S2, ..• , Sit}. If k symbols are grouped together, then the extended alphabet is

$$S^{(k)} = \{\overbrace{s_1 s_1 \ldots s_1}^{k\ symbols}, s_1 s_1 \ldots s_2, \ldots, s_1 s_1 \ldots s_n, s_1 s_1 \ldots s_2 s_1, \ldots, s_n s_n \ldots s_n\}$$

Note that the size of the new alphabet S(k) is nk . If k is relatively large (e.g., k ::: 3), then for most practical applications where lZ » I,ll would be a very large number, implying a huge symbol table. This overhead makes Extended Huffman Coding impractical. As shown in [2], if the entropy of S is 11, then the average number of bits needed for each symbol in S is now

$$\eta \leq \bar{l} < \eta + \frac{1}{k}$$

Adaptive Huffman Coding

The Huffman algorithm requires prior statistical knowledge about the information source, and such information is often not available. This is particularly true in multimedia applications, where future data is unknown before its arrival, as for example in live (or streaming) audio and video. Even when the statistics are available, the transmission of the symbol table could represent heavy overhead. For the non-extended version of Huffman coding, the above discussion assumes a so called order-O model- that is, symbols/characters were treated singly, without any context or history maintained. One possible way to include contextual information is to examine k preceding (or succeeding) symbols each time; this is known as an order-k model For example, an order-l model can incorporate such statistics as the probability of "qu" in addition to the individual probabilities of "q" and "u". Nevertheless, this again implies that much more statistical data has to be stored and sent for the order-k model when k ::: 1.The solutio11 is to use adaptive compression algorithms, in which statistics are gathered and updated dynamically as the data stream arrives. The probabilities are no longer based on prior knowledge but on the actual data received so far. The new coding methods are "adaptive" because, as the probability distribution of the received symbols changes, symbols will be given new (longer or sh011er) codes. This is especially desirable for multimedia data, when the content (the music or the color of

the scene) and hence the statistics can change rapidly. As an example, we introduce the Adaptive Huffman Coding algorithm in this section. Many ideas, however, are also applicable to other adaptive compression algorithms.
lnitial_code assigns symbols with some initially agreed-upon codes, without any prior knowledge of the frequency counts for them. For example, some conventional code such as ASCII may be used for coding character symbols.
update_tree is a procedure for constructing an adaptive Huffman tree. It basically does two things: it increments the frequency counts for the symbols (including any new ones), and updates the configuration of the tree.

The Huffman tree must always maintain its sibling property ~ that is, all nodes(internal and leaf) are arranged in the order of increasing counts. Nodes are numbered in order from left to right, bottom to top. (See Figure 7.6, in which the first node is l.A:(l), the second node is 2.B:(1), and so on, where the numbers in parentheses indicates the count.) If the sibling property is about to be violated, a swap procedure is invoked to update the tree by rearranging the nodes.
When a swap is necessary, the fattest node with count N is swapped with the node whose count has just been increased to N + 1. Note that if the node with count N is not a leaf-node ~ it is the root of a subtree - the entire subtree will go with it during the swap.

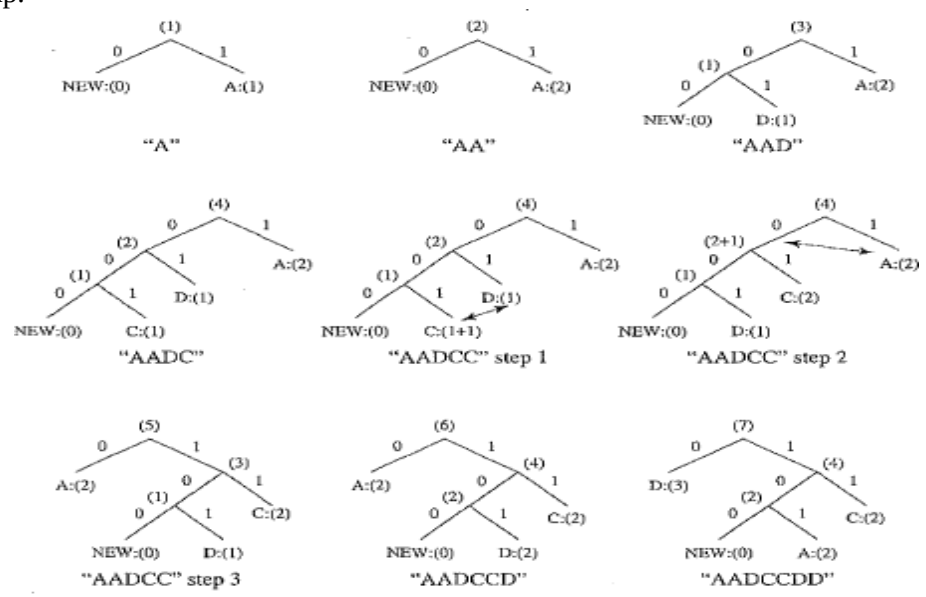


FIGURE 7.7: Adaptive Huffman tree for AADCCDD.

LOSS LESS IMAGE COMPRESSION

One of the most commonly used compression techniques in multimedia data compression is differential coding. The basis of data reduction in differential coding is the redundancy in consecutive symbols in a data stream. Recall that we considered lossless differential coding in Chapter 6, when we examined how audio must be dealt with via subtraction from predicted values. Audio is a signal indexed by one dimension, time. Here we consider how to apply the lessons learned from audio to the context of digital image signals ~hat are indexed by two, spatial, dimensions (x, y).

Lossless JPEG

Lossless IPEG is a special case of the JPEG image compression. It differs drastically from other IPEG modes in that the algorithm has no lossy steps. Thus we treat it here and consider the more used JPEG methods in Chapter 9. Lossless JPEG is invoked when the user selects a 100% quality factor in an image tool. Essentially, lossless IPEG is included in the JPEG compression standard simply for completeness. The following predictive method is applied on the unprocessed original image (or each color band of the original color image). It essentially involves two steps: forming a differential prediction and encoding.

DISTORTION MEASURES

A distortion measure is a mathematical quantity that specifies how close an approximation is to its originll1, using somejistortiQ.l1 crkeria.--when -looking· at -compressed data, it is natural to think of the distortion in terms of the numerical difference between the original data and the reconstructed data. However, when the data to be compressed is an image, such a measure may not yield the intended result.

For example, if the reconstructed image is the same as original image except that it is shifted to the right by one vertical scan line, an average human observer would have a hard time distinguishing it from the original and would therefore conclude that the distortion is small. However, when the calculation is carried out numerically, we find a large distortion, because of the large changes in individual pixels of the reconstructed image. The problem is that we need a measure of perceptual distortion, not a more naive numerical approach. However, the study of perceptual distortions is beyond the scope of this book. Of the many numerical distortion measures that have been defined, we present the three Most commonly used in image compression. If we are interested in the average pixel difference, the mean square error (MSE) 0'2 is often used. It is defined as

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2$$

Where Xn, Yn, and N are the input data sequence, reconstructed data sequence, and length of the data sequence, respectively. **If** we are interested in the size of the error relative to the signal, we can measure the signal-to noise ratio (SNR) by taking the ratio of the average square of the original data sequence and the mean square error (MSE), as discussed in Chapter 6. In decibel units (dB),it is defined as

$$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2}$$

Where a is the average square value of the original data sequence and aJ is the MSE. Another commonly used measure for distortion is the peak-signal-to-noise ratio (PSNR), which measures the size of the error relative to the peak value of the signal xpeak. It is given by

$$PSNR = 10\log_{10}\frac{x_{peak}^2}{\sigma_d^2}$$

THE RATE-DISTORTION THEORY

Lossy compression always involves a tradeoff between rate and distortion. Rate is the average number of bits required to represent each source symbol. Within this framework, the tradeoff between rate and distortion is represented in the form of a rate-distortion function R(D).

Intuitively, for a given source and a given distortion measure, if D is a tolerable amount of distortion, R(D) specifies the lowest rate at which the source data can be encoded while keeping the distortion bounded above by D. It is ea~ to se~t~'!!.~vl1~JLD-= .Q-NeJlave a lossless compression of the source. The rate-distortion function is meant to describe a fundamental limit for the performance of a coding algorithm and so can be used to evaluate the performance of different algorithms.

Figure 8.1 shows a typical rate-distortion function. Notice that the minimum possible rate at D = 0, no loss, is the entropy of the source data. The distortion corresponding to a rate R(D) ~ 0 is the maximum amount of distortion incurred when "nothing" is coded. Finding a closed-form analytic description of the rate-distortion function for a given source is difficult, if not impossible. Gyorgy [1] presents analytic expressions of the rate distortion function for various sources. For sources for which an analytic solution cannot be readily obtained, the rate-distortion function can be calculated numerically, using algorithms developed by Arimoto [2] and Blahut [3].

QUANTIZATION

Quantization in some form is the heart of any lossy scheme. Without quantization, we would indeed be losing little information. Here, we embark on a more detailed discussion of quantization. Each algorithm (each Quantizer) can be uniquelydeterrnined by its partition of the input range~on theenco<!ezii~~and the set of output values, on the decoder~ide. The input and Qutput –Lif of each quantizes can be either scalar values or vector values, thus leading to se aTar quantizes and vector quantizers. In this section, we examine the design of both uniform and non uniform scalar quantizes and briefly introduce the topic of vector quantization (VQ).

Uniform Scalar Quantization

A uniform scalar quantizer partitions the domain of input values into equally spaced intervals, except possibly at the two outer intervals. The endpoints of partition intervals are called the quantizer's decision boundaries. The output or reconstruction value corresponding to each interval is taken to be the midpoint of the interval. The length of each interval is referred to as the step size, denoted by the symbol 1:1. Uniform scalar quantizers are of two types: midrise and midtread. A midtread quantizer has zero as one of its output values, whereas the midrise quantizer has a partition interval that brackets zero (see Figure 8.2). A midrise quantizer is used with an even number of output levels, and a rnidtread quantizer with an odd number the performance of an M level quantizer. Let B = {hoI hi, ... , bM} be the set of decision boundaries and Y = (YI. Y2, ...,YM} be the set of reconstruction or output values. Suppose the input is uniformly distributed in the interval [-Xmax , Xmax].The rate of the quantizer is

$$R = \lceil \log_2 M \rceil$$

That is, R is the number of bits required to code M things in this case, the M Output levels. The step size /1 is given by

$$\Delta = \frac{2X_{max}}{M}$$

Since the entire'range of input values is from - Xmax to Xmax • For bounded input, the quantization error caused by the quantizer is referred to as. If the quantizer replaces a whole range of values, from a maximum value to co~ and similarly for negative values, that· part of the distortion is called the overload distortion.

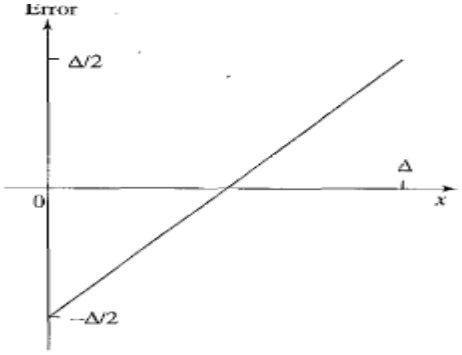


FIGURE 8.3: Quantization error of a uniformly distributed source.

$$D_{gran} = 2\sum_{i=1}^{\frac{M}{2}} \int_{(i-1)\Delta}^{i\Delta} \left(x - \frac{2i-1}{2}\Delta\right)^2 \frac{1}{2X_{max}}\, dx$$

No uniform Scalar Quantization

If the input source is not uniformly distributed, a uniform quantizer may be inefficient. Increasing the number of decision levels within the region where the source is densely distributed can effectively lower granular distortion. In addition, without having to increase the total number of decision levels, we can enlarge the region in which the source is sparsely distributed. Such no uniform quantizers thus have nonuniformly defined decision boundaries. There are two common approaches for non uniform quantization.

$$D_{gran} = \sum_{j=1}^{M} \int_{b_{j-1}}^{b_j} (x - y_j)^2 \frac{1}{X_{max}} f_X(x)\, dx$$

Companded Quantizer.

In companded quantization, the input is mapped by a compressor function G and then quantized using a uniform quantizer. After transmission, the quantized values are mapped back using an expander fimction G-1. The block diagram for the companding process is shown in Figure 8.4, where Xis the quantized version of X. If the input source is bounded by xmax , then any non uniform quantizer can be represented as a companded quantizer. The two commonly used companders are the
,u-law and A-law companders (Section 6.1).

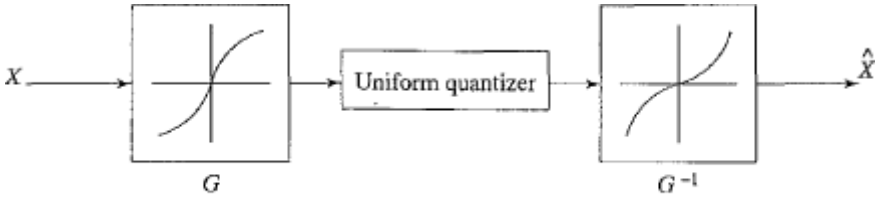


FIGURE 8.4: Companded quantization.

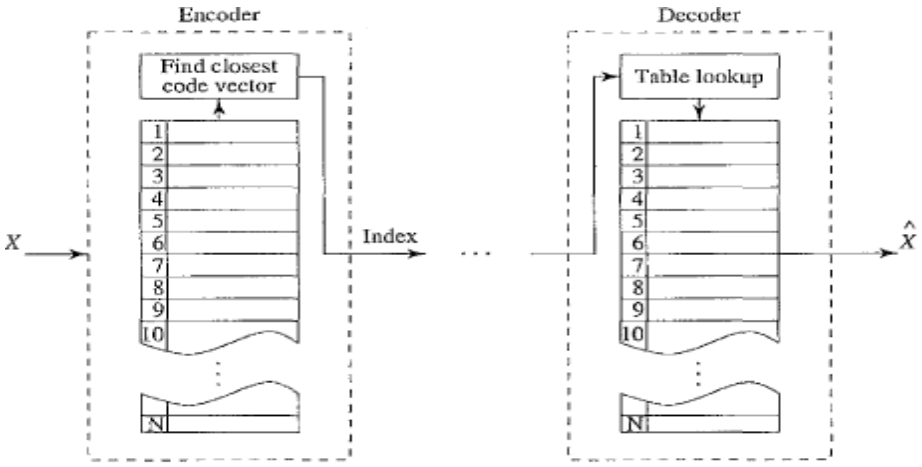


FIGURE 8.5: Basic vector quantization procedure.

Vector Quantization*

One of the fundamental ideas in Sharmon's original work on inf0l1llation theory is that any compression system performs better if it operates on vectors or groups of samples rather than on individual symbols or samples. We can form vectors of input samples by concatenating a number of consecutive samples into a single vector. For example, an input vector might be a segment of a speech sample, a group of consecutive pixels in an image, or a chunk of data in any other format.

The idea behind vector quantization (VQ) is similar to that of scalar quantization but extended into multiple dimensions. Instead of representing values within an interval in one-dimensional space by a reconstruction value, as in scalar quantization, in VQ an llcomponent code vector represents vectors that lie within a region in n-dimensional space. A collection of these code vectors forms the codebook for the vector quantizer.

Since there is no implicit ordering of code vectors, as there is in the one-dimensional case, an index set is also needed to index into the codebook. Figure 8.5 shows the basic vector quantization procedure. In the diagram, the encoder finds the closest code vector to the input vector and outputs the associated index. On the decoder side, exactly the same codebook is used. When the coded index of the input vector is received, a simple table lookup is performed to determine the reconstruction vector. such a way that the components of Yare much less correlated, then Ycan be coded For example, if most information in an RGB image is contained in a main axis, rotating so) that this direction is the first component means that luminance can be compressed differently from color information. This will approximate the luminance channel in the eye.

In higher dimensions than three, if most information is accurately described by the first few components of a transformed vector, the remaining components can be coarsely quantized, or even set to zero, with little signal distortion. The more decorrelated – that . is, the less effect one dimension has on another (the more orthogonal the axes), the more chance we have of dealing differently with the axes that store relatively minor amounts of information ,without affecting reasonably accurate reconstruction of the signal from its quantized or truncated transform coefficients.

Generally, the transform T itself does not cOrnpress any data. The compression comes f~pm th_e processing and quantization of the components of Y. In this section, we will study the Discrete Cosine Transform (DCT) as a tool to deco elate the input signal. We will also examine the Karhunen-Loeve Transfonn (KLT), which optimally decor relates the components of the input X.

JPEG Modes:

The JPEG standard supports numerous modes (variations). Some of the commonly used ones are:

- Sequential Mode
- Progressive Mode
- Hierarchical Mode
- Lossless Mode

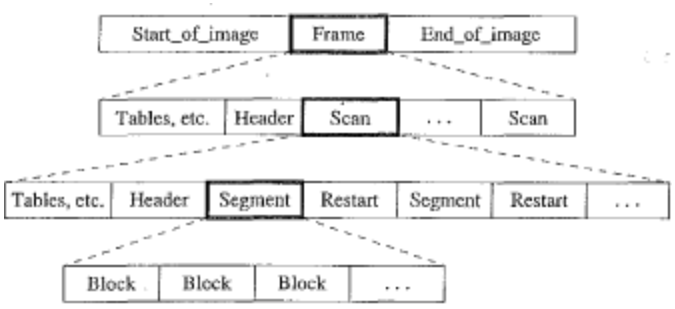


FIGURE 9.6: JPEG bitstream.

A Glance at the JPEG Bit stream

Figure 9.6 provides a hierarchical view of the organization of the bit stream for IPEG images. Here, a frame is a picture, a scan is a pass through the pixels (e.g., the red component), a segment is a group of blocks, and a block consists of 8 x 8 pixels. Examples of some header Information are:

- Frame header
  - Bits per pixel
  - (Width, height) of image
  - Number of components
  - Unique ill (for each component)
  - Horizontal/vertical sampling factors (for each component)
  - Quantization table to use (for each component)

Main Steps of JPEG2000 Image Compression

The main compression method used in JPEG2000 is the (Embedded Block Coding with Optimized Truncation) algorithm (EBCOn, designed by Taub man [5]. In addition to providing excellent compression efficiency, EBCOT produces a bit stream with a number of desirable features, including quality and resolution scalability and random access. The basic idea of EBCOT is the partition of each sub band LL, LB, HL, BB produced by the wavelet transform into small blocks called code blocks. Each code block is coded independently, in such a way that no information for any other block is used. A separate, scalable bit stream is generated for each code block. With its block-based coding scheme, the EBCOT algorithm has improved error resilience. The EBCOT algorithm consists of three steps:

1. Block coding and bit stream generation
2. Post compression rate distortion (PCRD) optimization

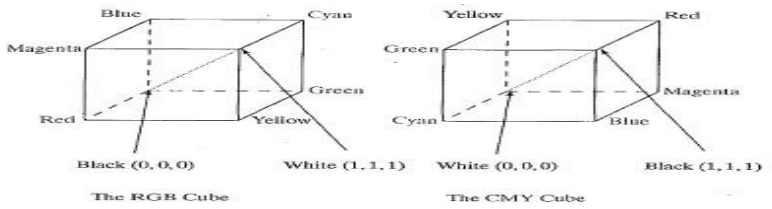


FIGURE 4.15: RGB and CMY color cubes.

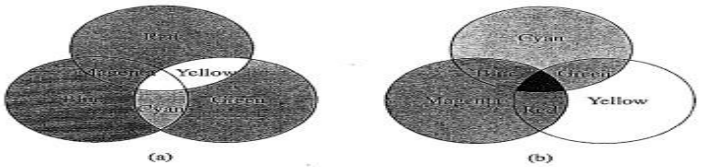


FIGURE 4.16: Additive and subtractive color: (a) RGB is used to specify additive color; (b) CMY is used to specify subtractive color.

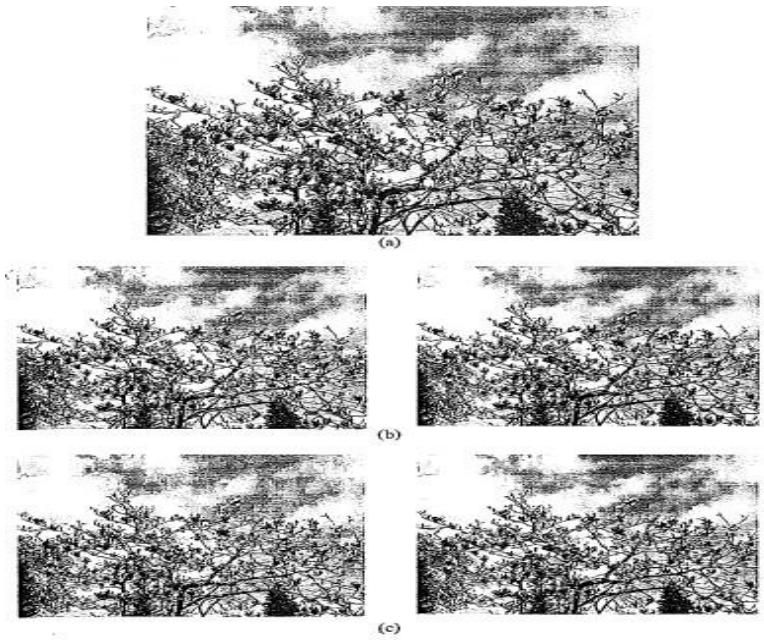


FIGURE 9.13: Comparison of JPEG and JPEG2000; (a) Original image; (b) JPEG (left) and JPEG2000 (right) images compressed at 0.75 bpp; (c) JPEG (left) and JPEG2000 (right) images compressed at 0.25 bpp.

UNIT-III

Basic Video compression techniques

MPEG-1:

The MPEG-1 audio/video digital compression standard was approved by the International Organization for Standardization International Electro technical Commission (ISOIIEC) NlPEG group in November 1991 for Coding of Audio for Digital Storage Media at up to about 1.5 Mbitls [5]. Common digital storage media include compact discs (CDs) and video compact discs (VCDs). Out of the specified 1.5 Mbps, 1.2 Mbps is intended for coded video, and 256 kbps can be used for stereo audio. This yields a picture quality comparable to VHS cassettes and a sound quality equal to CD audio. In general, MPEG-1 adopts the CCIR601 digital TV format, also known as Some Input Format (SIP). MPpG-1 supports only noninterna1 acedvi4eo. Normally, its picture resolution Is 352 X 240 video at 30 fps, or 352 x 288 for PAL video at 25 fps. It uses4:2:0 chroma sub sampling.

The MPEG-I standard, also referred to as ISOIIEC 11172 [5], has five parts: 11172-1 Systems, 11172-2 Video, 11172-3 Audio, 11172-4 Conformance, and 11172-5 Software. Briefly, Systems takes care of, among many things, dividing output into packets of bit streams, multiplexing, and synchronization of the video and audio streams. Conformance (or compliance) specifies the design of tests for verifying whether a bit stream or decoder complies with the standard. Software includes a complete software implementation of the MPEG-1 standard decoder and a sample software implementation of an encoder. We will

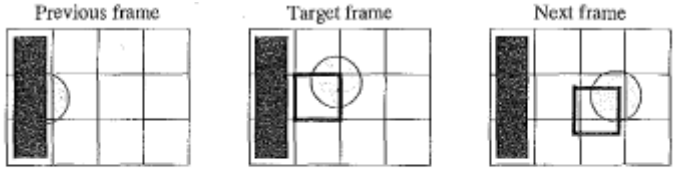


FIGURE 11.1: The need for bidirectional search.

examine the main features of MPEG-1 video coding and leave discussions of MPEG audio coding to Chapter 14.

Motion Compensation in MPEG·1

As discussed in the last chapter, motion-compensation-based video encoding in H.261 works as follows: In motion estimation, each macro block of the target P-frame is assigned a best matching macro block from the previously coded 1- or P-frame. This is called a prediction.

The difference between the macro block and its matching macro block is the prediction error, which is sent to DCT and its subsequent encoding steps. Since the prediction is from a previous frame, it is called fonvard prediction. Due to unexpected movements and occlusions in real scenes, the target macro block may not have, the previous frame. Figure 11.1 illustrated that the macro block containing part of a ball in the target frame cannot find a good matching macro block in the previous frame, because half of the ball was occluded by another object. However, a match can readily be obtained from the next frame. iVlPEG introduces a third frame type - B-frames - and their accompanying bidirectional I potion cQlnpensation. Figure 11.2 illustrates the motion-compensation-based B-frame coding idea. In addition to the forward prediction, a backward prediction is also performed, in which the matching macro block is obtained from a future I- or P-frame in the video sequence. Consequently, each macro block from a B-frame will specify up to two motion vectors, one from the f(lny'ard~l!d one from the backward prediction. If matching in both directions is successful, two motion vectors will be sent, and the two cor'rl:: sponding matching macro blocks are averaged (indicated by "%" in the figure) before co-riip-anng to the target macro block for generating the prediction error. If an acceptable.

TABLE 11.2: Default quantization table (Q_1) for intra-coding.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

For example, the slices can have different scale factors in the quantizer. This provides additional flexibility in bit rate control. Quantization. MPEG-l quantization uses different quantization tables for its intra and inter-coding (Tables 11.2 and 11.3). The quantizer numbers for intra-coding (Table 11.2) vary within a macro block. quantizer numbers for AC coefficknts are constant within a macro block.

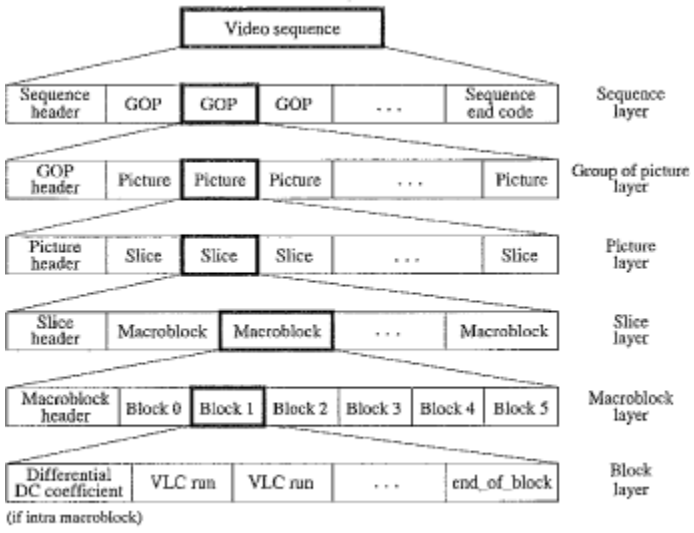


FIGURE 11.5: Layers of MPEG-1 video bitstream.

MPEG-1 Video Bit Stream:

Sequence layer. A video sequence consists of one or more group's ofpictures (Gaps): It always starts with a sequence header. The header contains infOImation about the picture, such as horizontal-size and vertical..size, pixel- aspectJatio,frame Jate, bitJate, buffer-..size, quantizatiOllJnatrix, and so on. Optional sequence headers between Gaps can indicate parameter changes.

Group of Pictures (GOPs) layer. A GOP contains one or more pictures, one of which must be an I-picture. The GOP header contains information such as time_code to indicate hour-minute-second-frame from the start of the sequence.

Picture layer. The three common MPEG-l picture types are I-picture (intra-coding), P-picture (predictive coding), and B-pict[(]"e (Bidirectional predictive coding), as dis- cussed above. There is also an uncommon type, D-picture (DC coded), in which only DC coefficients are retained. MPEG-l does not allow mixing D-pictures with other types, which makes D- pictures impractical.

Cussed above. There is also an uncommon type, D-picture (DC coded), in which only DC coefficients are retained. MPEG- l does not allow mixing D-pictures with other types, which makes D-pictures impractical.
Slice layer. As mentioned earlier, MPEG-l introduced the slice notion for bit rate control and for recovery and synchronization after lost or corrupted bits. Slices may have variable numbers of macro blocks in a single picture. The length and position of each slice are specified in the header.

MPEG·2:

Development of the MPEG-2 standard started in 1990. Unlike MPEG-l, which is basically a standard for storing and playing video on the CD of a single computer at a low bitrate (l.5 Mbps), MPEG-2 []J is for big!J.g..=-quality vide~)<lta.bltrate ol)!l0re tha.!!...1 M~p~ It was Initially developed as a standard~ for digital broadcast TV.

In the late 1980s, Advanced TV (A1Y) was envisioned, to broadcast HDTV via terrestrial networks. During the development of lvIPEG-2, digital ATV finally took precedence over various early attempts at analog solutions to HDTV. MPEG-2 has managed to meet the compression and bit rate requirements of digital TVIHDTV and in fact supersedes a separate standard, MPEG-3, initially thought necessary for HDTV.

The MPEG-2 audio/video compression standard, also referred to as ISO/IEC 13818 [8], was approved by the ISO/IEC Moving Picture Experts Group in November 1994. Similar to MPEG-l, it has parts for Systems, Video, Audio, Confonnance, and Software, plus other aspects.:NIPEG-2 bas gained wide acceptance beyond broadcasting digital TV over terrestrial, satellite, or cable networks. Among various applications such as Interactive TV, it is also adopted for digital video discs or digital versatile discs (DVDs).

TABLE 11.6: Four levels in the main profile of MPEG-2.

Level	Maximum resolution	Maximum fps	Maximum pixels/sec	Maximum coded data rate (Mbps)	Application
High	1,920 × 1,152	60	62.7×10^6	80	Film production
High 1440	1,440 × 1,152	60	47.0×10^6	60	Consumer HDTV
Main	720 × 576	30	10.4×10^6	15	Studio TV
Low	352 × 288	30	3.0×10^6	4	Consumer tape equivalent

MPEG-2 defined seven profiles aimed at different applications (e.g., low-delay video conferellCing, scalable video, HDTV). The profiles are Simple, Main, SNR scalable, Spatially scalable, High, 4:2:2, and Multiview (where two views would refer to stereoscopic video). Within each profile, up to four levels are defined. As Table 11.5 shows, not all profiles have four levels. For example, the Simple profile has only the Main level; whereas the High profile does not have the Low level.

Table 11.6 lists the four levels in the Main profile, with the maximum amount of data and targeted applications. For example, the High level supports a high picture resolution of 1,920 x 1,152, a maximum frame rate of 60 fps, maximum pixel rate of 62.7 x 106 per second, and a maximum data rate after coding of 80 Mbps. The Low level is targeted at SIP video; hence, it provides backward compatibility with MPEG-l. The Main level is for CCIR601 video, whereas High 1440 and High levels are aimed at European HDTV and North American HDTV, respectively.

The DVD video specification allows only four display resolutions: 720 x 480, 704 x 480, 352 x 480, and 352 x 240. Hence, the DVD video standard uses only a restricted form of the MPEG-2 Main profile at the Main and Low levels.

Supporting Interlaced Video

MPEG-J supports only non interlaced (progressive) video. Since MP-EG-2jsadopted by digital broadcast TV, it must also support interlaced video, because this is one of the options for digital broadcast TV and HDTV.
As mentioned earlier, in-interlaced-video, each frame consists of two _fit;:ldl>, refeQed to as the top-field and the bottom- field. In ajrame-picture, all scan lines from both fields are interleaved to -forJ!la 'sI~le frame. This is then divided into 16 x 16 macro blocks and c-oded usiiig motion compensation. On the other hand, if each -field is treated as a separate picture, then ‘it is called field-picture. As Figure 11.6(a) shows, each frame-picture can be split into two field-pictures. The figure shows 16 scan lines from a frame-picture on the left, as opposed to 8 scan lines in each of the two field portions of a field- picture on the right.

We see that, i'n terms of display area on the monitor/TV, each 16-column x 16-row macro block in the field-picture corresponds to a 16 x 32 block area in the frame-picture, whereas each 16 x 16 macro block in the frame-picture corresponds to a 16 x 8 block area.

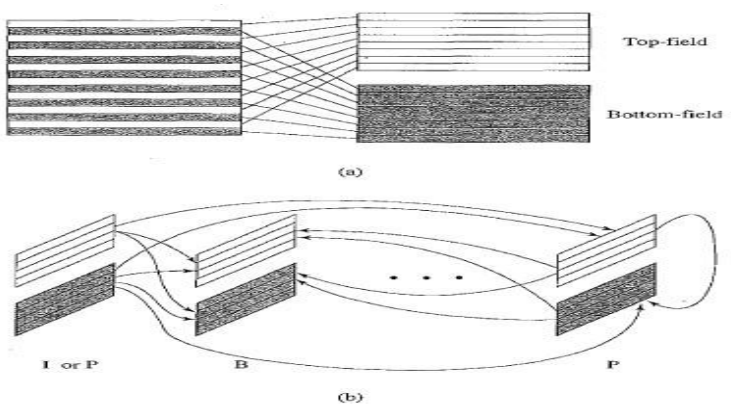


FIGURE 11.6: Field pictures and field-prediction for field-pictures in MPEG-2: (a) frame-picture versus field-pictures; (b) field prediction for field-pictures.

Five Modes of Predictions MPEG-2 define frame prediction and field prediction as well as five different prediction modes, suitable for a wide range of applications where the requirement for the accuracy and speed of motion compensation vary.

Frame predictipl1 for frame-pictures. This is identical to MPEG-l motion-compensation- based prediction methods in both P-frames and B-frarnes. Frame prediction works well for videos containing only slow and moderate object and camera motions.

Field prediction for field-pictures. This mode uses a macroblock size of 16 x 16 from field-pictures. For P-field-pictures (the rightmost ones shown in the figure), predictions are made from the two most recently encoded fields. Macroblocks in the top-field picture are forward-predicted from the top-field or bottom-field pictures of the preceding 1- or P-frame. Macroblocks in the bottom-field picture are predicted from the top-field picture of the same frame or the bottom-field picture of the preceding 1- or P- frame.For B-field-pictures, both forward and backward predictions are made from fieldpictures of preceding and succeeding 1- or P-frames. No regulation requires that field "parity" be maintained - that is, the top-field and bottom-field pictures can be predicted from either the top or bollom fields of the reference pictures.

Field prediction for frame-pictures. This mode treats the top-field and bottomfield of a frame-picture separately. Accordingly, each 16 x 16 macroblock from the target frame-picture is split into two 16 x 8 parts, each coming from one field. Field prediction is carried out for these 16 x 8 parts in a manner similar to that shown in Figure 11.6(b). Besides the smaller block size, the only difference is that the bottomfield will not be predicted from the top-field of the same frame, since we are dealing with frame-pictures now. For example, for P-frame-pictures, the bottom 16 x 8 part will instead be predicted from either field from the preceding 1- or P-frame. Two motion vectors are thus generated for each 16 x 16 macroblock in the P-frame-picture. Similarly, up to four motion vectors can be generated for each macroblock in the B- frame-picture.

Alternate Scan and Field-DCT:

Alternate Scan and field DCT are techniques aimed at improving the effectiveness of DCT on prediction errors. They are applicable only to frame-pictures in interlaced videos.
After frame prediction in frame-pictures, the prediction error is sent to DCT, where each block is of size 8 x 8. Due to the nature of interlaced video, the consecutive rows in these

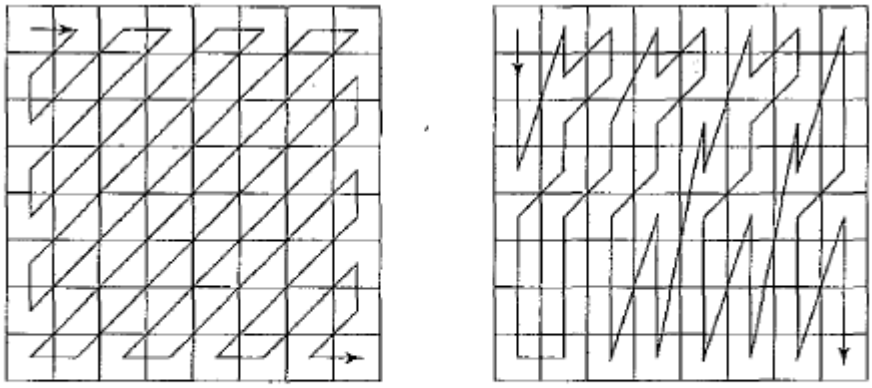


FIGURE 11.7: (a) Zigzag (progressive) and (b) alternate (interlaced) scans of DCT coefficients for videos in MPEG-2.

blocks are from different fields; hence, there is less correlation between them than between the alternate rows. This suggests that the DCT coefficients at low vertical spatial frequencies tend to have reduced magnitudes, compared to the ones in noninterlaced video. Based on the above analysis, an alternate scan is introduced. It may be applied on a picture-by- picture basis in MPEG-2 as an alternative to a zigzag scan. As Figure 11.7(a) indicates. zigzag scan assumes that in noninterlaced video, theDCT coefficients at the upper left corner of the block often have larger magnitudes. Alternate scan (Figure II.7(b» recognizes that in interlaced video, higher spatial frequency components may have Ihagnitudes and thus allows them to be sCllnned. earlier in these quence. Experiment shave shown [7] that alternate scan can impro~e the PSNR by up to 0.3 dB over zigzag scan and is most effective for videos with fast motion. . .... -- -,- -- - .In IvIPEG-2, Field..DCT can address the same issue. Before applying DCT, rows in the macro block of frame-pictures can be reordered, so that the first eight rows are from the top field and the last eight are from the bottom-field. This restores the higher spatial redundancy(and correlation) between consecutive rows. The reordering will be reversed after the mCT.Field..DCT is not applicable to chrominance images, where each macro block has only 8 x 8 pixels.

MPEG-2 Scalabilities:

- The MPEG-2 scalable coding: A base layer and one or more enhancement layers can be defined — also known as layered coding.
- The base layer can be independently encoded, transmitted and decoded to obtain basic video quality.
- The encoding and decoding of the enhancement layer is dependent on the base layer or the previous enhancement layer.
- Scalable coding is especially useful for MPEG-2 video transmitted over networks with following characteristics:
- Networks with very different bit-rates.
- Networks with variable bit rate (VBR) channels.
- Networks with noisy connections.

MPEG-2 supports the following scalabilities:

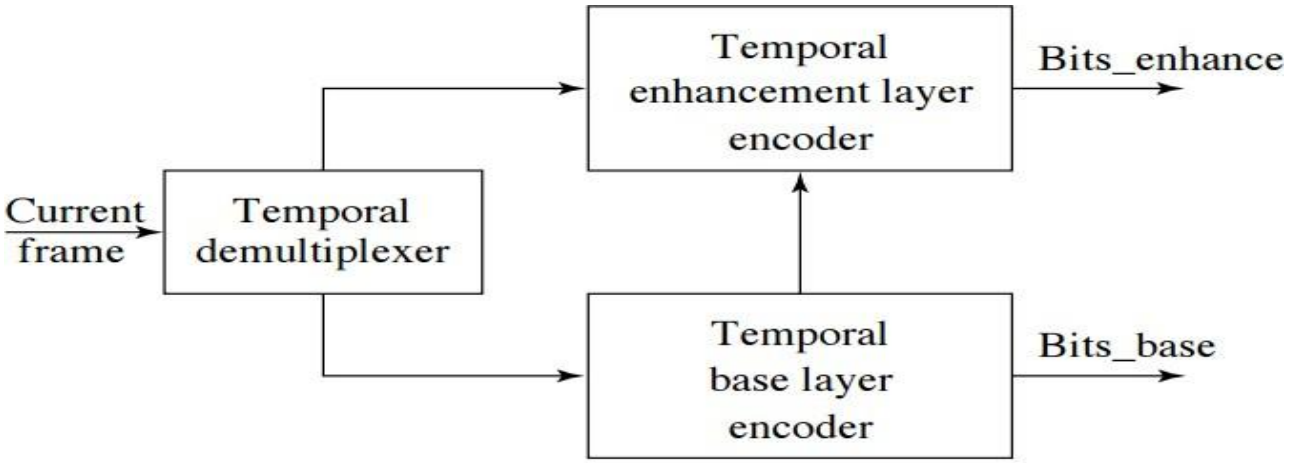
- SNR Scalability — enhancement layer provides higher SNR.
- Spatial Scalability — enhancement layer provides higher spatial resolution.
- Temporal Scalability — enhancement layer facilitates higher frame rate.
- Hybrid Scalability — combination of any two of the above three scalabilities.
- Data Partitioning — quantized DCT coefficients are split into partitions.

SNR Scalability:

Refers to the enhencement/refinement over the base layer to improve the Signal-Noise-Ratio (SNR).
The MPEG-2 SNR scalable encoder will generate output bitstreams Bits base and Bits enhance at two layers:
At the Base Layer, a coarse quantization of the DCT coefficients is employed which results in fewer bits and a relatively low quality video.
The coarsely quantized DCT coefficients are then inversely quantized (Q−1) and fed to the Enhancement Layer to be compared with the original DCT coefficient.
Their difference is finely quantized to generate a DCT coefficient re- finement, which, after VLC, becomes the bitstream called Bits enhance.

Temporal Scalability:

The input video is temporally demultiplexed into two pieces, each carrying half of the original frame rate.
Base Layer Encoder carries out the normal single-layer coding procedures for its own input video and yields the output bitstream Bits base.
The prediction of matching MBs at the Enhancement Layer can be obtained in two ways:
Interlayer MC (Motion-Compensated) Prediction
Combined MC Prediction and Interlayer MC Prediction



(a) Block Diagram

Hybrid Scalability:

- Any two of the above three scalabilities can be combined to form hybrid scalability:
- Spatial and Temporal Hybrid Scalability.
- SNR and Spatial Hybrid Scalability.
- SNR and Temporal Hybrid Scalability.
- Usually, a three-layer hybrid coder will be adopted which consists of Base Layer, Enhancement Layer 1, and Enhancement Layer 2.

Data Partitioning:

- Base partition contains lower-frequency DCT coefficients, enhancement partition contains high-frequency DCT coef- ficients.
- Strictly speaking, data partitioning is not layered coding, since a single stream of video data is simply divided up and there is no further dependence on the base partition in generating the enhancement partition.
- Useful for transmission over noisy channels and for progressive transmission.

Other Major Differences from MPEG-1:

- Better resilience to bit-errors: In addition to Program Stream, a Transport Stream is added to MPEG-2 bit streams.
- Support of 4:2:2 and 4:4:4 chroma subsampling.
- More restricted slice structure: MPEG-2 slices must start and end in the same macroblock row. In other words, the left edge of a picture always starts a new slice and the longest slice in MPEG-2 can have only one row of macroblocks.
- More flexible video formats: It supports various picture resolutions as defined by DVD, ATV and HDTV.

What Is Web 2.0?

In a sense, this entire chapter defines Web 2.0, but let's begin with a brief, one-section discussion. Web 1.0 was focused on a relatively small number of companies and advertisers producing content for users to access—some people called the web at the time the "brochure web." Web 2.0 involves the user—not only is the content often created by users, but users help organize it, share it, remix it, critique it, update it, etc. One way to look at Web 1.0 is as a lecture, a small number of professors informing a large audience of students. In comparison, Web 2.0 is a conversation, with everyone having the opportunity to speak and share views.

Web 2.0 embraces architecture of participation—a design that encourages user interaction and community contributions.6 You, the user, are the most important aspect of Web 2.0—so important, in fact, that in 2006, TIME Magazine's "Person of the Year" was "you."7 The article recognized the social phenomenon of Web 2.0—the shift away from a powerful few to an empowered many. "We can't be device centric...we must be user centric." —Bill Gates, MIX06 conference8.

Many Web 2.0 companies are built almost entirely on user-generated content and harnessing collective intelligence. The significance is not just in having user-generated content, but in how it is used. Google—the leading search engine and Internet advertising company—sends its users to user-generated websites by considering what users collectively have valued in the past. For websites like MySpace®, Flickr™, YouTube and Wikipedia®, users create the content, while the sites provide the platforms. These companies trust their users—without such trust, users cannot make significant contributions to the sites."A platform beats an application every time."

The architecture of participation is seen in software development as well. Open source software is available for anyone to use and modify with few or no restrictions—this has played a major role in Web 2.0 development. Harnessing collective intelligence,10 communities collaborate to develop software that many people believe is better than proprietary software. You, the user, are not only contributing content and developing open source software, but you are also directing how media is delivered, and deciding which news and information outlets you trust. Many popular blogs now compete with traditional media powerhouses. Social bookmarking sites such as del.icio.us and Ma.gnolia allow users to recommend their favorite sites to others. Social media sites such as Digg™ or Reddit enable the community to decide which news articles are the most significant. You are also changing the way we find the information on these sites by tagging (i.e., labeling) web content by subject or keyword in a way that helps anyone locate information more effectively. This is just one of the ways Web 2.0 helps users identify new meaning in already Existing content. RSS feeds (Chapter 12, XML and RSS) enable you to receive new information as it is updated—pushing the content right to your desktop. The rise of social networks has changed the way we interact and network. MySpace— the largest social network—has rapidly become the world's most popular website. Other popular social networking sites include Face book, Bebo, LinkedIn, and Second Life—a 3D virtual world where you interact with others via your online persona called an avatar.

Search:

In Web 2.0, the saying "content is king" remains a prevailing theme. With seemingly endless content available online, the findability of content becomes key. Search engines are the primary tools people use to find information on the web. Today, you perform searches with keywords, but the future of web search will use natural language (see, for example, Powerset.com). Currently, when you enter a keyword or phrase, the search engine finds matching web pages and show you a search engine results page (SERP) with recommended web pages listed and sorted by relevance. People-assisted search engines have also emerged, such as Mahalo, which pays people to develop search results.14 The popularity of vertical search engines—ones that focus on a specific topic or industry—is on the rise, though traffic to these search engines is still far behind the major (more generalized) search engines.

Traffic to the major search engines is growing rapidly—according to a recent com-Score (a web analytics company) report, Americans conducted 8 billion search queries in June 2007, up 26% from the previous year. In the same report, the comScore analysis of U.S. market share across the most popular search engines reported Google at the top with49.5% of the U.S. search market, followed by Yahoo! with 25.1%, Microsoft with 13.2%,Ask with 5.0% and Time Warner Network with 4.2% John Battelle's book The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture provides an extensive history of search engines and presents strong arguments for the importance of search in almost every aspect of our personal and business lives. John Battelle's Search blog discusses search and technology issues (http://battellemedia.com).

Attention Economy:

"Telecommunications bandwidth is not a problem, but human bandwidth is."
—Thomas Davenport and John Beck, The Attention Economy.
The abundant amounts of information being produced and people's limited free time has led to an attention **economy**. More content is available than users can sort through on their own, especially given the demands on their time, such as responsibilities to children, parents, friends, employers, etc. The Attention Economy, by Thomas Davenport and John Beck, begins with the familiar story of a man whose attention is constantly demanded by work and family. The authors explain that the constant flow of information in today's world causes attention to continually be diverted.
Though it used to be difficult to obtain diverse content, there are now seemingly endless options competing for an audience's attention. As a result, search engines have gained popularity by helping users quickly find and filter the information they want.

Google Search:

Google is the leading search and online advertising company, founded by Larry Page and Sergey Brin while they were Ph.D. students at Stanford University. Google is so popular that its name has been added to the Oxford English Dictionary—the verb "Google" means to find something on the Internet using the Google search engine. ("google" with a lowercase "g" is a cricket term, whereas "googol" or 10100 is the mathematical term Google was named after.)

Google's success in search is largely based on its Page Rank algorithm (patented by Stanford University and Larry Page) and its unique infrastructure of servers that uses linked PCs to achieve faster responses and increased scalability at lower costs.19 Estimates on the number of Google servers range from hundreds of thousands to over one million.20 The Page Rank algorithm considers the number of links into a web page and the quality of the linking sites (among other factors) to determine the importance of the page. Each inbound link is a vote saying that site is valuable to someone else; however, votes are given different weights depending on the "voter" site's own value. So, two pages could have the same Page Rank even if one has numerous links in from other pages and the other has fewer links in but from pages with higher Page Rank. Google search also considers all of the content on the page, its fonts, its headers and the content of neighboring pages.
Sites with the highest Page Rank will appear at the top of the search results.

In addition to its regular search engine, Google offers specialty search engines for images, news, videos, blogs and more. Using Google web services, you can build Google Maps and other Google services into your applications (see Section 1.13, Web Services, Mashups, Widgets and Gadgets).

AdWords, Google's pay-per-click (PPC) contextual advertising program (launched in 2000), is the company's main source of revenue. AdWords ads appear next to search results on the Google site (and are related to the search query). Advertisers write their own ads, which are unobtrusive and uniform in appearance—each ad consists of a headline, limited text and a URL. Advertisers bid on search keywords related to their ads and pay based on the number of users who click on the ads. AdSense is Google's advertising program for publishers (sites like http:// www.deitel.com that offer content), inspired by Susan Wojcicki, the vice president of product management. (In 1998, Wojcicki rented a spare room in her house to Larry Page and Sergey Brin where they founded Google.)22 Ad Sense is a fundamental and popular form of website monetization, particularly for Web 2.0 startup companies. Google text ads (as well as banner and rich-media ads) are placed on participating sites with related content. Click-through rates on contextual ads are often higher than on non-contextual ads because the ads reach people expressing interest in a related topic. As a result, contextual pay-per-click ads generally pay a higher eCPM (effective cost per thousand impressions).

Yahoo!

Yahoo! was started in 1994 by Jerry Yang and David Filo (also Stanford Ph.D. students) as a web directory rather than a search engine. The original site, "Jerry and David's Guide to the World Wide Web," consisted of their favorite websites manually added to a categorized directory. As the web grew, maintaining the directory structure became increasingly difficult, and a search capability was created for better access to the data. Focusing more on search, Yahoo! also expanded into other areas, becoming a popular provider of e-mail, user groups and more. In 2003, Yahoo! acquired Overture (now Yahoo! Search Marketing), which was the first search engine to offer sponsored search results successfully.24

MSN

MSN search was created in 1998, a year after Google was launched.25 Over the past few years, Microsoft has made search engine technology development a top priority.26 Microsoft search query volume and its search market share grew rapidly in June 2007; analysis companies comScore and Compete attribute this boost largely to MSN's Live Search club, a program introduced in May 2007 to reward users of Live Search.27, 28 MSN's Live Search includes a new search engine, index and crawler.29 It allows you to search the web, performing specialized searches (news, images, or local listings) or MSN content searches. 30 Another approach that Microsoft is taking to increase its search market share is buying vertical search sites such as MedStory, a health search engine. 31 Microsoft is also looking to gain market share in the contextual advertising market through Microsoft ad-Center (similar to Google AdWords and Yahoo! Search Marketing).

Ask

Ask (formally known as AskJeeves.com) is owned by InterActiveCorp (IAC), which also owns Ticketmaster®, Match.com®, LendingTree.com®, RealEstate.com® and many other Internet properties. In June 2007, Ask launched a new search site, which includes a new design with a simple homepage default, customizable backgrounds, new video search(powered by Blinkx) and the ability to view video previews and listen to music clips. The search results are based on the searcher's location—Ask will report relevant local businesses and events. Searching for movies, for example, will show local show times.

Vertical Search

Vertical search engines are specialists (focusing on specific topics) in comparison to generalists (e.g., Google and Yahoo!).32 Vertical search engines enable you to search for resources in a specific area, with the goal of providing you with a smaller number of more relevant results. Popular vertical search engines include travel sites (such as Kayak or Expedia), real-estate sites (such as Zillow or Trulia), job search sites (such as Indeed or Monster) and shopping search engines (such as Shopzilla and MySimon).

Location-Based Search

Location-based search (offered by most major search engines as well as some smaller specialized ones) uses geographic information about the searcher to provide more relevant search results. For example, search engines can ask the user for a ZIP code or estimate the user's general location based on IP address. The engine can then use this information to give higher priority to search results physically located near the user. This is particularly useful when searching for businesses such as restaurants or car services. (See Section 1.14 for more information on location-based services.)

Creating Customized Search Engines

Rollyo—a build-your-own customized search engine website—allows you to explore, create and personalize search engines ("searchrolls") created by others. This helps you narrow your search to sites you already trust.33 Other custom search sites include Giga blast and Google Custom Search Engine.

Search Engine Optimization (SEO)

Search Engine Optimization (SEO) is the process of designing and tuning your website to maximize your findability and improve your rankings in organic (non-paid) search engine results. To maximize traffic, you need to take into consideration how search engines work when you design your website. There are two ways of employing SEO. The first, white hat SEO, refers to methods that are approved by search engines, do not attempt to deceive the search engines, and produce quality, long-term results. Top white hat techniques for SEO include: offering quality content, using proper metadata and effective keywords, and having inbound links from relevant high-quality pages.34 Black hat methods are used to deceive search engines. Although they may result in temporary improvement in search engine results, these tactics could get your site banned by the search engines. A"Google bomb" (or link bomb) is an example of a black hat method—it attempts to trick the Google algorithm into promoting a certain page (generally for humorous reasons).

Link Building

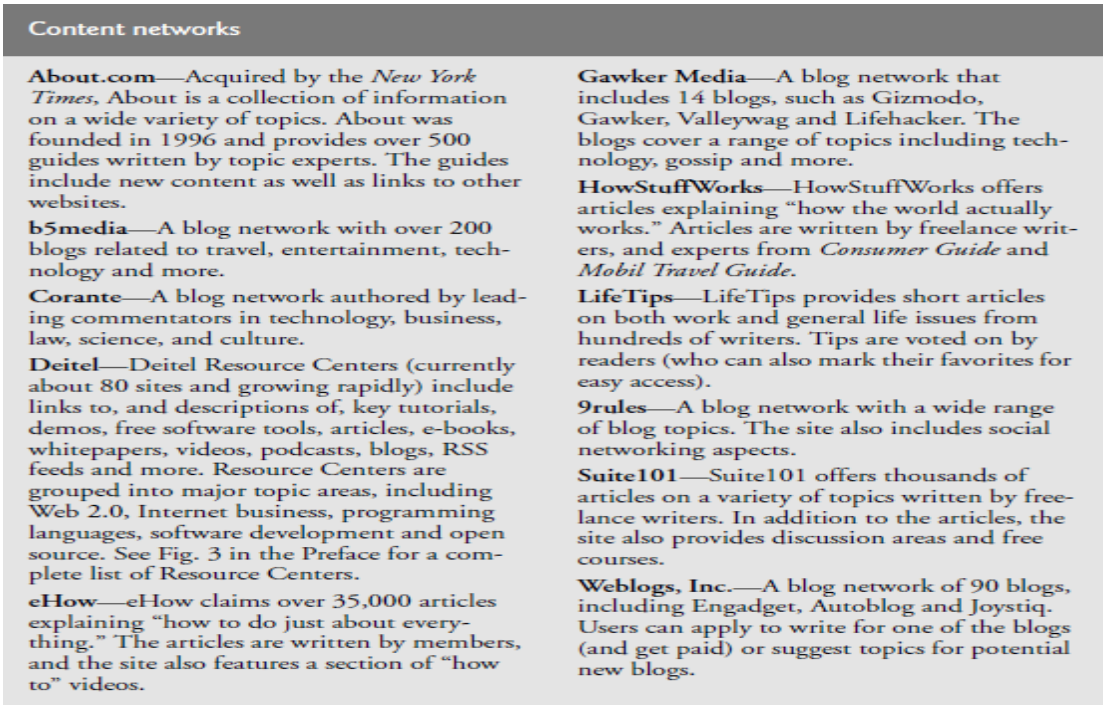
Link building is the process of increasing search engine rankings and traffic by generating inbound links to a particular website. Search engine algorithms regard each link as a vote for the destination website's content, so sites with the greatest link popularity (or number of high-quality inbound links) appear highest on search engine result pages (SERPs). The three most practiced methods of building links include reciprocal linking, link baiting and natural linking. Reciprocal linking is an exchange in which two related websites link toeach other, increasing the link popularity of both sites and adding value for site users. Link baiting involves creating attention-grabbing web content specifically for viral (exponentially increasing) exposure through social media and social bookmarking websites. Natural linking is the process of building one-way inbound links by optimizing website content and user experience without the explicit solicitation of a back link. Search algorithms are continuously updated to prevent black hat SEOs from deceiving search engines with automated linking software and links from directories or other low-quality websites. One way links from websites with strong, related pages are given greater weight than reciprocal links, links from sites with unrelated content or links from sites with low Page Rank.

Search Engine Marketing (SEM)

Search Engine Marketing (SEM) is the method of promoting your website to increase traffic and search results by raising the site's visibility on search engine results pages.Danny Sullivan (founder of Search Engine Watch and, more recently, Search Engine Land) introduced the term "Search Engine Marketing" in 2001 to include SEO, managing paid listings, developing online marketing strategies and submitting sites to directories. SEO is the most popular form of search engine marketing, which continues to takeaway business from other marketing channels (especially offline sources).

Content Networks:

Content networks are websites or collections of websites that provide information in various forms (such as articles, wikis, blogs, etc.). These provide another way of filtering the vast amounts of information on the Internet, by allowing users to go to a trusted site that has already sorted through many sources to find the best content or has provided its own content. Figure 1.2 shows some examples of content networks.



Content networks

About.com—Acquired by the *New York Times*, About is a collection of information on a wide variety of topics. About was founded in 1996 and provides over 500 guides written by topic experts. The guides include new content as well as links to other websites.

b5media—A blog network with over 200 blogs related to travel, entertainment, technology and more.

Corante—A blog network authored by leading commentators in technology, business, law, science, and culture.

Deitel—Deitel Resource Centers (currently about 80 sites and growing rapidly) include links to, and descriptions of, key tutorials, demos, free software tools, articles, e-books, whitepapers, videos, podcasts, blogs, RSS feeds and more. Resource Centers are grouped into major topic areas, including Web 2.0, Internet business, programming languages, software development and open source. See Fig. 3 in the Preface for a complete list of Resource Centers.

eHow—eHow claims over 35,000 articles explaining "how to do just about everything." The articles are written by members, and the site also features a section of "how to" videos.

Gawker Media—A blog network that includes 14 blogs, such as Gizmodo, Gawker, Valleywag and Lifehacker. The blogs cover a range of topics including technology, gossip and more.

HowStuffWorks—HowStuffWorks offers articles explaining "how the world actually works." Articles are written by freelance writers, and experts from *Consumer Guide* and *Mobil Travel Guide*.

LifeTips—LifeTips provides short articles on both work and general life issues from hundreds of writers. Tips are voted on by readers (who can also mark their favorites for easy access).

9rules—A blog network with a wide range of blog topics. The site also includes social networking aspects.

Suite101—Suite101 offers thousands of articles on a variety of topics written by freelance writers. In addition to the articles, the site also provides discussion areas and free courses.

Weblogs, Inc.—A blog network of 90 blogs, including Engadget, Autoblog and Joystiq. Users can apply to write for one of the blogs (and get paid) or suggest topics for potential new blogs.

User-Generated Content:

User-generated content has been the key to success for many of today's leading Web 2.0 companies, such as Amazon, eBay and Monster. The community adds value to these sites, which, in many cases, are almost entirely built on user- generated content. For example, eBay (an online auction site) relies on the community to buy and sell auction items, and Monster (a job search engine) connects job seekers with employers and recruiters. User-generated content includes explicitly generated content such as articles, home videos and photos. It can also include implicitly generated content— information that is gathered from the users' actions online. For example, every product you buy from Amazon and every video you watch on YouTube provides these sites with valuable information about your interests. Companies like Amazon have developed massive databases of anonymous user data to understand how users interact with their site. For example, Amazon uses your purchase history and compares it to purchases made by other users with similar interests to make personalized recommendations (e.g., "customers who bought this item also bought..."). Implicitly generated content is often considered hidden content. For example, web links and tags are hidden content; every site you link to from your own site or bookmark on a social bookmarking site could be considered a vote for that site's importance. Search engines such as Google (which uses the Page Rank algorithm) use the number and quality of these links to a site to determine the importance of a site in search results.

Collective Intelligence

Collective intelligence is the concept that collaboration can result in smart ideas. Working together, users combine their knowledge for everyone's benefit.The first chapter of Wikinomics, by Don Tapscott and Anthony D. Williams, tells the Goldcorp story. Inspired by the community efforts in Linux, the CEO of Goldcorpreleased to the public proprietary geological information about the company's land. Gold corpoffered cash rewards to people who could use this information to help the companylocate gold on the land. The community helped his company find 8million ounces of gold,catapulting Goldcorp from $100 million in stock equity to $9 billion.39 Goldcorp reapedamazing benefits by sharing information and encouraging community participation.User-generated content is significant to Web 2.0 companies because of the innovativeways companies are harnessing collective intelligence. We've already discussed Google'sPageRank (Section 1.3), which is a product of collective intelligence. Amazon's and Last.fm's personalized recommendations also result from collective intelligence, as algorithms evaluate user preferences to provide you with a better experience by helping youdiscover new products or music preferred by other people with similar interests. We sabe is a web community where members share their decisions about money and savings—the site uses the collective financial experiences of the community to create recommendations. Reputation systems (used by companies like eBay) also use collective intelligence to build trust between buyers and sellers by sharing user feedback with the community. Social bookmarking sites (Section 1.10), and social media sites (like Digg and Flickr) use collective intelligence to promote popular material, making it easier for others to find.

Collaborative Filtering

Though collaboration can result in a wealth of knowledge, some users might submit false or faulty information. For example, Wikipedia has experienced instances of people deliberately adding false information to entries. While moderation (monitoring of content by staff) is sometimes necessary, it is time consuming and costly. Many Web 2.0 companies rely on the community to help police their sites. This collaborative filtering lets users promote valuable material and flag offensive or inappropriate material. Users have the power to choose for themselves what is important. Examples of sites using collaborative filtering include Digg, a news site where users rate the stories (see Section 1.8), and social bookmarking sites such as del.icio.us, where users can easily find popular sites (see Section 1.10).Customer reviews on Amazon products also employ collaborative filtering—readers vote on the usefulness of each review (helping other readers to find the best reviews).

History of Blogging

Blogs are websites consisting of entries listed in reverse chronological order. They have existed since the mid-1990s; however, interest in blogging has grown exponentially in recent years because of easy-to-use blogging software and increasingly economical Internet access. The term "blog" evolved from weblog, a regularly updated list of interesting websites. These blogs consisted of short postings, in reverse chronological order, that contained

links to other web pages and short commentaries or reactions. Blogging has since taken on a looser structure—some blogs still follow the traditional format of links and small amounts of text, while others consist of essays, sometimes not containing any links. Blogs can also now incorporate media, such as music or videos. Many people are familiar with personal journal blogs, like those on Xanga or LiveJournal. These sites include social networking features and are particularly popular with teenage bloggers, who often write about their day-to-day lives for friends. Blogging has become a major social phenomenon, empowering users to participate in, rather than just view, the web. In July 2006 most **bloggers**, or blog authors, had not had a personal website before starting their blog.49 The increased availability of user-friendly blogging software has allowed blogging to become accessible to more mainstream Internet users.

Blog Components

Reader comments create an interactive experience, allowing readers to react to blog entries. According to a Pew Internet study, 87% of blogs allow reader comments.50 Success- ful bloggers pay attention to their readers and respond, often sparking interesting discussions and debates. However, allowing comments increases the possibility of spam (including irrelevant comments, inappropriate language and link spam—where a user tries to increase an irrelevant site's number of inbound links). By some estimates, over 90% of blog comments are spam. Permalinks provide blog readers with a way of linking to specific blog entries. Each blog post has a unique URL referring to that single post. Links stay relevant even after the blog entry moves off the homepage and into the archive.

Trackbacks tell bloggers who is linking to their posts. This enhances Internet content by making linking two-way. The blogger provides a trackback link, and sites that use the link are added to a list on the blog entry. For an example of a trackbacks section, visit http://www.techcrunch.com/2006/08/08/web-20-the-24-minute-documentary/. This is a permalink to a post on TechCrunch, a popular Internet technology blog, which features a Web 2.0 video from 2006. A blogroll is a list of the blogger's favorite blogs. Though not all blogs feature a blogroll, it is common for the main page of a blog to contain links to several other blogs. For example, Live Journal automatically incorporates a blog roll (consisting of users the blogger has marked as friends) into a user's profile page.

Blogging and Journalism:

Blogging has encouraged citizen journalism, allowing anyone to be a journalist. Blogs have become a significant news resource, drawing traffic away from the mainstream media. Some argue that this form of "participatory journalism" holds less biases than mainstream media, or at least makes these biases clear and provides many different views. This democratization of media allows a larger group to take part in journalism. Traditional journalists had previously been able to create a representative democracy (much like the political system of the United States) by speaking for the masses. However, blogging gives a voice to everyone with a computer and Internet access, creating a more direct democracy.

Many bloggers are recognized as members of the media. Just as television and radio increased the speed of news delivery over that of newspapers, blogs have become a fast and in-depth (and often "unwashed") news medium. The mass media is embracing blogging; many TV news anchors suggest that viewers read their blogs after the show, and many newspaper websites feature blogs by reporters.

Though journalism is a large part of the blogging phenomenon, according to a Pew Internet study only one-third of bloggers consider their blogs a form of journalism. Eighty-four percent of bloggers consider it a hobby, and only 10% spend more than ten hours a week blogging. Posting new content and responding to reader comments requires a substantial time commitment.

Gaia Online

Gaia Online is a popular teen virtual world. This online community allows teens to play games, make friends and express their creativity. Similar to Second Life, Gaia has its own marketplace where members can earn Gaia Gold for various actions they perform on the site (e.g., playing games or posting), and use their earnings at the virtual stores or for creating their own shops. Nearly 300,000 members login daily and about 2 million unique visitors login to Gaia every month.

Mobile Social Networking

Many social networking sites have found innovative ways of connecting people through the Internet and their mobile devices (such as cell phones and PDAs). Mobile users can send instant messages, check e-mail, and post content to the web from Internet-enabled mobile devices. The new Apple iPhone further realizes the dream of having the Internet in your pocket by allowing the full Internet (not a simplified mobile one) to be accessed wherever wireless Internet access is available. Google's Dodgeball.com provides users with mobile access to a network of friends in many cities. GPS chips in mobile devices allow Dodgeball users to update their location and be notified of nearby friends or "crushes." Dodge ball also provides an easy way of sending messages to groups of friends to plan get-togethers. (See Section 1.14, Location- Based Services.) Other sites such as Twitter provide similar services, accessible by text message, IM or a web client. Twitter users can message groups of friends at once and automatically receive their friends' updates on a cell phone or through a chat window. The site is considered to be a micro blogging service (since users are limited to a small number of characters for each update). Twitter offers a web services API, which allows developers to integrate Twitter into other applications. (See Section 1.13,Web Services, Mashups, Widgets and Gadgets, for more information on web services APIs.)

Social Media:

Social media refers to any media shared online (e.g., videos, music, photos, news, etc.). Hit wise reported that "increased broadband penetration, combined with the rise of consumer generated content and the proliferation of webcams and cell phone and home video cameras have firmly entrenched online video viewing into the habits of entertainment seekers in the United States.

YouTube

YouTube, launched in late 2005, is the leading Internet video site. In true Web 2.0 fashion, the entire site is based on user- generated content. Users upload videos, and rate and comment on videos posted by other users. YouTube's Quick Capture Flash software makes it easy to upload content directly from a webcam. Users can browse videos by category, tag, or by following "related video" links. Highly rated videos are featured on You-Tube's homepage. While many professionals and film students post content on the site, the most popular submissions are often simple spoofs or home videos. Because of the viral network effects of YouTube, these amateur videos can quickly gain worldwide attention.

Users can subscribe to other users' content, share videos with friends by e-mail, or embed videos directly into their blogs or other websites. YouTube addresses privacy and spam concerns by allowing users to set videos as "public" or "private" and flag inappropriate material for review by YouTube's staff.

Less than a year after its official launch, YouTube was acquired by Google (which had its own less popular Google Video site) for $1.65 billion. Less than six months after the acquisition, Viacom sued YouTube for $1 billion for copyright infringement. The Digital Millennium Copyright Act of 1998 protects companies from prosecution due to user actions if they work in "good faith" to remove offending content.85 However, interpretations of this act vary, and it has become a point of contention for many companies. You- Tube is developing a mechanism that automatically detects copyrighted material. Currently, illegal content is removed from the site manually.

Internet TV

Many mass-media companies now offer full-length episodes of popular television shows on their websites to tap into the increasingly popular Internet television market. The average American watches 4.5 hours of television a day, not including Internet television.86 Sites, such as **Joost**, Veoh and MobiTV, have emerged as a new way of watching television. Joost, for example, uses semantic technologies to help users find programs that interest them. (See Section 1.18, Future of the Web.) Limited by copyright issues, Internet TV sites must make deals with mainstream networks to offer their content online. Viacom made a deal with Joost, allowing the site to include some shows from networks such as MTV, VH1 and Comedy Central.87 As users take back the power to choose what they watch and when, networks may find themselves making more deals with Internet TV companies. As technologies continue to improve, Internet TV has the potential to radically change the television industry.

Already, smaller content creators are able to gain access to worldwide audiences. In late June 2007, MySpace joined the market with its MySpaceTV. With MySpace's enormous membership, it could rapidly become a direct competitor to YouTube and Internet TV websites. Internet TV allows advertisers to target their markets more precisely than with broadcast television. Advertisers can use demographic information, such as location, gender and age, to serve appropriate ads.

Digg

Digg features news, videos and podcasts, all posted and rated by users. It has gained popularity by allowing users to "digg" or "bury" posts and user comments. Valuable sites, marked by large numbers of diggs, are moved to the Digg front page where other users can easily find them. Formulas were adjusted to make sure the "wisdom of crowds" was not being hijacked by users trying to promote their own posts.88 Sites that are "dugg" and featured on the homepage typically experience a traffic surge. Bloggers can add Digg buttons to their sites, making it easy for readers to "digg" their posts. Digg uses collaborative filtering to help reduce spam by "burying" it (users can vote against posts they don't like). Users can also set the threshold of diggs to automatically filter out content with low ratings. The site was criticized for removing popular posts of HD DVD security cracks (on the advice of lawyers); however, Kevin Rose (Digg's founder) decided to support the crowds and "deal with whatever the consequences might be." Digg has additional social networking capabilities; users can view their friends' Digg activities and the Diggs of other users with similar interests. Some Digg-like sites include Netscape, Reddit and Newsvine.

Last.fm

Last.fm is an Internet radio website that usesWeb 2.0 concepts to make music recommendations and build communities. The site provides open source desktop software that can be integrated into most popular music players. Its scrobbling feature tracks the music users listen to so that Last.fm can provide users with personalized recommendations. A streamable radio with "discovery mode" and a network of like-minded listeners help users find new music. Groups and an events section add social value. The site also offers tagging and wiki pages for artists and record labels.

Digital Rights Management (DRM)

Digital Rights Management (DRM) systems add software to media files to prevent them from being misused, but these systems restrict compatibility with many media players. Companies want to protect their digital products from illegal distribution; however, users want unrestricted access to media they've purchased. **iTunes**, Apple's music store, has been criticized for restricting users' access to their own music by allowing only up to five computers to be authorized to play any given file. However, Apple's Steve Jobs advocated a DRM-free music world in February 2007, arguing the greater risk for piracy is in DRM-free CDs, which make up the majority of music sales.90 CDs remain DRM-free because many CD players are not compatible with DRM systems. In June 2007, Amazon offered DRM-free downloads from more than 12,000 record labels, and both iTunes and Amazon sell DRM-free music from EMI (one of the four major record companies).

Social Bookmarking:

Social bookmarking sites let you share your Internet bookmarks (e.g., your favorite websites, blogs, and articles) through a website. Users can access these bookmarks from any computer and discover new sites by searching popular bookmarks and tags. Some of the most popular social bookmarking sites are del.icio.us, Ma.gnolia, Blue Dot, Stumble- Upon, Simpy and Furl.

Services, Mashups, Widgets and Gadgets). For example, Adobe Illustrator uses the del.icio.us technology to organize bookmarks in the program's documentation.

Rich Internet Applications (RIAs):

"rich" features and functionality approaching that of desktop applications. Early Internet applications supported only a basic HTML graphical user interface (GUI). Though they could serve simple functions, these applications did not have the look or feel of a desktop application. The relatively slow Internet connections these applications relied on led to the term "World Wide Wait." RIAs are a result of today's more advanced technologies that allow greater responsiveness and advanced GUIs.

Ajax

The term Ajax (Asynchronous JavaScript and XML) was coined by Adaptive Path's Jesse James Garrett in February 2005. Ajax (see Chapter 13, Ajax-Enabled Rich Internet Applications)n allows partial page updates—meaning updates of individual pieces of a web page without having to reload the entire page. This creates a more responsive GUI, allowing busers to continue interacting with the page as the server processes requests. The technologies that make up Ajax— XHTML, CSS, JavaScript, the DOM, XML, and the XMLHttpRequest object—are not new. In fact, in the 1990s, Netscape used asynchronous page updates in LiveScript, which evolved into JavaScript. However, the popularity of Ajax has dramatically increased since its naming. Ajax performs a vital role inWeb 2.0, particularly in building webtop applications and enhancing the user's overall experience. The following toolkits and frameworks (environments with standard components that make development faster and easier) provide libraries and tools for convenient Ajax enabled application development.

Dojo

Dojo is an open source JavaScript toolkit—it is a library, not a framework. Dojo development began in late 2004. Dojo helps standardize JavaScript by providing a variety of packages for cross-browser compatibility, rich GUI controls, event handling and more.

JavaServer Faces (JSF) is a Java-based web application framework. JSF separates design elements from business logic and provides a set of user-interface components (JSF components) that make developing RIAs simple. One of the **Java BluePrints** projects provides additional resources and libraries for building Ajax-enabled applications.

ASP.NET Ajax

ASP.NET Ajax (Chapter 21) is an extension of the .NET framework for creating Ajaxenabled applications. It includes an open source Ajax Control Toolkit for implementing asynchronous functionality. ASP.NET Ajax is easily used in Microsoft Visual Web Developer or Microsoft Visual Studio to quickly create Rich Internet Applications.

Adobe Integrated Runtime and Google Gears

Though web application use has been increasing, many feel these programs cannot truly compete with desktop applications until the "Offline Problem" (not being able to access web applications and data when not connected to the Internet) has been solved. Busi- nesses can lose valuable time and money when Internet issues occur such as a slow or broken. Internet connection. Adobe released its Adobe Integrated Runtime (AIR; previously called Apollo) in beta form in June 2007. AIR allows users to run Flex web applications on their desktops even when they are not connected to the Internet, thus allowing users to remain efficient when they are unable to access the Internet or when an SaaS application server goes down. Users can continue their work and synchronize it with the servers again later. **Google Gears**, also in beta, is a similar product, allowing use of web applications while offline. Google Gears was created out of a Google engineer's 20% project, inspired by wanting to use Google Reader on a bus with "flaky" Internet access. (Google engineers devote 20% of their time to projects other than their usual work and 10% of their time to projects that are "truly new.")122 Dojo Offline (using the Dojo library) is built on top of Google Gears, creating an easy-to-use interface for using web applications offline.

Web Services, Mashups, Widgets and Gadgets:

Instead of reinventing the wheel with every new project, developers can use existing companies' web services to create feature-rich applications. Incorporating web services into new programs allows people to develop new applications quickly.

Widgets and Gadgets

Widgets, also referred to as gadgets, are mini applications designed to run either as standalone applications or as add-on features in web pages. Newsweek called 2007 the "Year of the Widget" because of the huge increase in popularity of these applications.127 Widgets can be used to personalize your Internet experience by displaying real-time weather conditions, aggregating RSS feeds, viewing maps, receiving event reminders, providing easy access to search engines and more. The availability of web services, APIs and various tools makes it easy even for beginner programmers to develop widgets. There are many catalogs of widgets online—one of the most all-inclusive is Widgipedia, which provides an extensive catalog of widgets and gadgets for a variety of platforms.

Amazon Web Services

Amazon is a leading provider of web services. The site provides historical pricing data and E-Commerce Services (ECS), which enable companies to use Amazon's systems to sell their own products. Amazon also offers hardware and communications infrastructure web services that are particularly popular with companies, providing economical web-scale computing. Amazon's Elastic Compute Cloud (EC2), Simple Storage Service (S3) and Simple Queue Service (SQS) enable businesses to pay for only the processing or storage space needed during any given period. This makes it possible for companies to save money (by not having to buy and maintain new hardware, software and communications equipment) while still being able to scale their storage and computing power to handle traffic surges (or reduce loss when the site's popularity declines). This is extremely significant in the Internet world, where a site's traffic can explode or crash overnight. Amazon also provides "artificial artificial intelligence" with its unique Mechanical Turk. This web service allows applications to call on people to perform tasks (such as identifying pictures) that are easier for humans to do than computers. People can sign up to become part of Mechanical Turk and bid on jobs (called Human Intelligence Tasks or HITs). This creates a competitive market, driving down developer costs, creating opportunities for people worldwide and allowing more applications to become feasible.

REST (Representational State Transfer)-Based Web Services

Representational State Transfer (REST) (originally proposed in Roy Thomas Fielding's doctoral dissertation128) refers to an architectural style for implementing web services. Though REST is not a standard, RESTful web services are implemented using web standards. Each operation in a RESTful web service is easily identified by a unique URL. So, when the server receives a request, it immediately knows what operation to perform. Such web services can be used in a program or directly from a web browser. In some cases, the results of a particular operation may be cached locally by the browser. This can make subsequent requests for the same operation faster by loading the result directly from the browser's cache.129 Amazon's S3 is RESTful, and many other Web 2.0 web services provide RESTful interfaces.
RESTful web services are alternatives to those implemented with SOAP (Simple Object Access Protocol). (We discuss both REST-based and SOAP-based web services in Chapter 24, Web Services.) With SOAP-based web services, the request and response are hidden (in entities known as a SOAP "envelopes"). SOAP requests must be deciphered as they are received at the server to determine the operation to perform and the arguments required to perform that operation. Similarly, the responses are encoded and deciphered on the client to obtain the result of the operation. SOAP does not currently provide a mechanism for caching results.

Location-Based Services:

Location-Based Services (LBS) are applications that take your geographic location (city, state, location of your mobile device, etc.) into consideration. While the term generally refers to services accessed on mobile devices using the Global Positioning System (GPS), it can also be used to describe web applications that take your location into account. Search engines including Yahoo! Local and Google Maps use **localization** to provide you with geographically relevant content. Local search is particularly useful when you want to find a nearby business (e.g., plumbers, taxis, etc.). Location-based services are becoming increasingly popular inWeb 2.0 applications.

GeoRSS and Geotagging

GeoRSS, based on the RSS standards, is a set of standards for representing geographical information in a feed. Location and geographical information in a GeoRSS feed can be used in GPS devices, mapping applications and other location- based services. For example, a blog post about a vacation could map the locations mentioned.134 Geotagging can be used to add location information (longitude, latitude, etc.) to websites, images, RSS feeds, videos and more. Websites can often determine a user's location by their IP address. Geotagging a website provides the user with location information about the site.135 Geographic information can be used to add value to search results. Geotagging could also be mashed up with existing visualization systems, such as Google Earth orMSN Virtual Earth, which provide advanced satellite images for anywhere on the planet. children (as long as the child is carrying the special cell phone).

Mapping Services

Google Maps is one of the most popular mapping applications available online. You can use Google Maps to locate businesses in your area, get driving directions and live traffic information, create custom maps with images and more. You can even get the information by using your mobile device. Google's local search allows you to locate a business in a geographic area and get its address, phone number, driving directions and even user reviews. Google Earth provides satellite images of virtually any location on the planet. In some areas, you can even get a panoramic view of a neighborhood at street level. You can use the Google Maps API to add mapping capabilities to your websites and web applications.

GeoRSS and Geotagging

GeoRSS, based on the RSS standards, is a set of standards for representing geographical information in a feed. Location and geographical information in a GeoRSS feed can be used in GPS devices, mapping applications and other location- based services. For example, a blog post about a vacation could map the locations mentioned.134 Geotagging can be used to add location information (longitude, latitude, etc.) to websites, images, RSS feeds, videos and more. Websites can often determine a user's location by their IP address. Geotagging a website provides the user with location information about the site.135 Geographic information can be used to add value to search results. Geotagging could also be mashed up with existing visualization systems, such as Google Earth orMSN Virtual Earth, which provide advanced satellite images for anywhere on the planet.

XML, RSS, Atom, JSON and VoIP:

For more information on any of the following technologies, visit the corresponding Resource Centers at http://www.deitel.com/resourcecenters.html (see Fig. 3 in the Preface for a complete list of Deitel Resource Centers). XML

XML (Extensible Markup Language, Chapter 12), developed in 1996 by the World Wide Web Consortium (W3C), is a markup language that allows you to label data based on its meaning. XML describes data in a way that is meaningful to both humans and computers.

XML documents are text files with a .xml extension; they can be created in text editors. These documents can reference a Document Type Definition (DTD) or a schema, which defines the structure for the document. This allows the information in the document to be verified by validating parsers, meaning they will check to make sure that no elements are missing (e.g., a last-name element in a document listing full names) and that the elements occur in the proper order. This makes XML data more reliable than data prepared with some other data-describing options. XML also can be used to create customized markup languages (e.g., XHTML for web content, CML for chemistry,MathML for mathematical content and formulas, and XBRL for financial data)—these are referred to as XML vocab- ularies. XHTML is described in Chapter 2, Introduction to XHTML. Chapter 12, XML and RSS, presents several examples that use MathML to render mathematical expressions.

RSS and Atom

Sites that offer RSS (Chapter 12) and Atom feeds can maintain an "open connection" with their readers. Users no longer have to regularly visit sites for updates—by subscribing to a site's feed, users receive updates as new information is posted to the site. The difference between RSS and Atom is subtle and unimportant to most users—many tools support both formats. Versions of RSS (an XML-based web content syndication format) have existed since the late 1990s; Atom dates to 2003. Most major web browsers support RSS and Atom feeds, and many aggregators (or feed readers) are available to help users organize their subscriptions. Feedburner (acquired by Google) is used by many blogs to provide their readers with new posts by e-mail. This service also helps bloggers get a better idea of the size of their audience (by allowing them to see the number of subscribers).

JSON

JavaScript Object Notation (JSON) was developed in 1999 as an alternative to XML. JSON (discussed in Chapter 13, Ajax-Enabled Rich Internet Applications) is a text-based data interchange format used to represent JavaScript objects as strings and transmit them over a network. It is commonly used in Ajax applications. JSON text is easy to produce and read—it is also faster to parse (or extract) than XML.

VoIP

Voice over Internet Protocol (VoIP) is the technology used to make free or inexpensive phone calls over the Internet. Some businesses and individuals have switched completely to VoIP and eliminated traditional phone lines to cut costs. There are many VoIP services, such as Vonage, Packet8 or Lingo; Skype is the most popular. Acquired by eBay to integrate buyer and seller voice communication into auctions, Skype offers free and feebased services (such as calling non- Skype phones). VoIP is an enabling technology that can be layered into Web 2.0 companies and websites.

Web 2.0 Monetization Models:

Many Web 1.0 businesses discovered that popularity ("eyeballs") was not the same as financial success. Web 2.0 companies are paying more attention to monetizing their traffic. Starting an Internet business is cheaper than ever, and the cost of failure is lower. Anyone can start earning modest amounts of money almost immediately, using the monetization models described in Fig. 1.5. Web 2.0 monetization is heavily reliant on advertising. Using Google's AdSense contextual advertising program is one of the fastest and most popular ways of monetizing a new Internet business.

Web 2.0 Business Models:

The technologies and collaborative nature of Web 2.0 have opened up new business models. Some of these would not have been feasible even ten years ago, but because of Moore's Law they are not only possible but thriving. At the moment, there is no foreseeable end to the advancements attributed to Moore's Law, so fantastic ideas that are impossible today may become possible within just a few years. Figure 1.6 outlines many popular Internet business models and lists some companies that use each one. In just about every case, there are many more companies using that business model.

Future of the Web:

The XHTML coding on websites defines their structure and layout, specifying colors, fonts, sizes, use of bold and italic, paragraphs, tables and the like, but not specifying the meaning of the data on the page. Web 1.0 servers sent mostly static web pages coded in HTML or XHTML to browsers that rendered the pages on the screen. Web 2.0 applications are more dynamic, generally enabling significant interaction between the user (the client) and the computer (the server), and among communities of users.

Computers have a hard time deciphering meaning from XHTML content. The web today involves users' interpretations of what pages and images mean, but the future entails a shift from XHTML to a more sophisticated system based on XML, enabling computers to better understand meaning. Web 2.0 companies use "data mining" to extract as much meaning as they can from XHTML-encoded pages. For example, Google's AdSense contextual advertising program does a remarkable job placing relevant ads next to content based on some interpretation of the meaning of that content. XHTML-encoded content does not explicitly convey meaning, but XML-encoded content does. So if we can encode in XML (and derivative technologies) much or all of the content on the web, we'll take a great leap forward towards realizing the Semantic Web. It is unlikely that web developers and users will directly encode all web content in XML—it's simply too tedious and probably too complex for most web designers. Rather, the XML encoding will occur naturally as a by-product of using various content creation tools. For example, to submit a resume on a website, there may be a tool that enables the user to fill out a form (with first name, last name, phone number, career goal, etc.). When the resume is submitted, the tool could create a computer readable microformat that could easily be found and read by applications that process resumes. Such tools might help a company find qualified potential employees, or help a job seeker who wants to write a resume find resumes of people with similar qualifications).

Tagging and Folksonomies

Tagging and folksonomies are early hints of a "web of meaning." Without tagging, searching for a picture on Flickr would be like searching for a needle in a giant haystack. Flickr's tagging system allows users to subjectively tag pictures with meaning, making photos findable by search engines. Tagging is a "loose" classification system, quite different, for example, from using the Dewey Decimal System for cataloging books, which follows a rigid taxonomy system, limiting your choices to a set of predetermined categories. Tagging is a more "democratic" labeling system that allows people, for example, to associate whatever meanings they choose with a picture (e.g. who is in the picture, where it was taken, what is going on, the colors, the mood, etc.).

Semantic Web

Many people consider the Semantic Web to be the next generation in web development, one that helps to realize the full potential of the web. This is Tim Berners-Lee's original vision of the web, also known as the "web of meaning." Though Web 2.0 applications are finding meaning in content, the Semantic Web will attempt to make those meanings clear to computers as well as humans. It will be a web able to answer complex and subtle questions.

Realization of the Semantic Web depends heavily on XML and XML-based technologies (see Chapter 12), which help make web content more understandable to computers. Currently, computers "understand" data on basic levels, but are progressing to find meaningful connections and links between data points. The emerging Semantic Web technologies highlight new relationships among web data. Some experiments that emphasize this are Flickr and FOAF (Friend of a Friend), a research project that "is creating a Web of machine-readable pages describing people, the links between them and the things they create and do." Programming in both instances involves links between databases ultimately allowing users to share, transfer, and use each other's information (photos, blogs,etc.).

Preparations for the Semantic Web have been going on for years. XML is already widely used in both online and offline applications, but still only a minute portion of the web is coded in XML or derivative technologies. Many companies, including Zepheira, an information management company, and Joost, an Internet TV provider, already use semantic technologies in working with data. Deterring Semantic Web development are concerns about the consequences of false information and the abuse of data. Since the Semantic Web will rely on computers having greater access to information and will yield a deeper understanding of its significance, some people worry about the potentially increased consequences of security breaches. The **Policy Aware Web Project** is an early attempt at developing standards to encourage data sharing by providing access policies that can sufficiently protect individuals' privacy concerns.

Microformats

Some people look at the web and see lots of "loose" information. Others see logical aggregates, such as business cards, resumes, events and so forth. Microformats are standard formats for representing information aggregates that can be understood by computers, enabling better search results and new types of applications. The key is for developers to use standard microformats, rather than developing customized, non-standard data aggregations. Microformat standards encourage sites to similarly organize their information, thus increasing interoperability. For example, if you want to create an event or an events calendar, you could use the hCalendar microformat. Some other microformats are adr for address information, hresume for resumes, and xfolk for collections of bookmarks.

Resource Description Framework (RDF)

The Resource Description Framework (RDF), developed by the World Wide Web Consortium (W3C), is based on XML and used to describe content in a way that is understood by computers. RDF helps connect isolated databases across the web with consistent semantics. The structure of any expression in RDF is a collection of triples.148 RDF triples consist of two pieces of information (subject and object) and a linking fact (predicate). Let's create a simple RDF triple. "Chapter 1, Dive Into® Web 2.0" is the title of this document and one property (the document's subject) that we'll use in our RDF triple. Another property of this chapter is "Deitel" as the author. So the sentence "Chapter 1, Dive Into® Web 2.0 is written by Deitel" is an RDF triple, containing two properties and a linking fact ("is written by").

DBpedia.org is currently transferring content into RDF from Wikipedia, one of the largest and most popular resources of online information. Using SPARQL (SPARQL Protocol and RDF Query Language), DBpedia.org is converting data from Wikipedia entries into RDF triples. In June 2007, they claimed to have over 91 million triples—this will allow the information (fromWikipedia) to be accessed bymore advanced search queries.

Ontologies

Ontologies are ways of organizing and describing related items, and are used to represent semantics. This is another means of cataloging Internet content in a way that can be understood by computers. RDF is designed for formatting ontologies. OWL (Web Ontology Language), also designed for formatting ontologies in XML, extends beyond the basic semantics of RDF ontologies to enable even deeper machine understanding of content.

Closing Comment

This book will get you up to speed on Web 2.0 applications development. Building a "web of meaning" will ultimately open a floodgate of opportunities for web developers and entrepreneurs to write new applications, create new kinds of businesses, etc. We don't know exactly what the "web of meaning" will look like, but it's starting to take shape. If it helps accomplish what many leaders in the web community believe is possible, the future of the web will be exciting indeed.

UNIT-IV

Rich Internet Applications (RIAS) with Adobe Flash

Adobe Flash Introduction:

Adobe **Flash CS3** (Creative Suite 3) is a commercial application that you can use to produce interactive, animated **movies**. Flash can be used to create web-based banner advertisements, interactive websites, games and web-based applications with stunning graphics and multimedia effects. It provides tools for drawing graphics, generating animations, and adding sound and video. Flash movies can be embedded in web pages, distributed on CDs and DVDs as independent applications, or converted into stand-alone, executable programs. Flash includes tools for coding in its scripting language—ActionScript 3.0—which is similar to JavaScript and enables interactive applications. A fully functional, 30-day trial version of Flash CS3 is available for download from:

To follow along with the examples in this chapter, please install this software before continuing. Follow the on-screen instructions to install the trial version of the Flash software. To play Flash movies, the Flash Player plug-in must be installed in your web browser. The most recent version of the plug-in (at the time of this writing) is version 9. You can download the latest version from:

According to Adobe's statistics, approximately 98.7 percent of web users have Flash Player version 6 or greater installed, and 83.4 percent of web users have Flash Player version 9 installed.1 There are ways to detect whether a user has the appropriate plug-in to view Flash content. Adobe provides a tool called the Flash PlayerDetection Kit which contains files that work together to detect whether a suitable version of Adobe Flash Player is installed in a user's web browser. This kit can be downloaded from:
www.adobe.com/products/flashplayer/download/detection_kit/
This chapter introduces building Flash movies. You'll create interactive buttons, add sound to movies, create special graphic effects and integrate ActionScript in movies.

Flash Movie Development:

Once Flash CS3 is installed, open the program. Flash's Welcome Screen appears by default. The Welcome Screen contains options such as Open a Recent Item, Create New and Create from Template. The bottom of the page contains links to useful help topics and tutorials. [Note: For additional help, refer to Flash's Help menu.]
To create a blank Flash document, click Flash File under the Create New heading. Flash opens a new file called Untitled-1 in the Flash development environment.

At the center of the development environment is the movie stage—the white area in which you place graphic elements during movie development. Above the stage is the timeline,which represents the time period over which a movie runs. The timeline is divided into increments called frames, represented by gray and white rectangles. Each frame depicts a moment in time during the movie, into which you can insert movie elements. The playhead indicates the current frame.

Common Programming Error:

Elements placed off stage can still appear if the user changes the aspect ratio of the movie. If anelement should not be visible, use an alpha of 0% to hide the element.The development environment contains several windows that provide options andtools for creating Flash movies. Many of these tools are located in the Tools bar, the verticalwindow located at the left side of the development environment. The Tools bar(Fig. 14.2) is divided into multiple sections, each containing tools and functions that helpyou create Flash movies. The tools near the top of the Tools bar select, add and removegraphics from Flash movies. The Hand and Zoom tools allow you to pan and zoom in thestage. Another section of tools provides colors for shapes, lines and filled areas. The lastsection contains settings for the active tool (i.e., the tool that is highlighted and in use).You can make a tool behave differently by selecting a new mode from the options sectionof the Tools bar.

Application windows called panels organize frequently used movie options. Paneloptions modify the size, shape, color, alignment and effects associated with a movie'sgraphic elements. By default, panels line the right and bottom edges of the window.

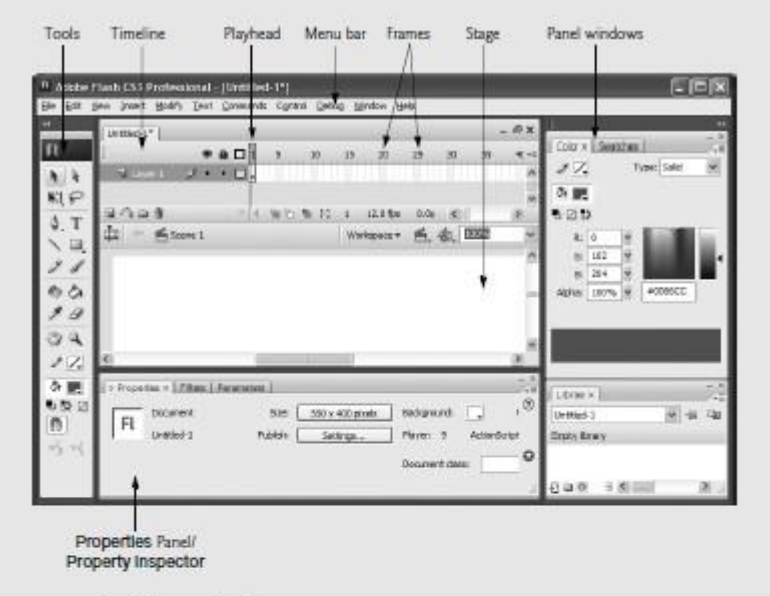


Fig. 14.1 | Flash CS3 development environment.

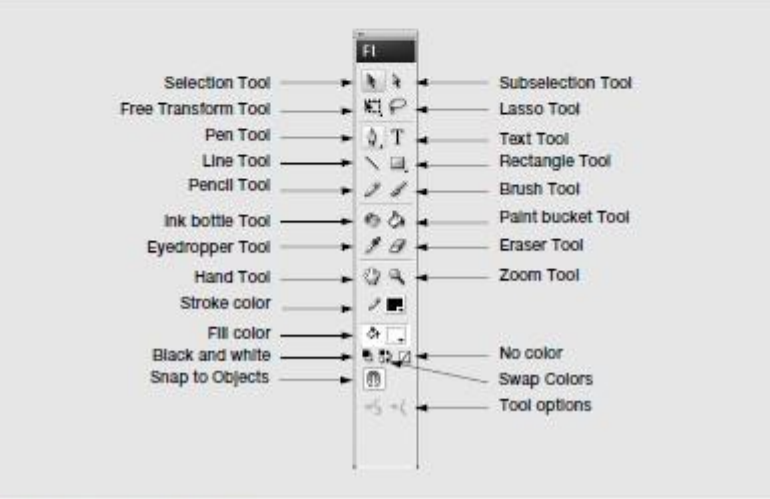


Fig. 14.2 | CS3 **Tools** bar.

Learning Flash with Hands-On Examples:

Now you'll create several complete Flash movies. The first example demonstrates how to create an interactive, animated button. ActionScript code will produce a random text string each time the button is clicked. To begin, create a new Flash movie. First, select File > New. In the New Document dialog (Fig. 14.3), select Flash File (ActionScript 3.0) under the General tab and click OK. Next, choose File > Save As… and save the movie as Ceo- Assistant.fla. The .fla file extension is a Flash-specific extension for editable movies.

Right click the stage to open a menu containing different movie options. Select Document Properties… to display the Document Properties dialog (Fig. 14.4). This dialog can also be accessed by selecting Document… from the Modify menu. Settings such as the Frame rate, Dimensions and Background color are configured in this dialog.

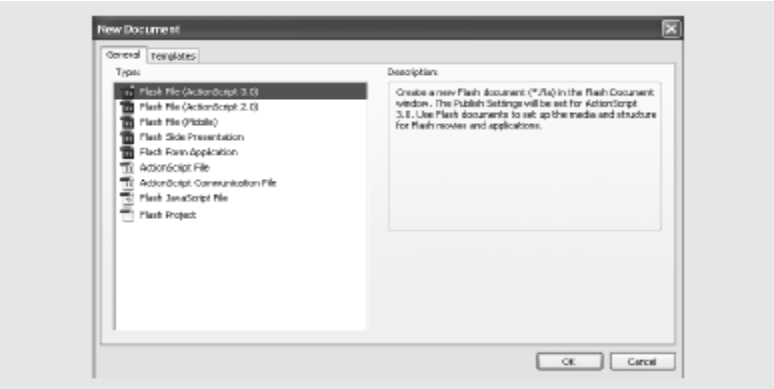


Fig. 14.3 | New Document dialog.

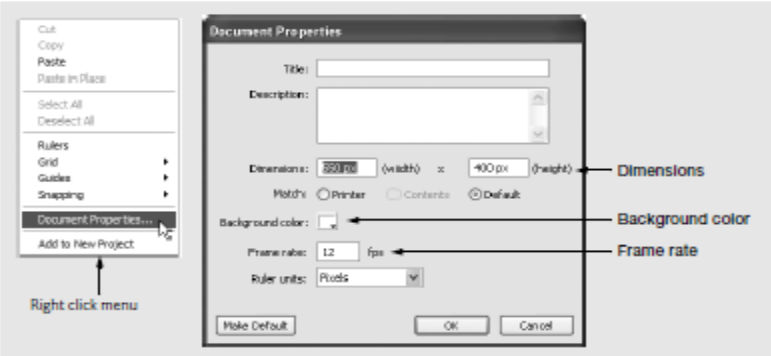


Fig. 14.4 | Document Properties dialog.

The Frame rate sets the speed at which movie frames display. A higher frame rate causes more frames to be displayed in a given unit of time (the standard measurement is seconds), thus creating a faster movie. The frame rate for Flash movies on the web is generally between 12 and 60 frames per second (fps). Flash's default frame rate is 12 fps. For this example, set the Frame Rate to 10 frames per second.

Performance Tip

Higher frame rates increase the amount of information to process, and thus increase the movie's processor usage and file size. Be especially aware of file sizes when catering to low bandwidth web users.

The background color determines the color of the stage. Click the background-color box (called a swatch) to select the background color. A new panel opens, presenting a websafe palette. Web-safe palettes and color selection are discussed in detail in Chapter 3. Note that the mouse pointer changes into an eyedropper, which indicates that you may select a color. Choose a light blue color (Fig. 14.5).

The box in the upper-left corner of the dialog displays the new background color. The hexadecimal notation for the selected color appears to the right of this box. The hexadecimal notation is the color code that a web browser uses to render color.

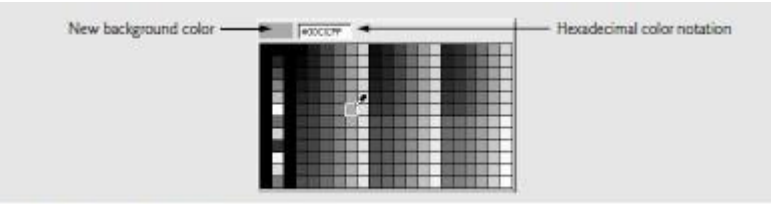


Fig. 14.5 | Selecting a background color.

Dimensions define the size of the movie as it displays on the screen. For this example, set the movie width to 200 pixels and the movie height to 180 pixels. Click OK to apply the changes in the movie settings.

The next step is to modify the shape's color. We'll apply a gradient fill—a gradual progression of color that fills the shape. Open the Swatches panel (Fig. 14.10), either by selecting Swatches from the Window menu or by pressing <Ctrl>-F9.

The Swatches panel provides four radial gradients and three linear gradients, although you also can create and edit gradients with the Color panel.

Click outside the circle with the Selection tool to deselect the circle. Now, select only the red fill with the Selection tool. Change the fill color by clicking the red radial gradient fill in the Swatches panel. The gradient fills are located at the bottom of the Swatches panel (Fig. 14.10). The circle should now have a red radial gradient fill with a black stroke surrounding it.

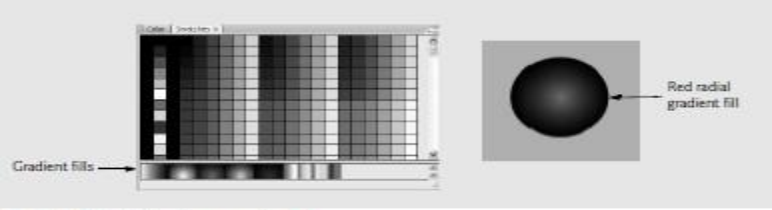


Fig. 14.10 | Choosing a gradient fill.

Converting a Shape into a Symbol:
A Flash movie consists of scenes and symbols. Each scene contains all graphics and symbols. The parent movie may contain several symbols that are reusable movie elements, such as graphics, buttons and movie clips. A scene timeline can contain numerous symbols, each with its own timeline and properties. A scene may have several instances of any given symbol (i.e., the same symbol can appear multiple times in one scene). You can edit symbols independently of the scene by using the symbol's editing stage. The editing stage is separate from the scene stage and contains only one symbol.

To make our button interactive, we must first convert the button into a button symbol. The button consists of distinct text, color fill and stroke elements on the parent. stage. These items are combined and treated as one object when the button is converted into a symbol. Use the Selection tool to drag a selection box around the button, selecting the button fill, the button stroke and the text all at one time
.
Now, select Convert to Symbol… from the Modify menu or use the shortcut F8 on the keyboard. This opens the Convert to Symbol dialog, in which you can set the properties of a new symbol.

Every symbol in a Flash movie must have a unique name. It is a good idea to name symbols by their contents or function, because this makes them easier to identify and reuse. Enter the name go button into the Name field of the Convert to Symbol dialog. The Behavior option determines the symbol's function in the movie.

You can create three different types of symbols—movie clips, buttons and graphics.A movie clip symbol's behavior is similar to that of a scene and thus it is ideal for recurring animations. Graphic symbols are ideal for static images and basic animations. Button symbols are objects that perform button actions, such as rollovers and hyperlinking. A rollover is an action that changes the appearance of a button when the mouse passes over it. For this example, select Button as the type of symbol and click OK. The button should now be surrounded by a blue box with crosshairs in the upper-left corner, indicating that the button is a symbol. Also, in the Properties window panel, name this instance of the go button symbol goButton in the field containing <Instance Name>. Use the selection tool to drag the button to the lower-right corner of the stage.

The Library panel (Fig. 14.15) stores every symbol present in a movie and is accessed through the Window menu or by the shortcuts <Ctrl>-L or F11. Multiple instances of a symbol can be placed in a movie by dragging and dropping the symbol from the Library panel onto the stage.

The Movie Explorer displays the movie structure and is accessed by selecting Movie Explorer from the Window menu or by pressing <Alt>-F3 (Fig. 14.16). The Movie Explorer panel illustrates the relationship between the current scene (Scene
1) and its symbols.

Fig. 14.13 | Selecting an object with the selection tool.

Fig. 14.14 | Creating a new symbol with the **Convert to Symbol** dialog.

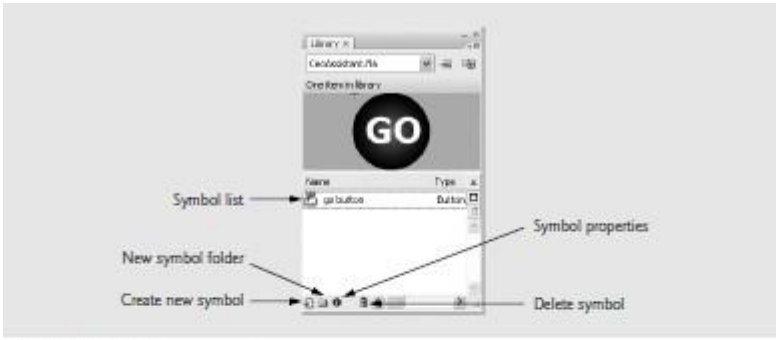


Fig. 14.15 | **Library** panel.

Fig. 14.16 | **Movie Explorer** for `CeoAssistant.fla`.

Editing Button Symbols

The next step in this example is to make the button symbol interactive. The different components of a button symbol, such as its text, color fill and stroke, may be edited in the symbol's editing stage, which you can access by double clicking the icon next to the symbol in the Library. A button symbol's timeline contains four frames, one for each of the button states (up, over and down) and one for the hit area.

The up state (indicated by the Up frame on screen) is the default state before the user presses the button or rolls over it with the mouse. Control shifts to the over state (i.e., the Over frame) when the user rolls over the button with the mouse cursor. The button's down state (i.e., the Down frame) plays when a user presses a button. You can create interactive, user- responsive buttons by customizing the appearance of a button in each of these states. Graphic elements in the hit state (i.e., the Hit frame) are not visible to a viewer of the movie; they exist simply to define the active area of the button (i.e., the area that can be clicked).

By default, buttons have only the up state activated when they are created. You may activate other states by adding keyframes to the other three frames. Keyframes for a button, discussed in the next section, determine how a button reacts when it is rolled over or clicked with the mouse.

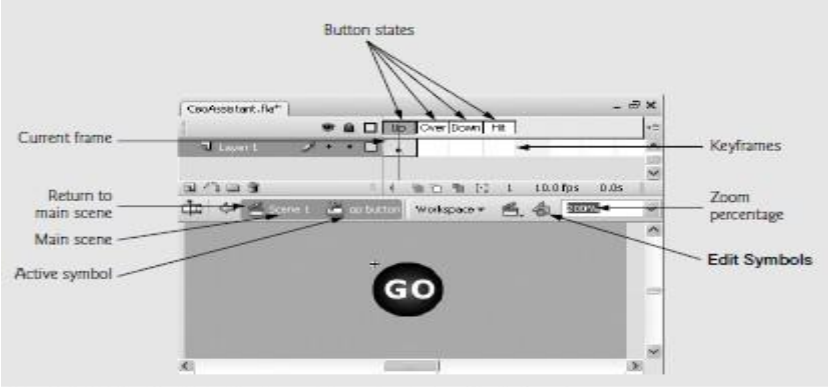


Fig. 14.17 | Modifying button states with a button's editing stage.

Adding Keyframes:

Keyframes are points of change in a Flash movie and appear in the timeline with a dot. By adding keyframes to a button symbol's timeline, you can control how the button reacts to user interactions. The following step shows how to create a button rollover effect, which is accomplished by inserting a keyframe in the button's Over frame, then changing the button's appearance in that frame. Right click the Over frame and select Insert Keyframe from the resulting menu or press F6.

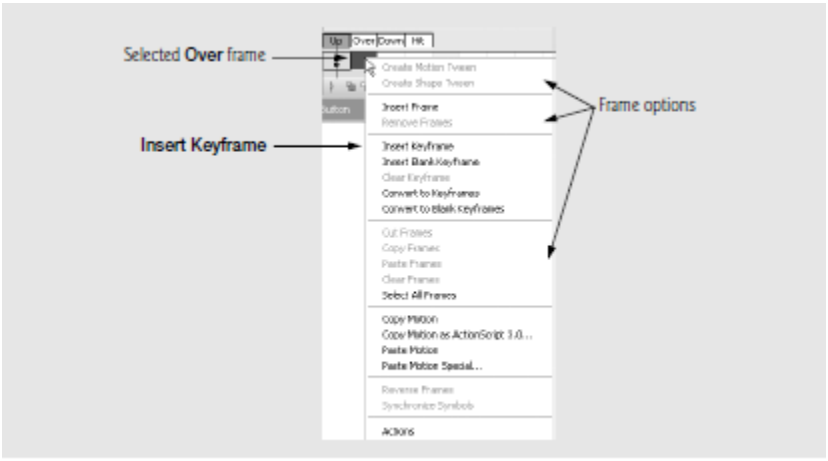


Fig. 14.18 | Inserting a keyframe.

Select the Over frame and click outside the button area with the selection tool to deselect the button's components. Change the color of the button in the Over state from red gradient fill to green gradient fill by selecting only the fill portion of the button with the Selection tool. Select the green gradient fill in the Swatches panel to change the color of the button in the Over state. Changing the color of the button in the over state does not affect the color of the button in the up state. Now, when the user moves the cursor over the button (in the up state) the button animation is replaced by the animation in the Over state. Here, we change only the button's color, but we could have created an entirely new animation in the Over state. The button will now change from red to green when the user rolls over the button with the mouse. The button will return to red when the mouse is no longer positioned over the button.

Publishing Your Flash Movie:

Flash movies must be published for users to view them outside the Flash CS3 environment and Flash Player. This section discusses the more common methods of publishing Flash movies. For this example, we want to publish in two formats, Flash and Windows Projector, which creates a standard Windows-executable file that works even if the user hasn't installed Flash. Select Publish Settings… from the File menu to open the Publish Settings dialog. Select the Flash, HTML and Windows Projector checkboxes and uncheck all the others. Then click the Flash tab at the top of the dialog. This section of the dialog allows you to choose the Flash settings. Flash movies may be published in an older Flash version if you wish to support older Flash Players.

Note that ActionScript 3.0 is not supported by older players, so choose a version with care. Publish the movie by clicking Publish in the Publish Settings dialog or by selecting Publish from the File menu. After you've published the movie, the directory in which you saved the movie will have several new files (Fig. 14.30). If you wish to place your movie on a website, be sure to copy the HTML, JavaScript and SWF files to your server.

Good Programming Practice:
It is not necessary to transfer the .fla version of your Flash movie to a web server unless you want other users to be able to download the editable version of the movie.

As we can see in the Ceo Assistant 1.0 example, Flash is a feature-rich program. We have only begun to use Flash to its full potential. ActionScript can create sophisticated programs and interactive movies. It also enables Flash to interact with ASP.NET (Chapter 21), PHP (Chapter 19), and JavaScript (Chapters 4–9), making it a program that integrates smoothly into a web environment.

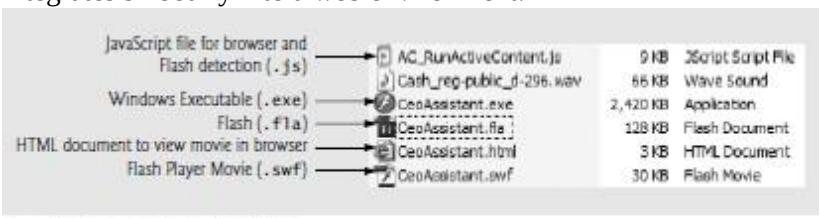


Fig. 14.30 | Published Flash files.

Creating Special Effects with Flash:

The following sections introduce several Flash special effects. The preceding example familiarized you with basic movie development. The next sections cover many additional topics, from importing bitmaps to creating splash screens that display before a web page loads.

Importing and Manipulating Bitmaps
Some of the examples in this chapter require importing bitmapped images and other media into a Flash movie. The importing process is similar for all types of media, including images, sound and video. The following example shows how to import an image into a Flash movie.

Begin by creating a new Flash document. The image we are going to import is located in the Chapter 14 examples folder. Select File > Import > Import to Stage… (or press <Ctrl>-R) to display the Import dialog. Browse to the folder on your system containing this chapter's examples and open the folder labeled images. Select bug.bmp and click OK to continue. A bug image should appear on the stage. The Library panel stores imported images. You can convert imported images into editable shapes by selecting the image and pressing <Ctrl>-B or by choosing Break Apart from the Modify menu. Once an imported image is broken apart, it may be shape tweened or edited with editing tools, such as the Lasso, Paint bucket, Eraser and Paintbrush. The editing tools are found in the toolbox and apply changes to a shape. Dragging with the Lasso tool selects areas of shapes.

The color of a selected area may be changed or the selected area may be moved. Click and drag with the Lasso tool to draw the boundaries of the selection. As with the button in the last example, when you select a shape area, a mesh of white dots covers the selection. Once an area is selected, you may change its color by selecting a new fill color with the fill swatch or by clicking the selection with the Paint bucket tool. The Lasso tool has different options (located in the Options section of the toolbox) including Magic wand and Polygon mode. The Magic wand option changes the Lasso tool into the Magic wand tool, which selects areas of similar colors. The polygonal lasso selects straight-edged areas.

The Eraser tool removes shape areas when you click and drag the tool across an area. You can change the eraser size using the tool options. Other options include settings that make the tool erase only fills or strokes.
The Brush tool applies color in the same way that the eraser removes color. The paintbrush color is selected with the fill swatch. The paintbrush tool options include a Brush mode option. These modes are Paint behind, which sets the tool to paint only in areas with no color information; Paint selection, which paints only areas that have

been selected; and Paint inside, which paints inside a line boundary.

Each of these tools can create original graphics. Experiment with the different tools to change the shape and color of the imported bug graphic.

Creating an Advertisement Banner with Masking:

Masking hides portions of layers. A masking layer hides objects in the layers beneath it, revealing only the areas that can be seen through the shape of the mask. Items drawn on a masking layer define the mask's shape and cannot be seen in the final movie. The next example, which builds a website banner, shows how to use masking frames to add animation and color effects to text.
Create a new Flash document and set the size of the stage to 470 pixels wide by 60 pixels high. Create three layers named top, middle and bottom according to their positions in the layer hierarchy. These names help track the masked layer and the visible layers. The top layer contains the mask, the middle layer becomes the masked animation and the bottom layer contains an imported bitmapped logo. Import the graphic bug_apple.bmp (from the images folder in this chapter's examples folder) into the first frame of the top layer, using the method described in the preceding section. This image will appear too large to fit in the stage area. Select the image with the selection tool and align it with the upper-left corner of the stage. Then select the Free transform tool in the toolbox .

The Free transform tool allows us to resize an image.When an object is selected with this tool, anchors appear around its corners and sides. Click and drag an anchor to resize the image in any direction. Holding the Shift key while dragging a corner anchor ensures

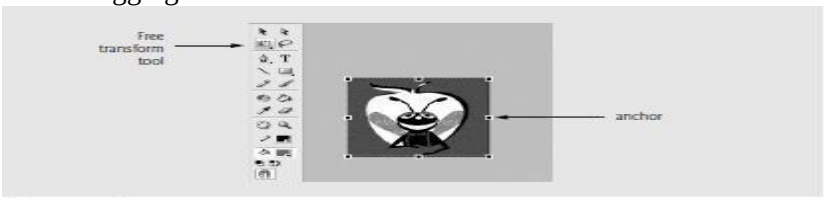


Fig. 14.31 | Resizing an image with the Free transform tool.

that the image maintains the original height and width ratio. Hold down the Shift key while dragging the lower-right anchor upward until the image fits on the stage.

Use the text tool to add text to frame 1 of the top layer. Use Verdana, 28 pt bold, as the font. Select a blue text color, and make sure that Static Text is selected in the Properties window. Type the banner text "Deitel and Associates", making sure that the text fits inside the stage area, and use the Selection tool to position the text next to the image. This text becomes the object that masks an animation.

We must convert the text into a shape before using it as a mask. Click the text field with the Selection tool to ensure that it is active and select Break Apart twice from the Modify menu. Breaking the text apart once converts each letter into its own text field. Breaking it apart again converts the letters into shapes that cannot be edited with the text tool, but can be manipulated as regular graphics.

Copy the contents of the top layer to the bottom layer before creating themask, so that the text remains visible when the mask is added. Right click frame 1 of the top layer, and select Copy Frames from the resulting menu. Paste the contents of the top layer into frame 1 of the bottom layer by right clicking frame 1 of the bottom layer and selecting Paste Frames from the menu. This shortcut pastes the frame's contents in the same positions as the original frame. Delete the extra copy of the bug image by selecting the bug image in the top layer with the selection tool and pressing the Delete key.

Next, you'll create the animated graphic that the banner text in the top layer masks. Click in the first frame of the middle layer and use the Oval tool to draw a circle to the left of the image that is taller than the text. The oval does not need to fit inside the banner area. Set the oval stroke to no color by clicking the stroke swatch and selecting the No color option. Set the fill color to the rainbow gradient (Fig. 14.32), found at the bottom of the Swatches panel.

Select the oval by clicking it with the Selection tool, and convert the oval to a symbol by pressing F8. Name the symbol oval and set the behavior to Graphic. When the banner is complete, the oval will move across the stage; however, it will be visible only through the text mask in the top layer. Move the oval just outside the left edge of the stage, indicating the point at which the oval begins its animation. Create a keyframe in frame 20 of the middle layer and another in frame 40. These keyframes indicate the different locations of the oval symbol during the animation.

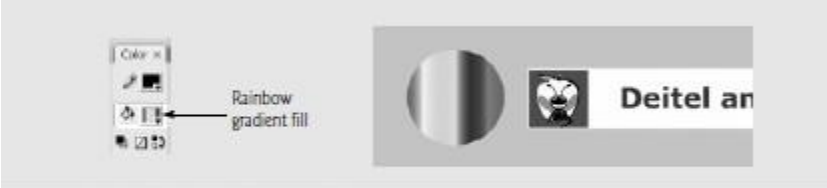


Fig. 14.32 | Creating the oval graphic.

Click frame 20 and move the oval just outside the right side of the stage to indicate the animation's next key position. Do not move the position of the oval graphic in frame 40, so that the oval will return to its original position at the end of the animation. Create the first part of the animation by right clicking frame 1 of the middle layer and choosing Create Motion Tween from the menu. Repeat this step for frame 20 of the middle layer, making the oval symbol move from left to right and back. Add key frames to frame 40 of both the top and bottom layers so that the other movie elements appear throughout the movie.

Now that all the supporting movie elements are in place, the next step is to apply the masking effect. To do so, right click the top layer and select Mask (Fig. 14.33). Adding a mask to the top layer masks only the items in the layer directly below it (the middle layer), so the bug logo in the bottom layer remains visible at all times. Adding a mask also locks the top and middle layers to prevent further editing.

Now that the movie is complete, save it as banner.fla and test it with the Flash Player. The rainbow oval is visible through the text as it animates from left to right. The text in the bottom layer is visible in the portions not containing the rainbow (Fig. 14.34).

Adding Online Help to Forms

In this section, we build on Flash techniques introduced earlier in this chapter, including tweening, masking, importing sound and bitmapped images, and writing ActionScript. In the following example, we apply these various techniques to create an online form that offers interactive help. The interactive help consists of animations that appear when a user presses buttons located next to the form fields. Each button contains a script that triggers an animation, and each animation provides the user with information regarding the form field that corresponds to the pressed button.
Each animation is a movie-clip symbol that is placed in a separate frame and layer of the scene. Adding a stop action to frame 1 pauses the movie until the user presses a button.

Begin by creating a new movie, using default movie size settings. Set the frame rate to 24 fps. The first layer will contain the site name, form title and form captions. Change the name of Layer 1 to text. Add a stop action to frame 1 of the text layer. Create the site name Bug2Bug.com as static text in the text layer using a large, bold font, and place the title at the top of the page. Next, place the form name Registration Form as static text beneath the site name, using the same font, but in a smaller size and different color. The final text element added to this layer is the text box containing the form labels. Create a text box using the Text Tool, and enter the text: Name:, Member #: and Password:, pressing Enter after entering each label to put it on a different line. Next, adjust the value of the Line Spacing field (the amount of space between lines of text) found by clicking the Edit Format Options button ( ) in the Properties window. Change the form field caption line spacing to 22 in the Format Options dialog (Fig. 14.35) and set the text alignment (found in the Properties window) to right justify.

Now we'll create the form fields for our help form. The first step in the production of these form fields is to create a new layer named form. In the form layer, draw a rectangle that is roughly the same height as the caption text. This rectangle will serve as a background for the form text fields (Fig. 14.36).We set a Rectangle corner radius of 6 px in the Properties panel. Feel free to experiment with other shapes and colors.

The next step is to convert the rectangle into a symbol so that it may be reused in the movie. Select the rectangle fill and stroke with the selection tool and press F8 to convert the selection to a symbol. Set the symbol behavior to Graphic and name the symbol form field.

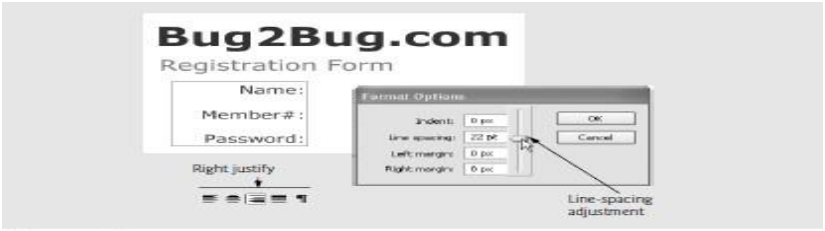


Fig. 14.35 | Adjusting the line spacing with the **Format Options** dialog.

Fig. 14.36 | Creating a rectangle with rounded corners.

This symbol should be positioned next to the Name: caption. When the symbol is in place, open the Library panel by pressing <Ctrl>-L, select the **form** layer and drag two copies of the **form field** symbol from the **Library** onto the stage. This will create two new instances of this symbol. Use the Selection tool to align the fields with their corresponding captions. For more precise alignment, select the desired object with the Selection tool and press the arrow key on the keyboard in the direction you want to move the object. After alignment of the form field symbols, the movie should resemble Fig. 14.37.

We now add input text fields to our movie. An input text field is a text field into which the user can enter text. Select the Text tool and, using the Properties window, set the font to Verdana, 16 pt, with dark blue as the color. In the Text type pull-down menu in the Properties window, select Input Text (Fig. 14.38). Then, click and drag in the stage to create a text field slightly smaller than the form field symbol we just created. With the Selection tool, position the text field over the instance of the form field symbol associated with the name. Create a similar text field for member number and password. Select the Password text field, and select Password in the Line type pull-down menu in the Properties window. Selecting Password causes any text entered into the field by the user to appear as an asterisk (*). We have now created all the input text fields for our help form.

Fig. 14.37 | Creating multiple instances of a symbol with the **Library** panel.

Fig. 14.38 | Input and password text-field creation.

In this example, we won't actually process the text entered into these fields. Using ActionScript, we could give each input text field a variable name, and send the values of these variables to a server-side script for processing.

Now that the form fields are in place, we can create the help associated with each field. Add two new layers. Name one layer button and the other labels. The labels layer will hold the frame label for each keyframe. A frame label is a text string that corresponds to a specific frame or series of frames. In the labels layer, create keyframes in frames 2, 3 and 4. Select frame 2 and enter name into the Frame field in the Properties window.

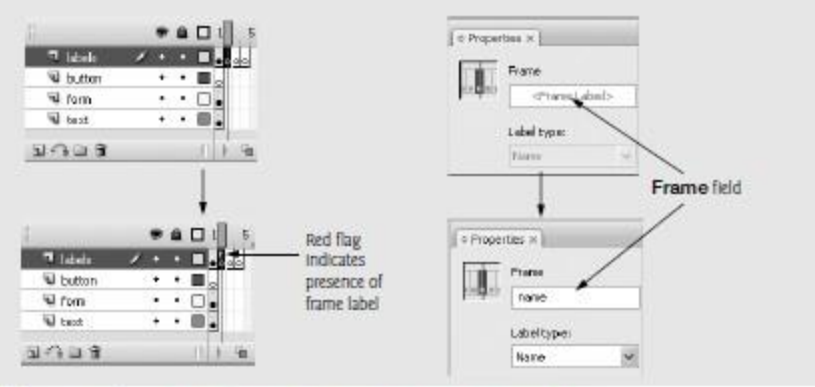


Fig. 14.39 | Adding **Frame Labels** using the **Properties window**.

The movie is now complete. Press <Ctrl>-Enter to preview it with the Flash Player. If the triggered animations do not appear in the correct locations, return to the scene and adjust their position. The final movie is displayed in Fig. 14.46. In our example, we have added a picture beneath the text layer. Movies can be enhanced in many ways, such as by changing colors and fonts or by adding pictures. Our movie (bug2bug.fla) can be found in the this chapter's examples directory. If you want to use our symbols to recreate the movie, select Open External Library… from the Import submenu of the File menu and open bug2bug.fla. The Open External Library… option allows you to reuse symbols from another movie.

Fig. 14.46 | **Bug2Bug.com** help form.

Creating a Website Splash Screen:

Flash is becoming an important tool for e-businesses. Many organizations use Flash to create website splash screens (i.e., introductions), product demos and web applications. Others use Flash to build games and interactive entertainment in an effort to attract new visitors. However, these types of applications can take a long time to load, causing visitors— especially those with slow connections—to leave the site. One way to alleviate this problem is to provide visitors with an animated Flash introduction that draws and keeps their attention. Flash animations are ideal for amusing visitors while conveying information as the rest of a page downloads "behind the scenes."

A preloader or splash screen is a simple animation that plays while the rest of the web page is loading. Several techniques are used to create animation preloaders. The following example creates an animation preloader that uses ActionScript to pause the movie at a particular frame until all the movie elements have loaded. To start building the animation preloader, create a new Flash document. Use the default size, and set the background color to a light blue. First, you'll create the movie pieces that will be loaded later in the process. Create five new layers, and rename Layer 2 to C++, Layer 3 to Java and Layer 4 to IW3. Layer 5 will contain the movie's ActionScript, so rename it actions. Because Layer 1 contains the introductory animation, rename this layer animation.

The preloaded objects we use in this example are animated movie clip symbols. Create the first symbol by clicking frame 2 of the C++ layer, inserting a keyframe, and creating a new movie-clip symbol named cppbook. When the symbol's editing stage opens, import the image cpphtp.gif (found in the images folder with this chapter's examples). Place a keyframe in frame 20 of Layer 1 and add a stop action to this frame. The animation in this example is produced with the motion tween Rotate option, which causes an object to spin on its axis.

Create a motion tween in frame 1 with the Properties window, setting the Rotate option to CCW (counterclockwise) and the times field to 2 (Fig. 14.47). This causes the image cpphtp.gif to spin two times counterclockwise over a period of 20 frames.

After returning to the scene, drag and drop a copy of the cppbook symbol onto the stage in frame 2 of the C++ layer. Move this symbol to the left side of the stage. Insert a frame in frame 25 of the C++ layer.

Build a similar movie clip for the Java and IW3 layers, using the files java.gif and iw3.gif to create the symbols. Name the symbol for the Java layer jbook and the IW3 symbol ibook to identify the symbols with their contents. In the main scene, create a keyframe in frame 8 of the Java layer, and place the jbook symbol in the center of the stage. Insert a frame in frame 25 of the Java layer. Insert the ibook symbol in a keyframe in frame 14 of the IW3 layer, and position it to the right of the jbook symbol. Insert a frame in frame 25 of the IW3 layer. Make sure to leave some space between these symbols so that they will not overlap when they spin (Fig. 14.48). Add a keyframe to the 25th frame of the actions layer, then add a stop to the Actions panel of that frame.

Now that the loading objects have been placed, it is time to create the preloading animation. By placing the preloading animation in the frame preceding the frame that contains the objects, we can use ActionScript to pause the movie until the objects have loaded. Begin by adding a stop action to frame 1 of the actions layer. Select frame 1 of the animation layer and create another new movie-clip symbol named loader. Use the text tool with a medium-sized sans-serif font, and place the word Loading in the center of the symbol's

Fig. 14.47 | Creating a rotating object with the motion tween **Rotate** option.

Fig. 14.48 | Inserted movie clips.

editing stage. This title indicates to the user that objects are loading. Insert a keyframe into frame 14 and rename this layerload.

Create a new layer called orb to contain the animation. Draw a circle with no stroke about the size of a quarter above the word Loading. Give the circle a green-to-white radial gradient fill color. The colors of this gradient can be edited in the Color panel (Fig. 14.49). The block farthest to the left on the gradient range indicates the innermost color of the radial gradient, whereas the block farthest to the right indicates the outermost color of the radial gradient. Click the left block to reveal the gradient color swatch. Click the swatch and select a medium green as the inner color of the gradient. Select the right, outer color box and change its color to white. Deselect the circle by clicking on a blank portion of the stage. Note that a white ring appears around the circle due to the colored background. To make the circle fade into the background, we adjust its alpha value. Alpha is a value between 0 and 100% that corresponds to a color's transparency or opacity. An alpha value of 0% appears transparent, whereas a value of 100% appears completely opaque. Select the circle again and click the right gradient box (white). Adjust the value of the Alpha field in the Color Mixer panel to 0%. Deselect the circle. It should now appear to fade into the background.

The rate of progression in a gradient can also be changed by sliding the color boxes. Select the circle again. Slide the left color box to the middle so that the gradient contains more green than transparent white, then return the slider to the far left. Intermediate colors may be added to the gradient range by clicking beneath the bar, next to one of the existing.

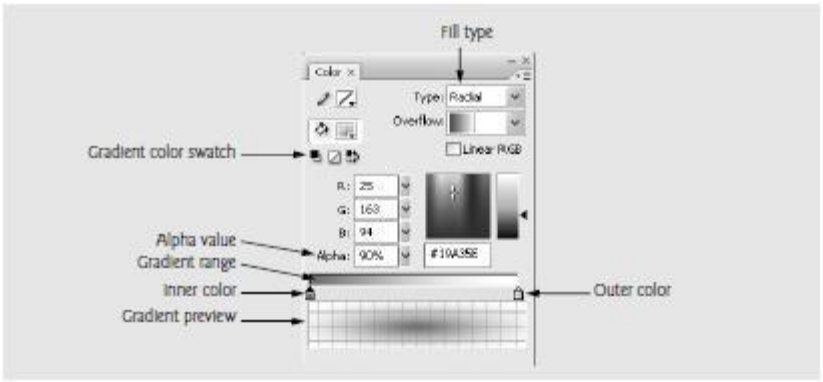


Fig. 14.49 | Changing gradient colors with the **Color** panel.

color boxes. Click to the right of the inner color box to add a new color box (Fig. 14.50). Slide the new color box to the right and change its color to a darker green. Any color box may be removed from a gradient by dragging it downward off the gradient range.

Insert keyframes into frame 7 and 14 of the orb layer. Select the circle in frame 7 with the selection tool. In the Color panel change the alpha of every color box to 0%. Select frame 1 in the Timeline and add shape tween. Change the value of the Ease field in the Properties window to –100. Ease controls the rate of change during tween animation. Negative values cause the animated change to be gradual at the beginning and become increasingly drastic. Positive values cause the animation to change quickly in the first frames, becoming less drastic as the animation progresses. Add shape tween to frame 7 and set the Ease value to 100. In frame 14, add the action gotoAndPlay(1); to repeat the animation. You can preview the animation by pressing Enter. The circle should now appear to pulse.

Before inserting the movie clip into the scene, we are going to create a hypertext linked button that will enable the user to skip over the animations to the final destination. Add a new layer called link to the loader symbol with keyframes in frames 1 and 14. Using the text tool, place the words skip directly to Deitel website below Loading in a smaller font size. Select the words with the selection tool and convert them into a button symbol named skip. Converting the text into a button simulates a text hyperlink created with XHTML. Double click the words to open the skip button's editing stage. For this example, we are going to edit only the hit state. When a button is created from a shape, the button's hit area is, by default, the area of the shape. It is important to change the hit state of a button created from text so that it includes the spaces between the letters; otherwise, the link will work only when the user hovers over a letter's area. Insert a keyframe in the hit state. Use the rectangle tool to draw the hit area of the button, covering the entire length and height of the text. This rectangle is not visible in the final movie, because it defines only the hit area.

The button is activated by giving it an action that links it to another web page. After returning to the loader movie-clip editing stage, give the skip button the instance name skipButton and open the Actions panel for the first frame of the link layer.

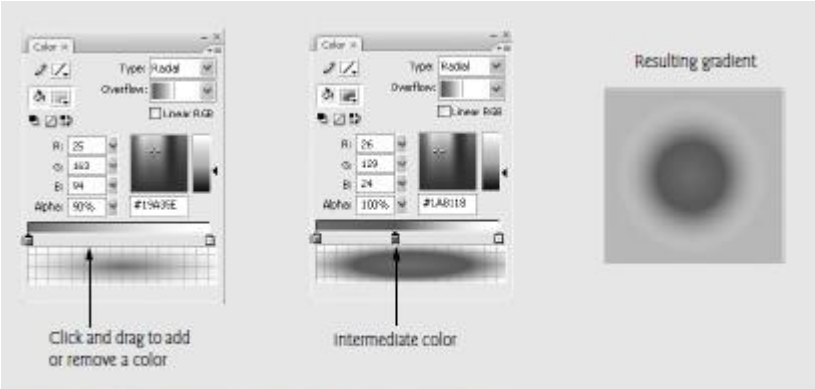


Fig. 14.50 | Adding an intermediate color to a gradient.

Fig. 14.51 | Defining the hit area of a button.

the button is clicked. Then, create an object of type URLRequest and give the constructor a parameter value of "http://www.deitel.com". The function onClick employs Flash's navigateToURL function to access the website given to it.

Thus, the code now reads

```
skipButton.addEventListener( MouseEvent.CLICK, onClick );
var url : URLRequest = new URLRequest( "http://www.deitel.com" );
function onClick( e : MouseEvent ) : void
{
navigateToURL( url, "_blank" );
} // end function onClick
```

The "_blank" parameter signifies that a new browser window displaying the Deitel website should open when the user presses the button. Return to the scene by clicking Scene 1 directly below the timeline, next to the name of the current symbol. Drag and drop a copy of the loader movie clip from the Library panel into frame 1 of the animation layer, center it on the stage, and set its Instance name to loadingClip.

The process is nearly complete. Open the Actions panel for the actions layer. The following actions direct the movie clip to play until all the scene's objects are loaded. First, add a stop to the frame so that it doesn't go to the second frame until we tell it to. Using the loadingClip movie instance, use the addEventListener function to invoke the functiononBegin whenever the event Event.ENTER_FRAME is triggered. The ENTER_FRAME event occurs every time the playhead enters a new frame. Since this movie's frame rate is 12 fps (frames per second), the ENTER_FRAME event will occur 12 times each second.

```
loadingClip.addEventListener( Event.ENTER_FRAME, onBegin );
```

The next action added to this sequence is the function onBegin. The condition of the if statement will be used to determine how many frames of the movie are loaded. Flash movies load frame by frame. Frames that contain complex images take longer to load. Flash will continue playing the current frame until the next frame has loaded. For our movie, if the number of frames loaded (frameLoaded) is equal to the total number of frames (totalFrames), then the movie is finished loading, so it will play frame 2. It also invokes the removeEventListener function to ensure that onBegin is not called for the remainder of the movie. If the number of frames loaded is less than the total number of frames, then the current movie clip continues to play. The code now reads:

```
stop();
loadingClip.addEventListener( Event.ENTER_FRAME, onBegin );
// check if all frames have been loaded
function onBegin( event : Event ) : void
{
if ( framesLoaded == totalFrames )
{
loadingClip.removeEventListener( Event.ENTER_FRAME, onBegin ); gotoAndPlay( 2 );
} // end if
} // end function onBegin
```

Create one more layer in the scene, and name the layer title. Add a keyframe to frame 2 of this layer, and use the Text tool to create a title that reads Recent Deitel Publications. Below the title, create another text hyperlink button to the Deitel website. The simplest way to do this is to duplicate the existing skip button and modify the text. Right click the skip symbol in the Library panel, and select Duplicate. Name the new button visit, and place it in frame 2 of the title layer. Label the instance visitButton, then create a keyframe in the second frame of the actions layer. Duplicate the code from the Actions panel of the first frame of the link layer in the loader symbol, and replace skipButton with visit- Button. Double click the visit button and edit the text to say visit the Deitel website. Add keyframes to each frame of the title layer and manipulate the text to create a typing effect similar to the one we created in the bug2bug example.

The movie is now complete. Test the movie with the Flash Player (Fig. 14.52). When viewed in the testing window, the loading sequence will play for only one frame because your processor loads all the frames almost instantly. Flash can simulate how amovie would appear to an online user, though. While still in the testing window, select 56K from the Download Settings submenu of the View menu. Also, select Bandwidth Profiler

from the View menu. Then select Simulate Download from the View menu or press <Ctrl>-Enter. The graph at the top of the window displays the amount of bandwidth required to load each frame.

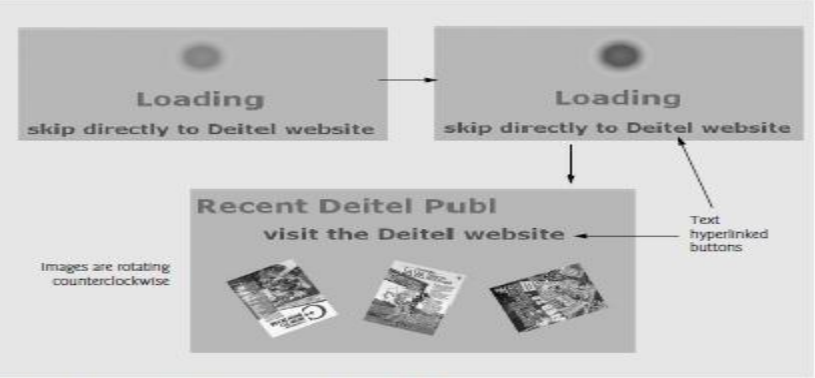


Fig. 14.52 | Creating an animation to preload images.

ActionScript:

Figure 14.53 lists common Flash Action Script 3.0 functions. By attaching these functions to frames and symbols, you can build some fairly complex Flash movies.

Function	Description
`gotoAndPlay`	Jump to a frame or scene in another part of the movie and start playing the movie.
`gotoAndStop`	Jump to a frame or scene in another part of the movie and stop the movie.
`play`	Start playing a movie from the beginning, or from wherever it has been stopped.
`stop`	Stop a movie.
`SoundMixer.stopAll`	Stop the sound track without affecting the movie.
`navigateToUrl`	Load a URL into a new or existing browser window.
`fscommand`	Insert JavaScript or other scripting languages into a Flash movie.
`Loader class`	Load a SWF or JPEG file into the Flash Player from the current movie. Can also load another SWF into a particular movie.
`framesLoaded`	Check whether certain frames have been loaded.
`addEventListener`	Assign functions to a movie clip based on specific events. The events include `load`, `unload`, `enterFrame`, `mouseUp`, `mouseDown`, `mouseMove`, `keyUp`, `keyDown` and `data`.
`if`	Set up condition statements that run only when the condition is true.
`while/do while`	Run a collection of statements while a condition statement is true.
`trace`	Display programming notes or variable values while testing a movie.
`Math.random`	Returns a random number less than or equal to 0 and less than 1.

Fig. 14.53 | Common ActionScript functions.

UNIT-V
Ajax- Enabled Rich Internet Application

Introduction:

Despite the tremendous technological growth of the Internet over the past decade, the usability of web applications has lagged behind compared to that of desktop applications. Every significant interaction in a web application results in a waiting period while the application communicates over the Internet with a server. Rich Internet Applications (RIAs) are web applications that approximate the look, feel and usability of desktop applications. RIAs have two key attributes— performance and a rich GUI.

RIA performance comes from Ajax (Asynchronous JavaScript and XML), which uses client-side scripting to make web applications more responsive. Ajax applications separate client-side user interaction and server communication, and run them in parallel, reducing the delays of server-side processing normally experienced by the user.

There are many ways to implement Ajax functionality. **"**Raw" Ajax uses JavaScript to send asynchronous requests to the server, then updates the page using the DOM (see Section 13.5). "Raw" Ajax is best suited for creating small Ajax components that asynchronously update a section of the page. However, when writing "raw" Ajax you need to deal directly with cross-browser portability issues, making it impractical for developing largescale applications. These portability issues are hidden by Ajax toolkits, such as Dojo (Section 13.8), Prototype, Script.aculo.us and ASP.NET Ajax, which provide powerful ready-to-use controls and functions that enrich web applications, and simplify JavaScript coding by making it cross-browser compatible.

Traditional web applications use XHTML forms to build simple and thin GUIs compared to the rich GUIs of Windows, Macintosh and desktop systems in general.We achieve rich GUI in RIAs with Ajax toolkits and with RIA environments such as Adobe Flex and JavaServer Faces, Such toolkits and environments provide powerful ready-to-use controls and functions that enrich web applications.
Previous chapters discussed XHTML, CSS, JavaScript, dynamic HTML, the DOM and XML. This chapter uses these technologies to build Ajax-enabled web applications.

The client-side of Ajax applications is written in XHTML and CSS, and uses JavaScript to add functionality to the user interface. XML is used to structure the data passed between the server and the client. We'll also use JSON (JavaScript Object Notation) for this purpose.

Traditional Web Applications vs. Ajax Applications:

In this section, we consider the key differences between traditional web applications and Ajax-based web applications.

Traditional Web Applications

The browser generates a request to the server, which receives the request and processes it. The server generates and sends a response containing the exact page that the browser will render, which causes the browser to load the new page (Step 4) and temporarily makes the browser window blank. Note that the client waits for the server to respond and reloads the entire page with the data from the response (Step 4). While such a synchronous request is being pro-cessed on the server, the user cannot interact with the client web page. Frequent long periods of waiting, due perhaps to Internet congestion, have led some users to refer to the World Wide Web as the "World Wide Wait." If the user interacts with and submits another form, the process begins again.

This model was originally designed for a web of hypertext documents—what some people call the "brochure web." As the web evolved into a full-scale applications platform, the model shown in Fig. 13.1 yielded "choppy" application performance. Every full-page refresh required users to re-establish their understanding of the full-page contents. Users began to demand a model that would yield the responsive feel of desktop applications.

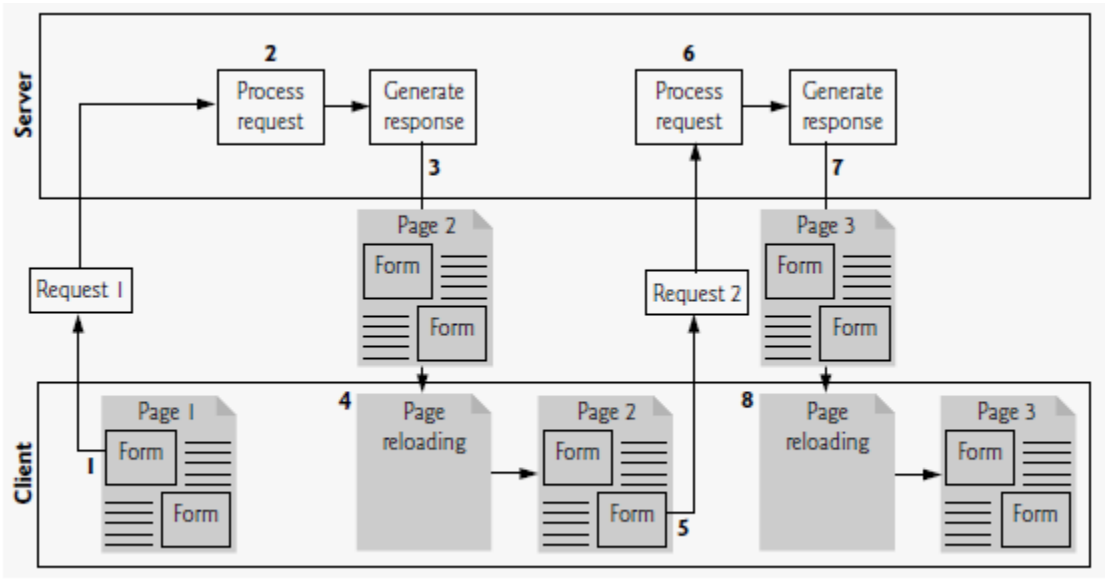


Fig. 13.1 | Classic web application reloading the page for every user interaction.

Ajax Web Applications

Ajax applications add a layer between the client and the server to manage communication between the two. When the user interacts with the page, the client creates an XMLHttpRequest object to manage a request. The XMLHttpRequest object sends the request to the server and awaits the response. The requests are **asynchronous**, so the user can continue interacting with the application on the client-side while the server processes the earlier request concurrently. Other user interactions could result in additional requests to the server. Once the server responds to the original request, the XMLHttpRequest object that issued the request calls a client-side function to process the data returned by the server. This function—known as a callback function— uses partial page updates to display the data in the existing web page without reloading the entire page. At the same time, the server may be responding to the second request and the client-side may be starting to do another partial page update. The callback function updates only a designated part of the page. Such partial page updates help make web applications more responsive, making them feel more like desktop applications. The web application does not load a new page while the user interacts with it.

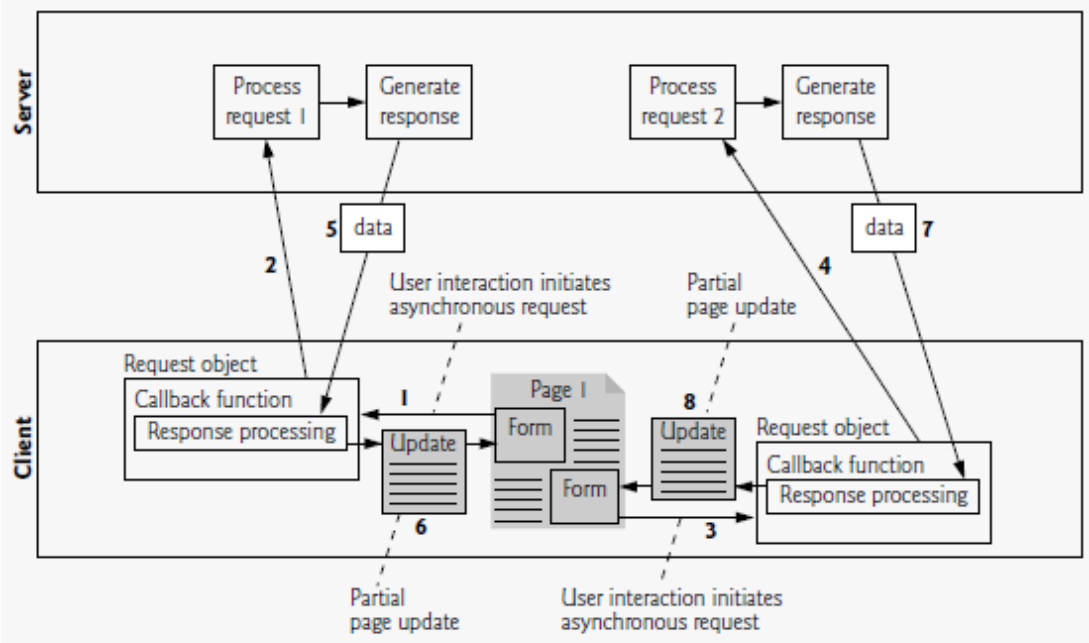


Fig. 13.2 | Ajax-enabled web application interacting with the server asynchronously.

Rich Internet Applications (RIAs) with Ajax:

Ajax improves the user experience by making interactive web applications more responsive. Consider a registration form with a number of fields (e.g., first name, last name email address, telephone number, etc.) and a Register (or Submit) button that sends the entered data to the server. Usually each field has rules that the user's entries have to follow.

When the user clicks Register, a classic XHTML form sends the server all of the data to be validated. While the server is validating the data, the user cannot interact with the page. The server finds invalid data, generates a new page identifying the errors in the form and sends it back to the client—which renders the page in the browser. Once the user fixes the errors and clicks the Register button, the cycle repeats until no errors are found, then the data is stored on the server. The entire page reloads every time the user submits invalid data.

Ajax-enabled forms are more interactive. Rather than sending the entire form to be validated, entries are validated dynamically as the user enters data into the fields. For example, consider a website registration form that requires a unique e-mail address. When the user enters an e-mail address into the appropriate field, then moves to the next form field to continue entering data, an asynchronous request is sent to the server to validate the e-mail address. If the e-mail address is not unique, the server sends an error message that is displayed on the page informing the user of the problem. By sending each entry asynchronously, the user can address each invalid entry quickly, versus making edits and resubmitting the entire form repeatedly until all entries are valid. Asynchronous requests could also be used to fill some fields based on previous.

History of Ajax:

The term Ajax was coined by Jesse James Garrett of Adaptive Path in February 2005, when he was presenting the previously unnamed technology to a client. The technologies of Ajax (XHTML, JavaScript, CSS, the DOM and XML) have all existed for many years.

Asynchronous page updates can be traced back to earlier browsers. In the 1990s, Netscape's LiveScript made it possible to include scripts in web pages that could run on the client. LiveScript evolved into JavaScript. In 1998,Microsoft introduced the XMLHttpRequest object to create and manage asynchronous requests and responses. Popular applications like Flickr and Google's Gmail use the XMLHttpRequest object to update pages dynamically. For example, Flickr uses the technology for its text editing, tagging and organizational features; Gmail continuously checks the server for new e-mail; and Google Maps allows you to drag a map in any direction, downloading the new areas on the map without reloading the entire page.

The name Ajax immediately caught on and brought attention to its component technologies. Ajax has become one of the hottest web-development technologies, enabling webtop applications to challenge the dominance of established desktop applications.

"Raw" Ajax Example Using the XMLHttp Request Object:

In this section, we use the XMLHttpRequest object to create and manage asynchronous requests. The XMLHttpRequest object (which resides on the client) is the layer between the client and the server that manages asynchronous requests in Ajax applications. This object is supported on most browsers, though they may implement it differently—a common issue in JavaScript programming. To initiate an asynchronous request (shown in Fig. 13.5), you create an instance of the XMLHttpRequest object, then use its open method to set up the request and its send method to initiate the request.

We use the onmouseover and onmouseout events (discussed in Chapter 11) to trigger events when the user moves the mouse over and out of an image, respectively. The onmouseover event calls function getContent with the URL of the document containing the book's description. The function makes this request asynchronously using an XMLHttpRequest object. When the XMLHttpRequest object receives the response, the book description is displayed below the book images. When the user moves the mouse out of the image, the onmouseout event calls function clearContent to clear the display box. These tasks are accomplished without reloading the page on the client. You can test-drive this example at test.deitel.com/examples/ajaxfp/ajax/ fig13_05/SwitchContent.html.

Asynchronous Requests

The function get Content sends the asynchronous request. Line 24 creates the XMLHttpRequest object, which manages the asynchronous request. We store the object in the global variable async Request so that it can be accessed anywhere in the script.

Line 28 calls the XMLHttpRequest open method to prepare an asynchronous GET request. In this example, the url parameter specifies the address of an HTML document containing the description of a particular book. When the third argument is true, the request is asynchronous. The URL is passed to function getContent in response to the onmouseover event for each image. The server by calling XMLHttpRequest send method. The argument null indicates that this request is not submitting data in the body of the request.

Exception Handling:

An **exception** is an indication of a problem that occurs during a program's execution. The name "exception" implies that the problem occurs infrequently—if the "rule" is that a statement normally executes correctly, then the "exception to the rule" is that a problem occurs. Exception handling enables you to create applications that can resolve (or handle) exceptions—in some cases allowing a program to continue executing as if no problem had been encountered.

The exception parameter's name (exception in this example) enables the catch block to interact with a caught exception object (for example, to obtain the name of the exception or an exception-specific error message via the exception object's name and message properties). In this case, we simply display our own error message 'Request Failed' and terminate the getContent function. The request can fail because a user accesses the web page with an older browser or the content that is being requested is located on a different domain.

Using XML and the DOM:

When passing structured data between the server and the client, Ajax applications often use XML because it is easy to generate and parse. When the XMLHttpRequest object receives XML data, it parses and stores the data as an XMLDOMobject in the responseXML property.

Rich Functionality

The previous examples in this chapter requested data from static files on the server. The example in Fig. 13.9 is an address- book application that communicates with a server-side application. The application uses server-side processing to give the page the functionality and usability of a desktop application. We use JSON to encode server-side responses and to create objects on the fly.

The application also enables the user to add another entry to the address book by clicking the addEntry button. The application displays a form that enables live field validation. As the user fills out the form, the zip-code value is validated and used to generate the city and state. The telephone number is validated for correct format.

Interacting with a Web Service on the Server

When the page loads, the onload event calls the showAddressBook function to load the address book onto the page. Function shows the addressBook element and hides the addEntry element using the HTML DOM. Then it calls function callWebService to make an asynchronous request to the server. Function callWebService requires an array of parameter objects to be sent to the server. In this case, the function we are invoking on the server requires no arguments, It creates an empty array to be passed to callWebService. Our program uses an ASP.NET web service that we created for this example to do the server-side processing. The web service contains a collection of methods that can be called from a web application.

Function callWebService contains the code to call our web service, given a method name, an array of parameter bindings and the name of a callback function. The web-service application and the method that is being called are specified in the request URL. When sending the request using the GET method, the parameters are concatenated URL starting with a ? symbol and followed by a list of parameter=value bindings and iterate over the array of

parameter bindings that was passed as an argument, and add them to the request URL. In this first call, we do not pass any parameters because the web method that returns all the entries requires none. However, future web method calls will send multiple parameter bindings to the web service.

Parsing JSON Data

Each of our web service's methods in this example returns a JSON representation of an object or array of objects. For example, when the web application requests the list of names in the address book, the list is returned as a JSON array.

Creating XHTML Elements and Setting Event Handlers on the Fly

The parameters are generated dynamically and not evaluated until the getAddress function is called. This enables each function to receive arguments that are specific to the entry the user clicked. The names on the page by accessing the first (first name) and last (last name) fields of each element of the data array. Function getAddress is called when the user clicks an entry. This request must keep track of the entry where the address is to be displayed on the page.

Implementing Type-Ahead

The input element declared in the user to search the address book by last name. As soon as the user starts typing in the input box, the on key up event handler calls the search function passing the input element's value as an argument. The search function performs an asynchronous request to locate entries with last names that start with its argument value. When the response is received, the application displays the matching list of names. Each time the user changes the text in the input box, function search is called again to make another asynchronous request.
The search function first clears the address-book entries from the page. If the input argument is the empty string, displays the entire address book by calling function show Address Book. Otherwise send a request to the server to search the data. JSON string to represent the parameter object to be sent as an argument to the callWebServices function. The string to an object and calls the call Web Services function. When the server responds, callback function parse Data is invoked, which calls function display Names to display the results on the page.

Implementing a Form with Asynchronous Validation

When the Add an Entry addEntry function is called, which hides the addressBook element and shows the addEntry element that allows the user to add a person to the address book. The addEntry element (lines contains a set of entry fields, some of which have event handlers that enable validation that occurs asynchronously as the user continues to interact with the page. When a user enters a zip code, the validateZip function is called. This function calls an external web service to validate the zip code. If it is valid, that external web service returns the corresponding city and state. It builds a parameter object containing validateZip's parameter name and argument value in JSON format. callWebService function with the appropriate method, the parameter object created and showCityState as the callback function.

Zip-code validation can take a long time due to network delays. The showCityState function is called every time the request object's readyState property changes. Until the request completes, display "Checking zip code..." on the page. After the request completes converts the JSON response text to an object. The response object has four properties—Validity, ErrorText, City and State. If the request is valid, updates the zipValid variable that keeps track of zip-code validity and lines show the city and state that the server generated using the zip code.

When the Submit button is clicked, the saveForm function is called retrieve the data from the form. Once the server saves the data, it queries the database for an updated list of entries and returns them; then function parseData displays the entries on the page.

Dojo Toolkit:

Developing web applications in general, and Ajax applications in particular, involves a certain amount of painstaking and tedious work. Cross-browser compatibility,DOMmanipulation and event handling can get cumbersome, particularly as an application's size increases. Dojo is a free, open source JavaScript library that takes care of these issues. Dojo reduces asynchronous request handling to a single function call. Dojo also provides crossbrowser DOM functions that simplify partial page updates. It covers many more areas of web development, from simple event handling to fully functional rich GUI controls.

To install Dojo, download the Dojo version 0.4.3 from www.Dojotoolkit.org/ downloads to your hard drive. Extract the files from the archive file you downloaded to your web development directory or web server. Including the dojo.js script file in your web application will give you access to all the Dojo functions. To do this, place the following script in the head element of your XHTML document:

<script type = "text/javascript" src = "path**/Dojo.js">**

Where path is the relative or complete path to the Dojo toolkit's files. Quick installation instructions for Dojo are provided at Dojotoolkit.org/book/Dojo-book-0-9/part-1- life-Dojo/quick-installation.
The calendar application that uses Dojo to create the user interface, communicate with the server asynchronously, handle events and manipulate the DOM. The application contains a calendar control that shows the user six weeks of dates .Various arrow buttons allow the user to traverse the calendar. When the user selects a date, an asynchronous request obtains from the server a list of the scheduled events for that date. There is an **Edit** button next to each scheduled event. When the **Edit** button is clicked, the item is replaced by a text box with the item's content, a **Save** button and a **Cancel** button. When the user presses **Save**, an asynchronous request saves the new value to the server and displays it on the page. This feature, often referred to as **edit-in-place**, is common in Ajax applications. You can test-drive this application at test.deitel.com/examples/ajaxfp/ajax/fig13_11/calendar.html.

Loading Dojo Packages

The Dojo framework. It links the dojo.js script file to the page, giving the script access to all the functions in the Dojo toolkit. Dojo is organized in packages of related functionality. The dojo.require call, provided by the dojo.js script to include the packages we need. The dojo.io package functions communicate with the server, the dojo.event package simplifies event handling, the dojo.widget package provides rich GUI controls, and the dojo.dom package contains additional DOM functions that are portable across many different browsers. The application cannot use any of this functionality until all the packages have been loaded.

Using an Existing Dojo Widget

A Dojo widget is any predefined user interface element that is part of the Dojo toolkit. The calendar control on the page is the DatePicker widget. To incorporate an existing Dojo widget onto a page, you must set the DojoType attribute of any HTML element to the type of widget that you want it to be. Dojo widgets also have their own widgetID property. Tthe dojo.widget.byId method, rather than the DOM's document.getElementById method, to obtain the calendar widget element. The dojo.events.connect method links functions together. The ValueChanged event handler to the retrieve Items function. When the user picks a date, a special onValueChanged event that is part of the DatePicker widget calls retrieveItems, passing the selected date as an argument. The retrieve Items function builds the parameters for the request to the server, and calls the callWeb- Service function.

Asynchronous Requests in Dojo

The callWebService function sends the asynchronous request to the specified web-service method. Build the request URL using the same code. Dojo reduces the asynchronous request to a single call to the dojo.io.bind method , which works on all the popular browsers such as Firefox, Internet Explorer, Opera, Mozilla and Safari. The method takes an array of parameters, formatted as a JavaScript object. The url parameter specifies the destination of the request, the handler parameter specifies the callback function, and the mimetype parameter specifies the format of the response. The handler parameter can be replaced by the load and error parameters. The

function passed as load handles successful requests and the function passed as error handles unsuccessful requests. Response handling is done differently in Dojo. Rather than calling the callback function every time the request's ready State property changes, Dojo calls the function passed as the "handler" parameter when the request completes. In addition, in Dojo the script does not have access to the request object.

All the response data is sent directly to the callback function The function sent as the handler argument must have three parameters—type, data and event. In the first request, the function display Items is set as the callback function. The request is successful, and display an error message if it isn't. The place-holder element (item List), where the items will be displayed, and clear its content. The JSON response text to a JavaScript object, using the same sample code.

Partial Page Updates Using Dojo's Cross-BrowserDOMManipulation Capabilities

The Dojo toolkit (like most other Ajax libraries) provides functionality that enables you to manipulate the DOM in a cross- browser portable manner. Server-side returned any items, and display an appropriate message if it didn't. For each item object returned from the server, create a div element and set its id to the item's id in the database. A container element for the item's description. Dojo's dojo.dom.insertAtIndex method to insert the description element as the first element in the item's element. For each entry, the application creates an Edit button that enables the user to edit the event's content on the page. Dojo Button widget programmatically. Button Place Holder div element for the button and paste it.

widget by calling the dojo.widget.create Widget function. This function takes three parameters—the type of widget to be created, a list of additional widget parameters and the element which is to be converted to a Dojo widget.

Adding Edit-In-Place Functionality

Dojo Button widgets use their own buttonClick event instead of the DOM onclick event to store the event handler.The Dojo..event.connect method to connect the button Click event of the Dojo Button widget and the handle Edit event handler. When the user clicks the Edit button, the Event object gets passed to the event handler as an argument. The Event object's currentTarget property contains the element that initiated the event. uses the current Target property to obtain the id of the item. This id is the same as the item's id in the server database. The web service's getItem ById method, using the callWebService function to obtain the item that needs to be edited.

Once the server responds, function display and replaces the item on the screen with the user interface for editing the item's content. The code for this is similar to the code in the display Items function. Make sure the request was successful and parse the data from the server. Create the container elements into which we insert the new user-interface elements. The element that displays the item and change its id. Now the id of the user-interface element is the same as the id of the item that it's editing stored in the database. The text-box element that will be used to edit the item's description, paste it into the text box, and paste the resulting text box on the page. Tse the same syntax that was used to create the Edit button widget to create Save and Cancel button widgets. The resulting element, containing the text box and two buttons, on the page.

When the user edits the content and clicks the Cancel button, the handle Cancel function restores the item element to what it looked like before the button was clicked. Deletes the edit UI that was created earlier, using Dojo's remove Node function. The item with the original element that was used to display the item, and change its id back to the item's id on the server database. When the user clicks the Save button, the handle Save function sends the text entered by the user to the server.