MULTIMEDIA

MCA SEMESTER 5

(Mumbai University)

Prepared By

MissionMca.com

www.missionmca.com

(For private circulation only)

INDEX

Sr.No	Topic	Page No.
1.	Introduction to Multimedia	03
2.	Multimedia Elements	13
3.	Sound, Audio and video	34
4.	Multimedia Authoring Tools	51
5.	Designing and Producing	58
6.	Planning and costing	75
7.	Coding and Compression	84

Syllabus

1. Introduction to Multimedia :
Definition and scope of Multimedia, its components and applications, Use of Multimedia in CBT, Presentation & Conferencing, Hypermedia Multimedia in Public place, Multimedia on Web, Multimedia in business

2. Multimedia Elements :
Image file format, Bitmap image, Vector drawing Images, Principles of animations, Animation types & technique, Animation file and formats, Using text in multimedia, Designing with text, Text fonts, Menus and Navigation, Font editing drawing tools, file formats, Hypermedia and Hypertext

3. Sound, Audio and video :
Multimedia system sounds, MIDI audio, Audio file formats, MIDI v/s Digital Audio, Adding sound to your Multimedia Project, Analog display standards Digital display standards, Digital audio, Video recording and tap formats, Optimizing video files for CD-ROM

4. Multimedia Authoring Tools :
Making instance multimedia, Types of authoring tools, Time based authoring tools, card and page based authoring tools, Icon and object based authoring tools.
Story boarding, Media Design, Developing Multimedia Packages, Content analysis for different applications

5. Designing and Producing :
Designing, designing the structure of multimedia, Different types of multimedia structure, Hot spots hyperlink Buttons, Hot spots Web pages, designing the user interfaces: GUIs, Audio Interface, A multimedia design case history.
Delivering : Testing, Preparing of delivery, Delivering on CD-ROM, Compact Disk technology

6. Planning and costing :
The process of making multimedia, Idea analysis, Idea management software, Pre testing, task planning, building a team, prototype development: Alpha Development, Beta Development

7. Coding and Compression:
Introduction to coding and compression techniques, Entropy encoding run length, Huffman, JPEG compression process, MPEG audio and video compression, Various CD Formats, MPEG Standards

1. Introduction to Multimedia

What is Multimedia?

Multimedia is a computer-based interactive communications process that incorporates text, graphics, sound, animation, and video.

Multimedia can have a many definitions these include:

Multimedia means that computer information can be represented through audio, video, and animation in addition to traditional media (i.e., text, graphics drawings, images)
A good general definition is:

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

A ***Multimedia Application*** is an Application which uses a collection of multiple media sources e.g. text, graphics, images, sound/audio, animation and/or video.
Hypermedia can be considered as one of the multimedia applications.
.

Multimedia System

A *Multimedia System* is a system capable of processing multimedia data and applications.

A *Multimedia System* is characterised by the processing, storage, generation, manipulation and rendition of Multimedia information
A Multimedia system has four basic characteristics:

Multimedia systems must be ***computer controlled*.**
Multimedia systems are ***integrated*.**
The information they handle must be represented ***digitally*.**
The interface to the final presentation of media is usually ***interactive*.**

Multimedia is, as described as a woven combination of digitally manipulated text, photographs, graphic art, sound, animation, and video elements. When you allow an end user also known as the viewer of a multimedia project to control what and when the elements are delivered, it is called interactive multimedia.

When you provide a structure of linked elements through which the user can navigate, interactive multimedia becomes hypermedia.

A multimedia project need not be interactive to be called multimedia: users can sit back and watch it just as they do a movie or the television. In such cases a project is linear, or starting at the beginning and running through to the end. When users are given navigational control and can wander through the content at will, multimedia becomes nonlinear and user interactive, and is a powerful personal gateway to information.

Multimedia elements are typically sewn together into a project using authoring tools. These software tools are designed to manage individual multimedia elements and provide user interaction. Integrated multimedia is the "weaving" part of the multimedia definition, where source documents such as montages, graphics, video cuts, and sounds merge into a final presentation.

In addition to providing a method for users to interact with the project, most authoring tools also offer facilities for creating and editing text and images and controls for playing back separate audio and video files that have been created with editing tools designed for these media. The sum of what gets played back and how it is presented to the viewer on a monitor is the graphical user interface, or GUI (pronounced "gooey"). The GUI is more than just the actual graphics on the screen—it also often provides the rules or structure for the user's input. The hardware and software that govern the limits of what can happen here are the multimedia platform or environment.

Components of Multimedia

Now let us consider the Components (Hardware and Software) required for a multimedia system:

- Capture devices
  - -- Video Camera, Video Recorder, Audio Microphone, Keyboards.
- Storage Devices
  - -- Hard disks, CD-ROMs, Zip drives, DVD, etc
- Communication Networks
  - -- Ethernet, Token Ring, FDDI, ATM, Intranets, Internets.
- Computer Systems
  - -- Multimedia Desktop machines, Workstations, MPEG/VIDEO/DSP Hardware
- Display Devices
  - -- CD-quality speakers, HDTV,SVGA, Hi-Res monitors, Colour printers etc.

Applications of Multimedia

Examples of Multimedia Applications include:

- World Wide Web
- Hypermedia courseware
- Video conferencing
- Video-on-demand
- Interactive TV
- Groupware
- Home shopping
- Games
- Virtual reality
- Digital video editing and production systems
- Multimedia Database systems

Use of Multimedia in CBT

The number of people and corporations that are turning toward multimedia-based training as an alternative to traditional classroom training is growing rapidly. This method of training offers learners incredible flexibility in scheduling and attending training sessions.

Cost
Multimedia-based training can reduce the cost of learning drastically. With the ability to package training material onto portable storage devices or distribute over the Internet, the cost of teaching is reduced.

Flexibility
Without the need for live instructors, multimedia training allows for great flexibility. Students are able to access training with their computers at a time that is easiest for them.

Accessibility
Multimedia-based training allows access to training that some people may not have otherwise. The distance between the student and the school no longer matters.

Presentation & Conferencing

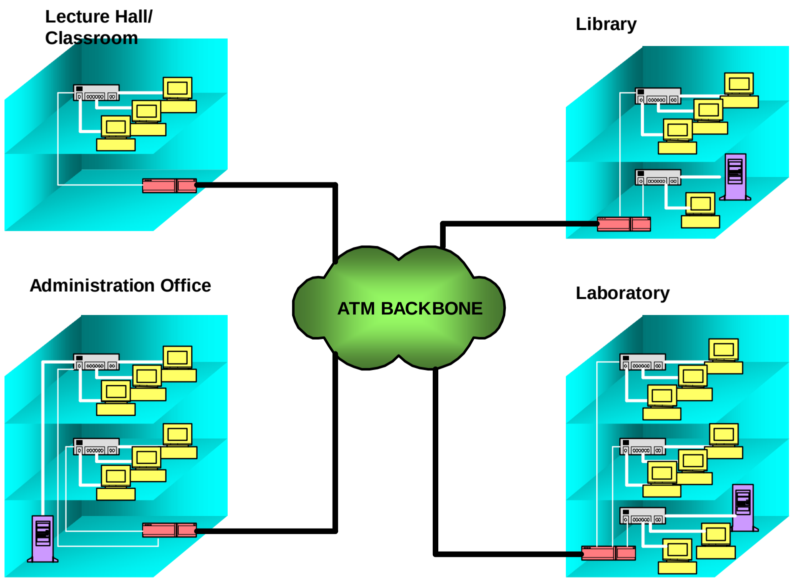
- Presentation is the practice of showing and explaining the content of a topic to an audience or learner. A presentation program, such as Microsoft PowerPoint, is often used to generate the presentation content.
- A presentation program is a computer software package used to display information, normally in the form of a slide show. It typically includes three major functions: an editor that allows text to be inserted and formatted, a method for

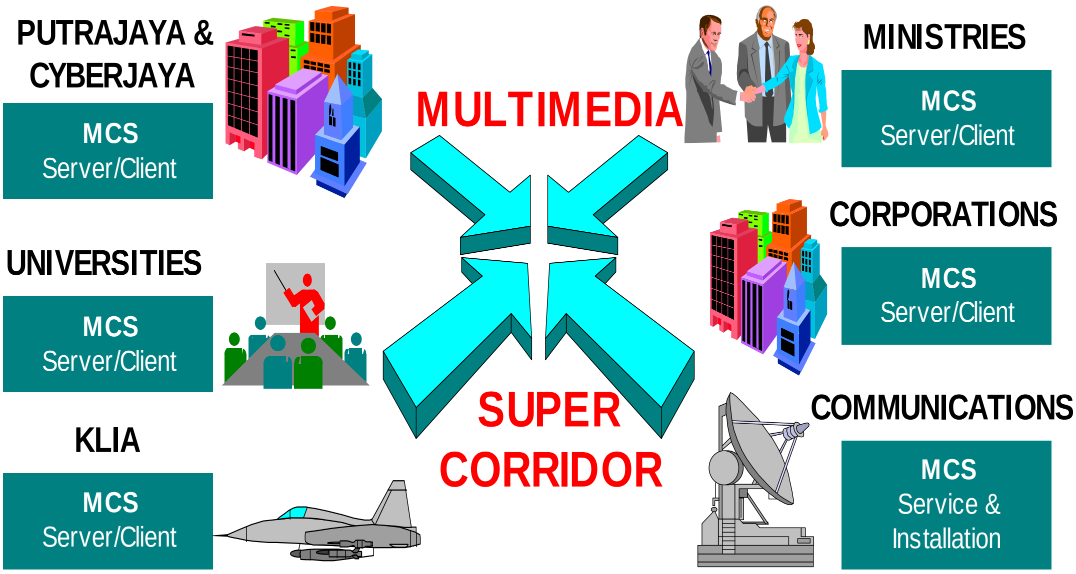
inserting and manipulating graphic images and a slide-show system to display the content.

- Presentation packages are designed to support the same media types, but without support for testing, scorekeeping, or tracking of end users.
- Some packages are intended for creation of simple linear presentations, which do not offer multiple pathways through the information.
- Presentation packages usually have a single price and simply offer different modes for creation and playback.
- Presentations are typically linear sequences of information, so the interfaces for users have often been organized along the time dimension.
- Current presentation packages allow users to construct interactive i.e. nonlinear, presentations that more closely resemble those produced by conventional authoring systems.

Multimedia **Conferencing** System, MCS is the first of its kind to offer virtually unlimited multipoint-to-multipoint interactivity for notebook, desktop, boardroom and wireless in Boardroom or Enterprise.
MCS is also already making a name for itself in the global marketplace with its other attractive features such as no call costs, low additional infrastructure needs, open-system hardware requirements and wireless capability.

To enhance its unique multipoint-to-multipoint feature, MCS has add-on capabilities to cover special needs like multimedia collaboration (Microsoft Word, Excel and PowerPoint file-sharing), multiple-party file transfer, global roaming, document confere ... e easily integrat





MCS is well suited for intra-corporate meetings, distance learning, telemedicine and long-distance/ international desktop conferencing.
At the same time, its relatively low system procurement and usage costs make MCS an excellent value investment for a far wider range of clientele.

Multimedia on Web

Today's Web presents a diversified multimedia experience. In fact, the Web has become a broadcast medium, offering live TV and radio, pre-recorded video, photos, images, and animations. Expect to encounter multimedia just about anywhere on the Web. This tutorial presents a brief overview.

Plugins, media players, and multimedia types

PLUGINS and **MEDIA PLAYERS** are software programs that allow you to experience multimedia on the Web. These terms are sometimes used interchangeably. File formats requiring this software are known as **MIME TYPES**. MIME stands for Multimedia Internet Mail Extension, and was originally developed to help e-mail software handle a variety of binary (non-textual) file attachments such as photos. The use of MIME has expanded to the Web. For example, the basic MIME type handled by Web browsers is text/html associated with the file extention .html. MIME types area also used to process multimedia on the Web. A few examples:

- Jpeg photo: image/jpeg
- MPEG video: video/mpeg
- Quicktime movie: video/quicktime
- MP3 audio: audio/x-mpeg-3
- Flash presentation: application/x-shockwave-flash

Nowadays, many personal computers come pre-loaded with plugins and media players. This is an acknowledgement of the importance of the Web multimedia experience. If your computer doesn't have a particular piece of software, it can be easily obtained from the website of the company that created it. Downloading is easy and instructions are usually provided.

PLUGINS are software programs that work with your Web browser to display multimedia. When your browser encounters a multimedia file, it hands off the data to the plugin to play or display the file. Working in conjunction with plugins, browsers can offer a seamless multimedia experience. The plugins needed to experience Web multimedia are available for free.

A common plugin used on the Web is the Adobe Reader. This software allows you to view documents created in Adobe's Portable Document Format (PDF). These documents are the MIME type "application/pdf" and are associated with the file extension .pdf. A PDF is a type of image file. When the Adobe Reader has been downloaded to your computer, the software will open and display the file when you click on its link on a Web page. Try viewing **this PDF document** from the U. S. Census Bureau. Sometimes a Web page will display the official PDF icon to alert you that the file is in PDF format.

MEDIA PLAYERS are software programs that can play audio and video files, both on and off the Web. The concept of **STREAMING MEDIA** is important to understanding how media can be delivered on the Web. With streaming technology, audio or video files are played as they are downloading, or streaming, into your computer. Sometimes a small wait, called buffering, is necessary before the file begins to play. Extensive pre-recorded files such as interviews, lectures, televised video clips, podcasts, and music work very well with these players. They can also be used for real-time radio and TV, including Web-only TV. Popular media players include the Windows Media Player, RealPlayer, QuickTime Player, and Flash Player.

Audio

Audio files, including music, are an important part of the Web experience. Listening to music on the Web is a popular pastime. Audio files of many types are supported by the Web with the appropriate players. The MP3 file format probably the most popular option for audio files.

MP3 files are also the source of **PODCASTS**. These are audio files distributed through RSS feeds, though the term is sometimes also used to describe video programming (or **VODCAST**). You can subscribe to a podcast's RSS feed, and listen to the podcast series, with a special type of player called a **PODCATCHER**. A podcatcher can be either available on the Web or downloaded to your computer like any other plugin. **iTunes** can serve as a podcatcher. Keep in mind that you can often listen to a podcast on the originating site. For an example, visit**NYTimes.com Podcasts**.

Thousands of radio stations broadcast live on the Web. Just use a search engine to locate a station's website, and follow the links to the live broadcast. Visit **this page** for an example of one radio station's live broadcasts.

Video

Streaming video is the backbone of live and pre-recorded broadcasting on the Web. **YouTube** is one of the most popular sites on the Web for pre-recorded video. Real-time professional or personal broadcasts are also very popular.

The Web is a medium for exchanging information among professionals. A live professional broadcast from a conference, company, or institution is sometimes referred to as a **WEBCAST**. A variation on this is a **WEBINAR**, a seminar broadcast on the Web.

To watch video discussions by experts in their fields, take a look at:

- **Academic Earth**, a collection of free video lectures by top scholars
- **BigThink**, where experts discuss current events
- **Bloggingheads.tv**, where academics, journalists, and others have two-way conversations, or diavlogs, on substantive topics
- **Hulu**, a site offering TV broadcasts and movies
- **iTunes U**, which offers free lectures from a handful of universities
- **WebMedia: Special Events** at Princeton University, offering archived speeches and conferences

LIVE CAMS and **LIVE TV** are a part of the real-time video experience available on the Web. Live cams are video cameras that send their data in real time to a Web server. These cams may appear in all kinds of locations, both serious and whimsical: an office, on top of a building, a scenic locale, a special event, a fish tank, and so on. Live cams are stationary and only broadcast what is in their line of sight. Moving video takes live broadcasting to the next level: TV on the Web. Some people wear portable cameras and allow the public to observe their lives - an intense form of reality TV. **Justin.tv** was a pioneer in this type of live broadcasting. Other people broadcast their involvement in specific topics, such as cooking or technology. Check out **Blip.tv**and **Ustream** for examples.

Live TV broadcasts abound on the Web. As with radio stations mentioned above, use a search engine to locate a station's website and follow the links to the live broadcast. There are also plenty of pre-recorded network TV shows available on the Web. Check out **Hulu** for an example of a site that hosts this type of content.

Photos

Photos may seem like old hat, but the Web has found interesting ways of presenting them. Most major search engines have an option for searching for photos. This is usually combined with a search for non-photographic images along with photos, and is therefore called an image search. **Google Image Search** is a good example. There are also search engines dedicated to image searching, for example **Picsearch**.

The social Web has come up with innovative ideas for photo sharing. Check out **Flickr** and**Pinterest** for examples of different ways in which you can use the social Web to display and share your photos - and also to view, share and comment on the photos of others.

Embedded media

As you browse the Web, you can experience multimedia on the sites of the people who sponsor or create the broadcasts. There are also aggregator sites you can visit, including Flickr and YouTube.

It's also possible to embed multimedia on your own Web pages. The capacity for unlimited distribution is a major reason why multimedia on the Web has become so popular. Also, it's easy to do. In most cases, embedding a media file is just a matter of copying code and pasting it onto your Web page. The two examples below took only a few minutes to complete. Both require the Flash player (MIME type of application/x-shockwave-flash).

Applications of Multimedia

Multimedia in Business

Business applications for multimedia include presentations, training, marketing, advertising, product demos, simulations, databases, catalogs, instant messaging, and networked communications. Voice mail and video conferencing are provided on many local and wide area networks (LANs and WANs) using distributed networks and Internet protocols. After a morning of mind-numbing overhead presentations delivered from the podium of a national sales conference, a multimedia presentation can make an audience come alive. Most presentation software packages let you make pretty text and add audio and video clips to the usual slide show of graphics and text material.

Multimedia is enjoying widespread use in training programs. Flight attendants learn to manage international terrorism and security through simulation. Drug enforcement agencies of the UN are trained using interactive videos and photographs to recognize likely hiding places on airplanes and ships. Medical doctors and veterinarians can practice surgery methods via simulation prior to actual surgery. Mechanics learn to repair engines. Salespeople learn about product lines and leave behind software to train their customers. Fighter pilots practice full-terrain sorties before spooling up for the real thing. Increasingly easy-to-use authoring programs and media production tools even let workers on assembly lines create their own training programs for use by their peers.

Multimedia around the office has also become more commonplace. Image capture hardware is used for building employee ID and badging databases, scanning medical insurance cards, for video annotation, and for real-time teleconferencing. Presentation documents attached to e-mail and video conferencing are widely available. Laptop computers and high resolution projectors are commonplace for multimedia presentations on the road. Mobile phones and personal digital assistants (PDAs) utilizing Bluetooth and Wi-Fi communications technology make communication and the pursuit of business more efficient.

As companies and businesses catch on to the power of multimedia, the cost of installing multimedia capability decreases, meaning that more applications can be developed both in-house and by third parties, which allow businesses to run more smoothly and effectively. These advances are changing the very way business is transacted by affirming that the use of multimedia offers a significant contribution to the bottom line while also advertising the public image of the business as an investor in technology.

Multimedia in Public Places

In hotels, train stations, shopping malls, museums, libraries, and grocery stores, multimedia is already available at stand-alone terminals or kiosks, providing information and help for customers. Multimedia is piped to wireless devices such as cell phones and PDAs. Such installations reduce demand on traditional information booths and personnel, add value, and are available around the clock, even in the middle of the night, when live help is off duty. The way we live is changing as multimedia penetrates our day-to-day experience and our culture. Imagine a friend’s bout of maudlin drunk dialing (DD) on a new iPhone, with the camera accidentally enabled.

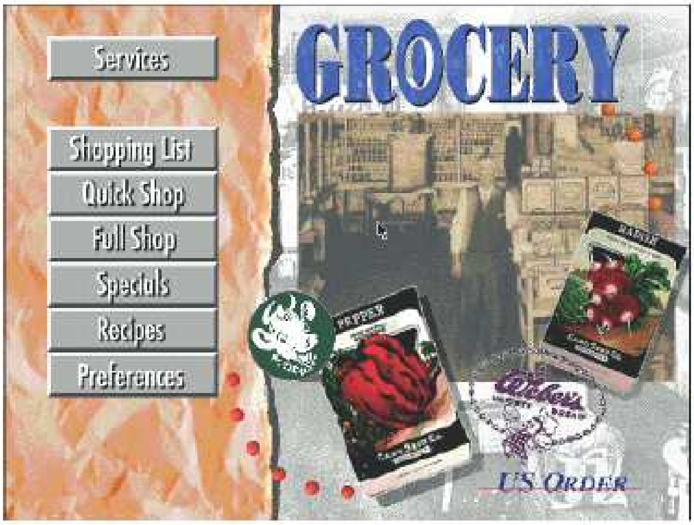


Figure 1-3 Kiosks in public places can make everyday life simpler.

Figure 1-3 shows a menu screen from a supermarket kiosk that provides services ranging from meal planning to coupons. Hotel kiosks list nearby restaurants, maps of the city, airline schedules, and provide guest services such as automated checkout. Printers are often attached so that users can walk away with a printed copy of the information. Museum kiosks are not only used to guide patrons through the exhibits, but when installed at each exhibit, provide great added depth, allowing visitors to browse through richly detailed information specific to that display.

The power of multimedia has been part of the human experience for many thousands of years, and the mystical chants of monks, cantors, and shamans accompanied by potent visual cues, raised icons, and persuasive text has long been known to produce effective responses in public places.

2. Multimedia Elements

Image File Formats

Most applications on any operating system can manage JPEG, GIF, PNG, and TIFF image formats. An older format used on the Macintosh, PICT, is a complicated but versatile format developed by Apple where both bitmaps and vector-drawn objects can live side by side. The device-independent bitmap (DIB), also known as a BMP, is a common Windows palette–based image file format similar to PNG.

PCX files were originally developed for use in Z-Soft MS-DOS paint packages; these files can be opened and saved by almost all MS-DOS paint software and desktop publishing software. TIFF, or Tagged Interchange File Format, was designed to be a universal bitmapped image format and is also used extensively in desktop publishing packages. Often, applications use a proprietary file format to store their images. Adobe creates a PSD file for Photoshop and an AI file for Illustrator; Corel creates a CDR file.

DXF was developed by AutoDesk as an ASCII-based drawing interchange file for AutoCAD, but the format is used today by many computer-aided design applications.

IGS (or IGES, for Initial Graphics Exchange Standard) was developed by an industry committee as a broader standard for transferring CAD drawings. These formats are also used in 3-D rendering and animation programs.

JPEG, PNG, and GIF images are the most common bitmap formats used on the Web and may be considered cross-platform, as all browsers will display them. Adobe's popular PDF (Portable Document File) file manages both bitmaps and drawn art (as well as text and other multimedia content), and is commonly used to deliver a "finished product" that contains multiple assets.

Bitmap image

A bit is the simplest element in the digital world, an electronic digit that is either on or off, black or white, or true (1) or false (0). This is referred to as binary, since only two states (on or off) are available. A map is a two dimensional matrix of these bits. A bitmap, then, is a simple matrix of the tiny dots that form an image and are displayed on a computer screen or printed.

A one-dimensional matrix (1-bit depth) is used to display monochrome images a bitmap where each bit is most commonly set to black or white. Depending upon your software, any two colors that represent the on and off (1 or 0) states may be used. More information is required to describe shades of gray or the more than 16 million colors that each picture element might have in a color image, as illustrated in Figure 3-1.

These picture elements (known as pels or, more commonly, pixels) can be either on or off, as in the 1-bit bitmap, or, by using more bits to describe them, can represent varying shades of color (4 bits for 16 colors; 8 bits for 256 colors; 15 bits for 32,768 colors; 16

bits for 65,536 colors; 24 bits for 16,772,216 colors). Thus, with 2 bits, for example, the available zeros and ones can be combined in only four possible ways and can, then, describe only four possible colors:

Bit Depth	Number of Colors Possible	Available Binary Combinations for Describing a Color
1-bit	2	0, 1
2-bit	4	00, 01, 10, 11
4-bit	16	0000, 0001, 0011, 0111, 1111, 0010, 0100, 1000, 0110, 1100, 1010, 0101, 1110, 1101, 1001, 1011

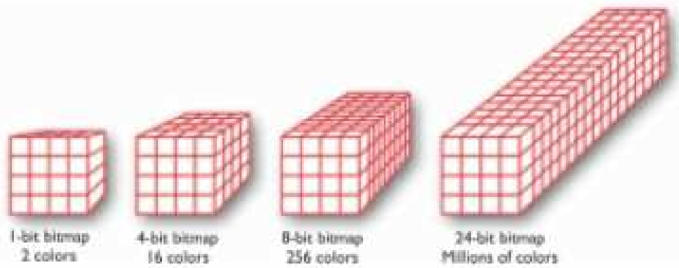


Figure 3-1 A bitmap is a data matrix that describes the characteristics of all the pixels making up an image. Here, each cube represents the data required to display a 4 × 4–pixel image (the face of the cube) at various color depths (with each cube extending behind the face indicating the number of bits—zeros or ones—used to represent the color for that pixel).

Together, the state of all the pixels on a computer screen make up the image seen by the viewer, whether in combinations of black and white or colored pixels in a line of text, a photograph-like picture, or a simple background pattern. Figure 3-2 demonstrates various color depths and compression formats.

Image 1 is 24 bits deep (millions of colors);
Image 2 is dithered to 8 bits using an adaptive palette (the best 256 colors to represent the image);

Image 3 is also dithered to 8 bits, but uses the Macintosh system palette (an optimized standard mix of 256 colors).

Image 4 is dithered to 4 bits (any 16 colors);
Image 5 is dithered to 8-bit gray-scale (256 shades of gray);
Image 6 is dithered to 4-bit gray-scale (16 shades of gray); and
Image 7 is dithered to 1 bit (two colors—in this case, black and white).

Bitmap Sources

Where do bitmaps come from? How are they made? You can do the following:

- Make a bitmap from scratch with a paint or drawing program.
- Grab a bitmap from an active computer screen with a screen capture program, and then paste it into a paint program or your application.
- Capture a bitmap from a photo or other artwork using a scanner to digitize the image.
- Once made, a bitmap can be copied, altered, e-mailed, and otherwise used in many creative ways.

If you do not want to make your own, you can get bitmaps from suppliers of clip art, and from photograph suppliers who have already digitized the images for you. Libraries of clip art are available on CD-ROMs and downloadable through online services. Many graphics applications are shipped with clip art and useful graphics. A clip art collection may contain a random assortment of images, or it may contain a series of graphics, photographs, sound, and video related to a single topic. Some 3-D modeling programs incorporate libraries of pre-made 3-D models into the application, allowing you to drag and drop common objects into a scene.

Vector drawing Images

Most multimedia authoring systems provide for use of vector-drawn objects such as lines, rectangles, ovals, polygons, complex drawings created from those objects, and text.

- Computer-aided design (CAD) programs have traditionally used vector-drawn object systems for creating the highly complex and geometric renderings needed by architects and engineers.
- Graphic artists designing for print media use vector-drawn objects because the same mathematics that put a rectangle on your screen can also place that rectangle (or the fancy curves of a good line-art illustration) on paper without jaggies. This requires the higher resolution of the printer, using a page description format such as Portable Document Format (PDF).
- Programs for 3-D animation also use vector-drawn graphics. For example, the various changes of position, rotation, and shading of light required to spin an extruded corporate logo must be calculated mathematically. (Animation is discussed in Chapter 5.)

How Vector Drawing Works

A vector is a line that is described by the location of its two endpoints. Vector drawing uses Cartesian coordinates where a pair of numbers describes a point in two-dimensional space as the intersection of horizontal and vertical lines (the x and y axes). The numbers are always listed in the order x,y. In three-dimensional space, a third dimension—depth— is described by a z axis (x,y,z). This coordinate system is named for the French philosopher and mathematician, René Descartes. So a line might be simply

<line x1="0" y1="0" x2="200" y2="100">

where x1 and y1 define the starting point (in the upper-left corner of the viewing box) and x2 and y2 define the end point.

A simple rectangle is computed from starting point and size: your software will draw a rectangle (rect) starting at the upper-left corner of your viewing area (0,0) and going 200 pixels horizontally to the right and 100 pixels downward to mark the opposite corner. Add color information Like

```
<rect x="0" y="0" width="200" height="100" fill="#FFFFFF" stroke="#FF0000"/>
```

and your software will draw the rectangle with a red boundary line and fill it with the color white. You can, of course, add other parameters to describe a fill pattern or the width of the boundary line. Circles are defined by a location and a radius:

```
<circle cx="50" cy="50" r="10" fill="none" stroke="#000000" />
```

Type the following code into a text editor and save it as plain text with a .svg extension. This is a Scalable Vector Graphics file. Open it in an HTML5-capable browser (File:Open File…) and you will see:

```
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
width="200"
height="200"
viewBox="-100 -100 300 300">
<rect x="0" y="0" fill="yellow" stroke="red" width="200" height="100"/>
<text transform="matrix(1 0 0 1 60 60)" font-family="'TimesNewRomanPS-BoldMT'" font-
size="36">SVG</text>
</svg>
```

Because these SVG files can be saved in a small amount of memory and because they are scalable without distortion (try changing the width and height of the view box in the preceding code), SVG (Tiny) is supported by browsers on most mobile phones and PDAs. The SVG specification also includes timebased changes or animations that can be embedded within the image code Figure 3-8 shows Adobe Illustrator saving a file in SVG format. Vector drawing tools use Bézier curves or paths to mathematically represent a curve. In practical terms, editing software shows you points on the path, each point having a "handle." Changing the location of the handle changes the shape of the curve. Mastering Bézier curves is an important skill: these curves not only create graphic shapes but represent motion paths when creating animations.

Vector-Drawn Objects vs. Bitmaps

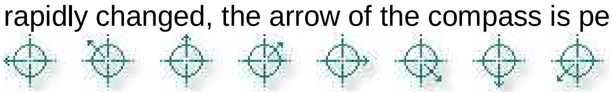
Vector-drawn objects are described and drawn to the computer screen using a fraction of the memory space required to describe and store the same object in bitmap form. The file containing the vector-drawn colored rectangle described in the preceding section is less than 698 bytes of alphanumeric data (even less—468 bytes—when the description is tokenized or compressed as .svgz). On the other hand, the same rectangle saved as a .gif image with a 64-color palette takes 1,100 bytes. Because of this file size advantage, web pages that use vector graphics as SVG files or in plug-ins such as Flash download faster and, when used for animation, draw faster than pages displaying bitmaps. It is only when you draw many hundreds of objects on your screen that you may experience a slowdown while you wait for the screen to be refreshed—the size, location, and other properties for each of the objects must be computed.

Thus, a single image made up of 500 individual line and rectangle objects, for example, may take longer for the computer to process and place on the screen than an image consisting of just a few drawn circle objects. A vector-drawn object is created “on the fly,” that is, the computer draws the image from the instructions it has been given, rather than displaying a pre created image.

This means that vector objects are easily scalable without loss of resolution or image quality. A large drawn image can be shrunk to the size of a postage stamp, and while it may not look good on a computer monitor at 72 dpi, it may look great when printed at 300 dpi to a color printer. Resizing a bitmapped image requires either duplicating pixels (creating a blocky, jagged look called pixelation) or throwing pixels away (eliminating details). Because vector images are drawn from instructions on the fly, a rescaled image retains the quality of the original.

Principles of Animation

Animation is possible because of a biological phenomenon known as persistence of vision and a psychological phenomenon called phi. An object seen by the human eye remains chemically mapped on the eye’s retina for a brief time after viewing. Combined with the human mind’s need to conceptually complete a perceived action, this makes it possible for a series of images that are changed very slightly and very rapidly, one after the other, to seemingly blend together into a visual illusion of movement. The illustration shows a few cels, or frames, of a rotating logo. When the images are progressively and rapidly changed, the arrow of the compass is perceived to be spinning.



Digital television video builds 24, 30, or 60 entire frames or pictures every second, depending upon settings; the speed with which each frame is replaced by the next one makes the images appear to blend smoothly into movement. Movies on film are typically shot at a shutter rate of 24 frames per second, but using projection tricks (the projector’s shutter flashes light through each image twice), the flicker rate is increased to 48 times per second, and the human eye thus sees a motion picture.

On some film projectors, each frame is shown three times before the pull-down claw moves to the next frame, for a total of 72 flickers per second, which helps to eliminate the flicker effect: the more interruptions per second, the more continuous the beam of light appears. Quickly changing the viewed image is the principle of an animatic, a flip-book, or a zoetrope.

To make an object travel across the screen while it changes its shape, just change the shape and also move, or translate, it a few pixels for each frame. Then, when you play the frames back at a faster speed, the changes blend together and you have motion and animation. It's the same magic as when the hand is quicker than the eye, and you don't see the pea moving in the blur of the gypsy's cups.

Animation Types

Using appropriate software and techniques, you can animate visual images in many ways. The simplest animations occur in two-dimensional (2-D) space; more complicated animations occur in an intermediate "2½-D" space (where shadowing, highlights, and forced perspective provide an illusion of depth, the third dimension); and the most realistic animations occur in three-dimensional (3-D) space.

In 2-D space, the visual changes that bring an image alive occur on the flat Cartesian x and y axes of the screen. A blinking word, a color-cycling logo (where the colors of an image are rapidly altered according to a formula), a cel animation (described more fully later on in this chapter), or a button or tab that changes state on mouse rollover to let a user know it is active are all examples of 2-D animations. These are simple and static, not changing their position on the screen. Path animation in 2-D space increases the complexity of an animation and provides motion, changing the location of an image along a predetermined path (position) during a specified amount of time (speed). Authoring and presentation software such as Flash or PowerPoint provide user-friendly tools to compute position changes and redraw an image in a new location, allowing you to generate a bouncing ball or slide a corporate mascot onto the screen. Combining changes in an image with changes in its position allows you to "walk" your corporate mascot onto the stage. Changing its size from small to large as it walks onstage will give you a 3-D perception of distance.

In 2½-D animation, an illusion of depth (the z axis) is added to an image through shadowing and highlighting, but the image itself still rests on the flat x and y axes in two dimensions. Embossing, shadowing, beveling, and highlighting provide a sense of depth by raising an image or cutting it into a background. Zaxwerks' 3D Invigorator (www.zaxwerks.com), for example, provides 3-D effects for text and images and, while calling itself "3D," works within the 2-D space of image editors and drawing programs such as Adobe Illustrator, Photoshop, Fireworks, and After Effects.

In 3-D animation, software creates a virtual realm in three dimensions, and changes (motion) are calculated along all three axes (x, y, and z), allowing an image or object that itself is created with a front, back, sides, top, and bottom to move toward or away from the viewer, or, in this virtual space of light sources and points of view, allowing the viewer to wander around and get a look at all the object's parts from all angles.

Such animations are typically rendered frame by frame by high-end 3-D animation programs such as NewTek's Lightwave or AutoDesk's Maya. Today, computers have taken the handwork out of the animation and rendering process, and commercial films such as *Shrek, Coraline, Toy Story,* and *Avatar* have utilized the power of computers. (See Chapter 3 for an account of the historic "computer wall" of 117 Sun SPARCstations used to render the animated feature *Toy Story.*)

Animation Techniques

When you create an animation, organize its execution into a series of logical steps. First, gather up in your mind all the activities you wish to provide in the animation. If it is complicated, you may wish to create a written script with a list of activities and required objects and then create a storyboard to visualize the animation.

Choose the animation tool best suited for the job, and then build and tweak your sequences. This may include creating objects, planning their movements, texturing their surfaces, adding lights, experimenting with lighting effects, and positioning the camera or point of view. Allow plenty of time for this phase when you are experimenting and testing. Finally, post-process your animation, doing any special renderings and adding sound effects.

Cel Animation

The animation techniques made famous by Disney use a series of progressively different graphics or cels on each frame of movie film (which plays at 24 frames per second). A minute of animation may thus require as many as 1,440 separate frames, and each frame may be composed of many layers of cels. The term cel derives from the clear celluloid sheets that were used for drawing each frame, which have been replaced today by layers of digital imagery. Cels of famous animated cartoons have become sought-after, suitable-for-framing collector's items.

Cel animation artwork begins with keyframes (the first and last frame of an action). For example, when an animated figure of a woman walks across the screen, she balances the weight of her entire body on one foot and then the other in a series of falls and recoveries, with the opposite foot and leg catching up to support the body. Thus the first keyframe to portray a single step might be the woman pitching her body weight forward off the left foot and leg, while her center of gravity shifts forward; the feet are close together, and she appears to be falling.

The last keyframe might be the right foot and leg catching the body's fall, with the center of gravity now centered between the outstretched stride and the left and right feet positioned far apart. The series of frames in between the keyframes are drawn in a process called tweening. Tweening is an action that requires calculating the number of frames between keyframes and the path the action takes, and then actually sketching with pencil the series of progressively different outlines. As tweening progresses, the action sequence is checked by flipping through the frames. The penciled frames are assembled and then actually filmed as a pencil test to check smoothness, continuity, and timing.

When the pencil frames are satisfactory, they are permanently inked, photocopied onto cels, and given to artists who use acrylic colors to paint the details for each cel. Women were often preferred for this painstaking inking and painting work as they were deemed patient, neat, and had great eyes for detail. In the hands of a master, cel paint applied to the back of acetate can be simply flat and perfectly even, or it can produce beautiful and subtle effects, with feathered edges or smudges. The cels for each frame of our example of a walking woman—which may consist of a text title, a background, foreground, characters (with perhaps separate cels for a left arm, a right arm, legs, shoes, a body, and facial features)—are carefully registered and stacked.

It is this composite that Becomes the final photographed single frame in an animated movie. To replicate natural motion, traditional cel animators often utilized “motion capture” by photographing a woman walking, a horse trotting, or a cat jumping to help visualize timings and movements. Today, animators use reflective sensors applied to a person, animal, or other object whose motion is to be captured. Cameras and computers convert the precise locations of the sensors into x,y,z coordinates and the data is rendered into 3-D surfaces moving over time.

Computer Animation

Computer animation programs typically employ the same logic and procedural concepts as cel animation and use the vocabulary of classic cel animation terms such as layer, keyframe, and tweening.

The primary difference among animation software programs is in how much must be drawn by the animator and how much is automatically generated by the software (see Figure 5-2). In path-based 2-D and 2½-D animation, an animator simply creates an object (or imports an object as clip art) and describes a path for the object to follow. The computer software then takes over, actually creating the animation on the fly as the program is being viewed by your user. In cel-based 2-D animation, each frame of an animation is provided by the animator, and the frames are then composited (usually with some tweening help available from the software) into a single file of images to be played in sequence.

ULead’s GIF Animator (www.ulead.com/ga) and Alchemy’s GIF Construction Set Pro (www .mindworkshop.com) simply string together your collection of frames. For 3-D animation, most of your effort may be spent in creating the models of individual objects and designing the characteristics of their shapes and surfaces. It is the software that then computes the movement of the objects within the 3-D space and renders each frame, in the end stitching them together in a digital output file or container such as an AVI or QuickTime movie. On the computer, paint is most often filled or drawn with tools using features such as gradients and anti-aliasing.

The word inks, in computer animation terminology, usually means special methods for computing color values, providing edge detection, and layering so that images can blend or otherwise mix their colors to produce special transparencies, inversions, and effects. You can usually set your own frame rates on the computer. 2-D celbased animated GIFs, for example, allow you to specify how long each frame is to be displayed and how many times the animation should loop before stopping. 3-D animations output as digital video files can be set to run at 15 or 24 or 30 frames per second.

However, the rate at which changes are computed and screens are actually refreshed will depend on the speed and power of your user's display platform and hardware, especially for animations such as path animations that are being generated by the computer on the fly. Although your animations will probably never push the limits of a monitor's scan rate (about 60 to 70 frames per second), animation does put raw computing horsepower to task. If you cannot compute all your changes and display them as a new frame on your monitor within, say, 1/15th of a second, then the animation may appear jerky and slow.

Luckily, when the files include audio, the software maintains the continuity of the audio at all cost, preferring to drop visual 3-D animations are typically delivered as "pre-rendered" digital video clips. Software such as Flash or PowerPoint, however, render animations as they are being viewed, so the animation can be programmed to be interactive: touch or click on the jumping cat and it turns toward you snarling; touch the walking woman and…

Kinematics Kinematics is the study of the movement and motion of structures that have joints, such as a walking man. Animating a walking step is tricky: you need to calculate the position, rotation, velocity, and acceleration of all the joints and articulated parts involved—knees bend, hips flex, shoulders swing, and the head bobs. Smith Micro's Poser models (male, female, infant, teenage, and superhero) in many poses, such as "walking" or "thinking." As you can see in Figure 5-3, you can pose figures in 3-D and then scale and manipulate individual body parts. Surface textures can then be applied to create muscle-bound hulks or smooth chrome androids. Inverse kinematics, available in high-end 3-D programs such as Lightwave and Maya, is the process by which you link objects such as hands to arms and define their relationships and limits (for example, elbows cannot bend backward). Once those relationships and parameters have been set, you can then drag these parts around and let the computer calculate the result

Morphing

Morphing is a popular (if not overused) effect in which one image transforms into another. Morphing applications and other modeling tools that offer this effect can transition not only between still images but often between moving images as well. Some products that offer morphing features are Black Belt's Easy Morph and WinImages (www.blackbeltsystems. com) and Human Software's Squizz (www.humansoftware. com). Figure 5-4 illustrates part of a morph in which 16 kindergarten children are dissolved one into the other in a continuous, compelling motion video.

The morphed images were built at a rate of eight frames per second, with each transition taking a total of four seconds (32 separate images for each transition), and the number of key points was held to a minimum to shorten rendering time. Setting key points is crucial for a smooth transition between two images. The point you set in the start image will move to the corresponding point in the end image—this is important for things like eyes and noses, which you want to end up in about the same place (even if they look different) after the transition. The more key points, the smoother the morph. In Figure 5-4, the red dot on each child's temple is a matching key point.

Animation File Formats

Some file formats are designed specifically to contain animations, so they can be ported among applications and platforms with the proper translators. Those formats include Director (.dir and .dcr), AnimatorPro (.fli and .flc), 3D Studio Max (.max), GIF89a (.gif), and Flash (.fla and .swf).
Because file size is a critical factor when downloading animations to play on web pages, file compression is an essential part of preparing animation files for the Web. A Director's native movie file (.dir), for example, must be preprocessed and compressed into a proprietary Shockwave animation file (.dcr) for the Web. Compression for Director movies is as much as 75 percent or more with this tool, turning 100K files into 25K files and significantly speeding up download/display times on the Internet. Flash, widely used for web-based animation, makes extensive use of vector graphics (see Chapter 3) to keep the post-compression file size at absolute minimums. As with Director, its native .fla files must be converted to Shockwave Flash files (.swf) in order to play on the Web.

In some cases, especially with 3-D animations, the individual rendered frames of an animation are put together into one of the standard digital video file containers, such as the Windows Audio Video Interleaved format (.avi), QuickTime (.qt, .mov), or Motion Picture Experts Group video (.mpeg or .mpg). These can be played using the media players shipped with computer operating systems.

New with HTML5 is animation built within a .svg (scalable vector graphics) file, where graphic elements can be programmed to change over time (www.w3.org/TR/SVG11/animate.html). In the following simple code, a patch of red expands within a rectangle, filling it in three seconds. Type this code into a text processor and save it as plain text with a .svg extension. Open the file with "File Open…" from a HTML5-compliant web browser to see it work. Change some parameters (duration, colors, location) and reload or refresh the file to see the effects of your changes.

```
<svg width="8cm" height="3cm" viewBox="0 0 800 300"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<rect x="1" y="1" width="800" height="300" fill="none" stroke="rgb(255,0,255)" stroke-width="4" />
<rect id="RectElement" x="300" y="100" width="300" height="100" fill="rgb(255,0,0)" >
<animate attributeName="x" attributeType="XML" begin="0s" dur="3s" fill="freeze"
from="300" to="0" />
<animate attributeName="y" attributeType="XML" begin="0s" dur="3s" fill="freeze"
from="100" to="0" />
<animate attributeName="width" attributeType="XML" begin="0s" dur="3s" fill="freeze"
from="300" to="800" />
<animate attributeName="height" attributeType="XML" begin="0s" dur="3s" fill="freeze"
from="100" to="300" />
</rect>
</svg>
```

Fonts and Faces

A typeface is a family of graphic characters that usually includes many type sizes and styles. A font is a collection of characters of a single size and style belonging to a particular typeface family. Typical font styles are boldface and italic. Your computer software may add other style attributes, such as underlining and outlining of characters. Type sizes are usually expressed in points; one point is 0.0138 inch, or about 1/72 of an inch. The font's size is the distance from the top of the capital letters to the bottom of the descenders in letters such as *g* and *y.* Helvetica, Times, and Courier are typefaces; Times 12-point italic is a font.

In the computer world, the term font is commonly used when typeface or face would be more correct. A font's size does not exactly describe the height or width of its characters. This is because the x-height (the height of the lowercase letter *x*) of two fonts may vary, while the height of the capital letters of those fonts may be the same (see Figure 2-1). Computer fonts automatically add space below the descender (and sometimes above) to provide appropriate line spacing, or leading (pronounced "ledding," named for the thin strips of lead inserted between the lines by traditional typesetters).

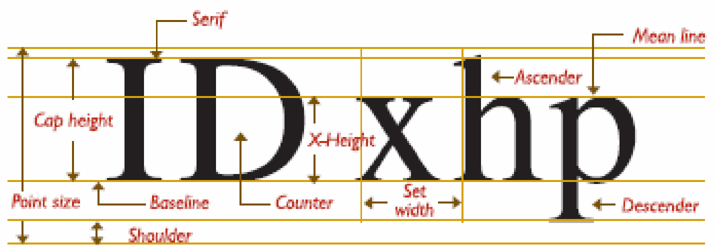
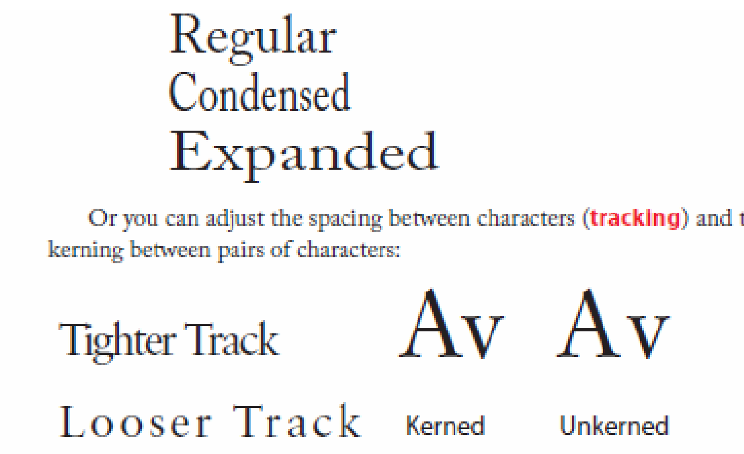


Figure 2-1 The measurement of type

and the PC. Typically you will find this fine-tuning adjustment in the Text menu of image-editing programs or the Paragraph menu of word processing programs, though this is not an official standard. No matter where your application has placed the controls for leading, you will need to experiment with them to achieve the best result for your font. With a font editing program like Fontographer from Fontlab, Ltd. at www.fontlab.com (you'll see an example of it later in the chapter), adjustments can also be made along the horizontal axis of text. In this program the character metrics of each character and the kerning of character pairs can be altered.

Character metrics are the general measurements applied to individual characters; kerning is the spacing between character pairs. When working with PostScript, TrueType, and Master fonts—but not bitmapped fonts— (see "Computers and Text" later in this chapter), the metrics of a font can be altered to create interesting effects. For example, you can adjust the body width of each character from regular to condensed to expanded, as displayed in this example using the Sabon font:



When it converts the letter *A* from a mathematical representation to a recognizable symbol displayed on the screen or in printed output (a process called rasterizing), the computer must know how to represent the letter using tiny square pixels (picture elements), or dots. It does this according to the hardware available and your specification, from a choice of available typefaces and fonts. Search for "free fonts." High-resolution monitors and printers can make more attractive-looking and varied characters because there are more fine little squares or dots per inch (dpi). And today's broad selection of software fonts makes it easier to find the right typeface and font for your needs. The same letter can look very different when you use different fonts and faces:

Cases

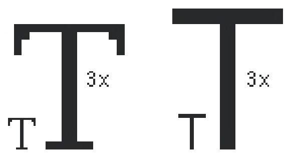
In centuries when type was set by hand, the type for a single font was always stored in two trays, or *cases*; the upper tray held capital letters, and the lower tray held the small letters. Today, a capital letter is called uppercase, and a small letter is called lowercase. In some situations, such as for passwords, a computer is case sensitive, meaning that the text's upper- and lowercase letters must match exactly to be recognized. But nowadays, in most situations requiring keyboard input, all computers recognize both the upper- and lowercase forms of a character to be the same. In that manner, the computer is said to be case insensitive.

Company and product names such as WordPerfect, OmniPage, PhotoDisc, FileMaker, and WebStar have become popular. Placing an uppercase letter in the middle of a word, called an intercap, is a trend that emerged from the computer programming community, where coders discovered they could better recognize the words they used for variables and commands when the words were lowercase but intercapped.

Serif vs. Sans Serif

Typefaces can be described in many ways, just as a home advertised by a realtor, a wine described by a food critic, or a political candidate's platform nine,\ masculine, delicate, formal, capricious, witty, comic, happy, technical, newsy—you name it. But one approach for categorizing typefaces is universally understood, and it has less to do with the reader's response to the type than it does with the type's mechanical and historical properties.

This approach uses the terms serif and sans serif. Serif versus sans serif is the simplest way to categorize a typeface; the type either has a serif or it doesn't (*sans* is French for "without"). The serif is the little decoration at the end of a letter stroke. Times, New Century Schoolbook, Bookman, and Palatino are examples of serif fonts. Helvetica, Verdana, Arial, Optima, and Avant Garde are sans serif. Notice the difference between serif (on the left) and sans serif:



On the printed page, serif fonts are traditionally used for body text because the serifs are said to help guide the reader's eye along the line of text. Sans serif fonts, on the other hand, are used for headlines and bold statements. But the computer world of standard, 72 dpi monitor resolution is not the same as the print world, and it can be argued that sans serif fonts are far more legible and attractive when used in the small sizes of a text field on a screen.

Indeed, careful selection of a sans serif font designed to be legible in the small sizes (such as Tahoma or Verdana) makes more sense when you are presenting a substantial amount of text on the screen.

The Times font at 9-point size may look too busy and actually be difficult and tiring to read. And a large, bold serif font for a title or headline can deliver a message of elegance and character in your graphic layout. Use what is right for your delivery system, which may not necessarily be the same as what is right when you're printing the material to paper. This is because when you're printing out what you create on a computer monitor, WYSIWYG (What You See Is What You Get) is more of a goal than an absolute fact.

Using Text in Multimedia

Imagine designing a project that used no text at all. Its content could not be at all complex, and you would need to use many pictures and symbols to train your audience how to navigate through the project. Certainly voice and sound could guide the audience, but users would quickly tire of this because greater effort is required to pay attention to spoken words than to browse text with the eye. A single item of menu text accompanied by a single action (a mouse click, keystroke, or finger pressed to the monitor) requires little training and is clean and immediate. Use text for titles and headlines (what it's all about), for menus (where to go), for navigation (how to get there), and for content (what you see when you get there).

Designing with Text

Computer screens provide a very small workspace for developing complex ideas. At some time or another, you will need to deliver high-impact or concise text messages on the computer screen in as condensed a form as possible. From a design perspective, your choice of font size and the number of headlines you place on a particular screen must be related both to the complexity of your message and to its venue. If your messages are part of an interactive project or web site where you know the user is seeking information, you can pack a great deal of text information onto the screen before it becomes overwhelmingly busy. Seekers want dense material, and while they travel along your navigational pathways, they will scroll through relevant text and study the details. Here is where you must strike a balance, however.

Too little text on a screen requires annoying page turns and unnecessary mouse clicks and waits; too much text can make the screen seem overcrowded and unpleasant. On the other hand, if you are creating presentation slides for public speaking support, the text will be keyed to a live presentation where the text accents the main message. In this case, use bulleted points in large fonts and few words with lots of white space. Let the audience focus on the speaker at the podium, rather than spend its time reading fine points and subpoints projected on a screen.

Choosing Text Fonts

Picking the fonts to use in your multimedia presentation may be somewhat difficult from a design standpoint. Here again, you must be a poet, an advertising psychologist, and also a graphic designer. Try to intuit the potential reaction of the user to what is on the screen. Here are a few design suggestions that may help:

- For small type, use the most legible font available. Decorative fonts that cannot be read are useless, as shown at right.
- Use as few different faces as possible in the same work, but vary the weight and size of your typeface using italic and bold styles where they look good. Using too many fonts on the same page is called ransom-note typography. Visit http://lifehacker.com/software/writing/ faster-ransom-notes-for-busy-kidnappers-248692.php to make your own ransom notes.
- In text blocks, adjust the leading for the most pleasing line spacing. Lines too tightly packed are difficult to read.

- Vary the size of a font in proportion to the importance of the message you are delivering.
- In large-size headlines, adjust the spacing between letters (kerning) so that the spacing feels right. Big gaps between large letters can turn your title into a toothless waif. You may need to kern by hand, using a bitmapped version of your text.
- To make your type stand out or be more legible, explore the effects of different colors and of placing the text on various backgrounds. Try reverse type for a stark, white-on-black message.
- Use anti-aliased text where you want a gentle and blended look for titles and headlines. This can give a more professional appearance. Anti-aliasing blends the colors along the edges of the letters (called dithering) to create a soft transition between the letter and its background.
- try drop caps (like the T to the left) and initial caps to accent your words. Most word processors and text editors will let you create drop caps and small caps in your text. Adobe and others make initial caps (such as the one shown to the right from Adobe, called Gothic). The letters are actually carefully drawn artwork and are available in special libraries as encapsulated PostScript files (EPSF).
- Coding an initial cap for a web page is simple. Use CSS attributes:
  p:first-letter { font-size: 200%; }
  p:first-line { line-height: 100%; }

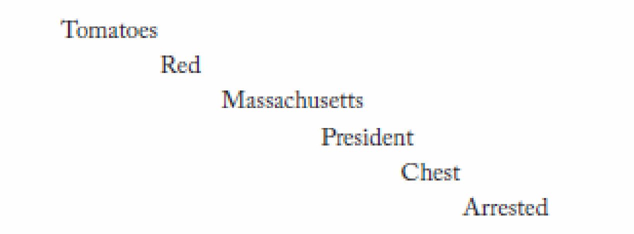
- If you are using centered type in a text block, keep the number of lines and their width to a minimum.
- For attention-grabbing results with single words or short phrases, try graphically altering and distorting your text and delivering the result as an image. Wrap your word onto a sphere, bend it into a wave, or splash it with rainbow colors.
- Experiment with drop shadows. Place a copy of the word on top of the original, and offset the original up and over a few pixels. Then color the original gray (or any other color). The word may become more legible and provide much greater impact. With web sites, shadowed text and graphics on a plain white background add depth to a page. Surround headlines with plenty of white space. White space is a designer’s term for roomy blank areas, while programmers call the invisible character made by a space (ASCII 32) or a tab (ASCII 9) white space. Web designers use a nonbreaking space entity ( ) to force spaces into lines of text in HTML documents.

- Pick the fonts that seem right to you for getting your message across, then double-check your choice against other opinions. Learn to accept criticism.
- Use meaningful words or phrases for links and menu items.
- Text links on web pages can accent your message: they normally stand out by color and underlining. Use link colors consistently throughout a site, and avoid iridescent green on red or purple on puce.
- Bold or emphasize text to highlight ideas or concepts, but do not make text look like a link or a button when it is not.
- On a web page, put vital text elements and menus in the top 320 pixels. Studies of surfer habits have discovered that only 10 to 15 percent of surfers *ever* scroll *any* page.

Menus for Navigation

An interactive multimedia project or web site typically consists of a body of information, or content, through which a user navigates by pressing a key, clicking a mouse, or pressing a touch screen. The simplest menus consist of text lists of topics. Users choose a topic, click it, and go there. As multimedia and graphical user interfaces become pervasive in the computer community, certain intuitive actions are being widely learned. For example, if there are three words on a computer screen, the typical response from the user, without prompting, is to click one of these words to evoke activity. Sometimes menu items are surrounded by boxes or made to look like push buttons. Or, to conserve space, text such as Throw Tomatoes, Play Video, and Press to Quit is often shortened to Tomatoes, Video, and Quit. Regardless, the user deduces the function. Text is helpful to users to provide persistent cues about their location within the body of content. When users must click up and down through many layers of menus to reach their goal, they may not get lost, but they may feel transported to the winding and narrow streets of a medieval city where only the locals know the way.

This is especially true if the user moves slowly from screen to screen en route to that goal. If Throw Tomatoes leads to Red or Green, then to California or Massachusetts, then to President or Vice President, then to Forehead or Chest, then to Arrested or Got Away, and so on, the user can end up tangled in the branches of a navigation tree without cues or a map. However, if an interactive textual or symbolic list of the branches taken (all the way from the beginning) is continuously displayed, the user can at any time skip intervening steps in a nonlinear manner or easily return to one of the previous locations in the list.



The more locations included in the menu list, the more options available for navigation. On the Web, designers typically place on every page at least a Main Menu of links that offers the user a handhold and mechanism for returning to the beginning. Often they will also place a list, such as

along the tops of storefronts to let shoppers know where they are currently located within the store. Inventive interface developers first referred to this array of menu items as “breadcrumbs,” for they represent a map of the virtual forest and often the “trail” users have taken, like the edible markers so intelligently placed by Hänsel und Gretel along the way to the witch’s house in the Brother Grimms’ famous fairytale

Font Editing and Design Tools

Special font editing tools can be used to make your own type, so you can communicate an idea or graphic feeling exactly. With these tools, professional typographers create distinct text and display faces. Graphic designers, publishers, and ad agencies can design instant variations of existing typefaces. Typeface designs fall into the category of industrial design and have been determined by the courts in some cases to be protected by patent.

For example, design patents have been issued for Bigelow & Holmes' Lucida, ITC Stone, and Adobe's Minion. Occasionally in your projects you may require special characters. With the tools described in the paragraphs that follow, you can easily substitute characters of your own design for any unused characters in the extended character set. You can even include several custom versions of your client's company logo or other

Fontlab

Fontlab, Ltd., located at www.fontlab.com, specializes in font editors for both Macintosh and Windows platforms. You can use this software to develop PostScript, TrueType, and OpenType fonts for Macintosh, Windows, and Sun workstations. Designers can also modify existing typefaces, incorporate PostScript artwork, automatically trace scanned images, and create designs from scratch. A sample of the Fontographer screen is shown in Figure 2-7.

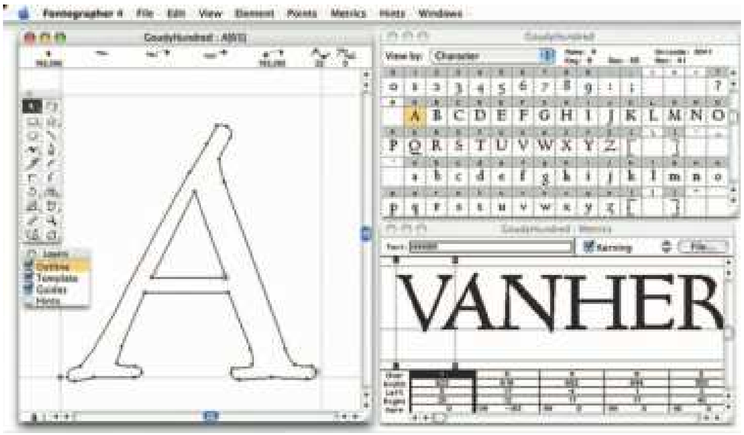


Figure 2-7 Fontographer is a powerful font editor for Macintosh and Windows.

Fontographer's features include a freehand drawing tool to create professional and precise inline and outline drawings of calligraphic and script characters, using either the mouse or alternative input methods (such as a pressure-sensitive pen system). Fontographer allows the creation of multiple font designs from two existing typefaces, and you can design lighter or heavier fonts by modifying the weight of an entire typeface

Hypermedia and Hypertext

Multimedia—the combination of text, graphic, and audio elements into a single collection or presentation—becomes interactive multimedia when you give the user some control over what information is viewed and when it is viewed. Interactive multimedia becomes hypermedia when its designer provides a structure of linked elements through which a user can navigate and interact. When a hypermedia project includes large amounts of text or symbolic content, this content can be indexed and its elements then linked together to afford rapid electronic retrieval of the associated information. When words are keyed or indexed to other words, you have a hypertext system; the "text" part of this term represents the project's content and meaning, rather than the graphical presentation of the text. Hypertext is what the World Wide Web is all about. When text is stored in a computer instead of on printed pages, the computer's powerful processing capabilities can be applied to make the text more accessible and meaningful. The text can then be called hypertext; because the words, sections, and thoughts are linked, the user can navigate through text in a nonlinear way, quickly and intuitively. Using hypertext systems, you can electronically search through all the text of a computer-resident book, locate references to a certain word, and then immediately view the page where the word was found. Or you can create complicated Boolean searches (using terms such as AND, OR, NOT, and BOTH) to locate the occurrences of several related words, such as "Elwood," "Gloria," "mortgage," and "happiness," in a paragraph or on a page. Whole documents can be linked to other documents.

A word can be made hot, as can a button, thus leading the user from one reference to another. Click on the word "Elwood," and you may find yourself reading a biography or resume; click on "mortgage," and a calculator pops up. Some authoring systems incorporate a hypertext facility that allows you to identify words in a text field using a bold or colored style, then link them to other words, pages, or activities, such as playing a sound or video clip related to that hot word.

You cannot do this kind of nonlinear and associative navigation in a sequentially organized book. But on a CD-ROM, where you might have more than 100,000 pages of text to investigate, search, and browse, hypertext is invaluable. Because hypertext is the organized cross-linking of words not only to other words but also to associated images, video clips, sounds, and other exhibits, hypertext often becomes simply an additional feature within an overall multimedia design.

The term "hyper" (from the Greek word "over" has come to imply that user interaction is a critical part of the design, whether for text browsing or for the multimedia project as a whole. When interaction and cross-linking is then added to multimedia, and the navigation system is nonlinear, multimedia becomes hypermedia. In 1945, Vannevar Bush wrote a seminal eight-page article, "As We May Think," for the *Atlantic Monthly* (www.theatlantic.com/unbound/ flashbks/computer/bushf.htm).

This short treatise, in which he discusses the need for new methodologies for accessing information, has become the historic cornerstone of hypertext experimentation. Doug Englebart (inventor of the mouse) and Ted Nelson (who coined the term "hypertext" in 1965) have actively championed the research and innovations required of computer technology for implementing useful hypertext systems, and they have worked to combat the historic inertia of linear thought. Nelson linear and that computer-based hypertext systems will fundamentally alter the way humans approach literature and the expression of ideas during the coming decades.

The argument against this theory of associative thought is that people are, indeed, more comfortable with linear thinking and are easily overwhelmed by too much freedom, becoming quickly lost in the chaos of nonlinear gigabytes. As a practical reminder, it is important always to provide location markers, either text-and-symbol menus or illustrative maps, for users who travel the threads of nonlinear systems.

The Power of Hypertext

In a fully indexed hypertext system, all words can be found immediately. Suppose you search a large database for “boats,” and you come up with a whopping 1,623 references, or hits—among them, Noah’s Ark (open boat in water), television situation comedies *(The Love Boat),* political criticisms of cabinet members who challenged the status quo (rocked the boat), cabinet members who were stupid (missed the boat), and Christmas dinner trimmings (Grandmother’s gravy boat). So you narrow your searchand look for “boats” and “water” when both words are mentioned on the same page; this time you get 286 hits. “Boats,” “water,” and “storms” gets you 37; “boats,” “water,” “storms,” and “San Francisco,” a single hit. With over a thousand hits, you are lost. With one hit, you have something! But you still may not find what you are looking for, as you can see in this fictional example:

The *storm* had come and gone quickly across the Colorado plains, but *water* was still puddled at the foot of the house-high bank of mud that had slid into town when the dam burst. In front of the general store, which had remained standing, four strong men carefully lifted a tiny *boat* onto the large dray wagon borrowed from the woodcutters. On a layer of blankets in the bilge of the *boat,* the undertaker had carefully laid out the remains of both the mayor and his paramour. The mayor had not drowned in the flood, but died of a heart attack in the midst of the panic. Children covered the *boat* with freshly cut pine boughs while horses were quickly harnessed to the wagon, and a strange procession began to move slowly down *San Francisco* Street toward the new cemetery. …

The power of such search-and-retrieval systems provided by a computer for large volumes of data is immense, but clearly this power must be channeled in meaningful ways. Links among words or clusters of information need to be designed so that they make sense. Judgments must be made about relationships and the way information content is organized and made available to users. The lenses through which vast amounts of data are viewed must necessarily be ground and shaped by those who design the access system. The issue of who designs the lenses and how the designers maintain impartial focus is troubling to many scientists, archivists, and students of cognitive thinking. The scientists would remain “hermeneutically” neutral, they would balance freedom against authority and warn against the epistemological unknowns of this new intellectual technology.

They are aware of the forces that allow advertising and marketing craftspeople to intuitively twist meanings and spin events to their own purposes, with actions that can affect the knowledge and views of many millions of people and thus history itself. But these forces remain poorly understood, are not easily controlled by authority, and will express themselves with immeasurably far-reaching, long-term impact on the shape of human culture.

The multimedia designer controls the filtering mechanisms and places the lenses within the multimedia project. A manufacturer, for instance, that presents its products using interactive multimedia can bring abundant information and selling power within reach of the user, including background information, collateral marketing material, pricing statistics, and technical data. The project design will be, of course, biased—to sell more of the manufacturer’s products and generate more profit; but this bias is assumed and understood in these circumstances. When the assumptions and understandings of inherent bias in any information base break down, when fiction or incomplete data is presented as full fact, these are the times when the powerful forces of multimedia and hypermedia can have their greatest deleterious effect.

Using Hypertext

Special programs for information management and hypertext have been designed to present electronic text, images, and other elements in a database fashion. Commercial systems have been used for large and complicated mixtures of text and images—for example, a detailed repair manual for a Boeing 747 aircraft, a parts catalog for Pratt & Whitney jet turbine engines, an instant reference to hazardous chemicals, and electronic reference libraries used in legal and library environments. Such searchable database engines are widely used on the Web, where software robots visit millions of web pages and index entire web sites.

Hypertext databases rely upon proprietary indexing systems that carefully scan the entire body of text and create very fast cross-referencing indexes that point to the location of specific words, documents, and images. Indeed, a hypertext index by itself can be as large as 50 percent to 100 percent the size of the original document. Indexes are essential for speedy performance. Google's
search engine produces about 1,220,000,000 hits in less than a quarter of a second! Commercial hypertext systems were developed historically to retrofit gigantic bodies of information. Licenses for use and distribution of these commercial systems are expensive, and the hypertext-based projects typically require the large mass-storage capability of one or many CD-ROMs and/or dedicated gigabyte hard disks. Simpler but effective hypertext indexing tools are available for both Macintosh and Windows, and they offer fairly elaborate features designed to work in concert with many multimedia authoring systems. Server-based hypertext and database engines designed for the Web are now widely available and competitively priced.

Searching for Words

Although the designer of a hypermedia database makes assumptions, he or she also presents users with tools and a meaningful interface to exercise the assumptions. Employing this interface, users can tailor word searches to find very specific combinations. Following are typical methods for word searching in hypermedia systems:

- Categories Selecting or limiting the documents, pages, or fields of text within which to search for a word or words.
- Word relationships Searching for words according to their general proximity and order. For example, you might search for "party" and "beer" only when they occur on the same page or in the same paragraph.
- Adjacency Searching for words occurring next to one another, usually in phrases and proper names. For instance, find "widow" only when "black" is the preceding adjacent word.
- Alternates Applying an OR criterion to search for two or more words, such as "bacon" or "eggs."

- Association Applying an AND criterion to search for two or more words, such as "skiff," "tender," "dinghy," and "rowboat."
- Negation Applying a NOT criterion to search exclusively for references to a word that are not associated with the word. For example, find all occurrences of "paste" when "library" is not present in the same sentence.
- Truncation Searching for a word with any of its possible suffixes. For example, to find all occurrences of "girl" and "girls," you may need to specify something like girl#. Multiple character suffixes can be \managed with another specifier, so geo* might yield "geo," "geology," and "geometry," as well as "George."
- Intermediate words Searching for words that occur between what might normally be adjacent words, such as a middle name or initial in a proper name.
- Frequency Searching for words based on how often they appear: the more times a term is mentioned in a document, the more relevant the document is to this term.

Hypermedia Structures

Two buzzwords used often in hypertext systems are link and node. Links are connections between the conceptual elements, that is, the nodes, which may consist of text, graphics, sounds, or related information in the knowledge base. Links connect Caesar Augustus with Rome, for example, and grapes with wine, and love with hate. The art of hypermedia design lies in the visualization of these nodes and their links so that they make sense, not nonsense, and can form the backbone of a knowledge access system. Along with the use of HTML for the World Wide Web, the term anchor is used for the reference from one document to another document, image, sound, or file on the Web. Links are the navigation pathways and menus; nodes are accessible topics, documents, messages, and content elements. A link anchor is where you come from; a link end is the destination node linked to the anchor. Some hypertext systems provide unidirectional navigation and offer no return pathway; others are bidirectional. The simplest way to navigate hypermedia structures is via buttons that let you access linked information (text, graphics, and sounds) that is contained at the nodes. When you've finished examining the information, you return to your starting location

Hypertext Tools

Two functions are common to most hypermedia text management systems, and they are often provided as separate applications: building (or authoring) and reading. The builder creates the links, identifies nodes, and generates the all-important index of words. The index methodology and the search algorithms used to find and group words according to user search criteria are typically proprietary, and they represent an area where computers are carefully optimized for performance finding search words among a list of many tens of thousands of words requires speed-demon programming.

Hypertext systems are currently used for electronic publishing and reference works, technical documentation, educational courseware, interactive kiosks, electronic catalogs, interactive fiction, and text and image databases. Today these tools are used extensively with information organized in a linear fashion; it still may be many years before the majority of multimedia project users become comfortable with fully nonlinear hypertext and hypermedia systems. they do, the methodology of human thought and conceptual management—indeed, the way we think—will be forever changed.

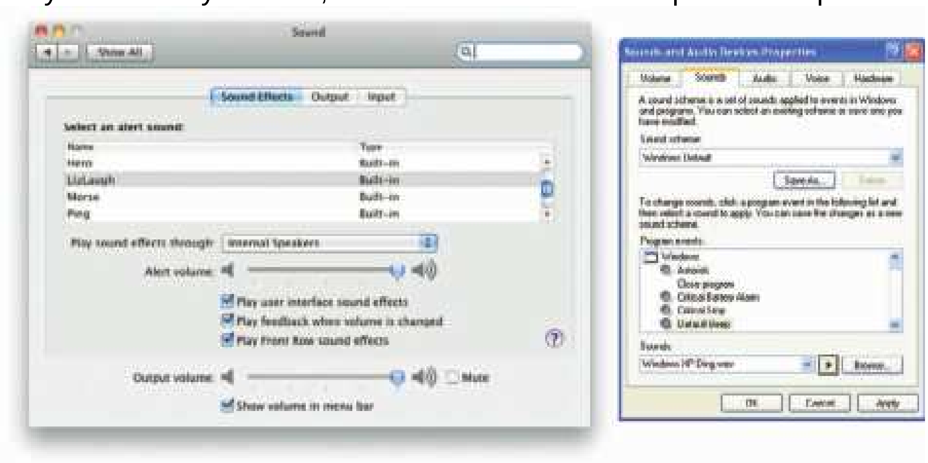
3. Sound, Audio and video

Multimedia System Sounds

You can use sound right off the bat on your computer because beeps and warning sounds are available as soon as you install the operating system. Open the Sound Control Panel to listen to your system sounds, change them, or make a new, custom sound (see Figure 4-7). In Windows, system sounds are WAV files, and they reside in the Windows\ Media subdirectory.

System event sounds include start.wav, chimes.wav, chord.wav, ding.wav, logoff.wav, notify.wav, recycle.wav, tada. wav, and the Microsoft sound.wav that typically plays when Windows starts up. As you can see in Figure 4-7, you can assign these sounds to system events such as Windows startup, warnings from other applications, or clicks outside of an open dialog box (which causes the default beep in Windows). And you can create schemes of sounds and select a particular scheme according to your mood.

You can also add your own sound files and install them so they play when system events occur: place the WAV sound files into your ~\Windows\Media directory and use the Sound Control Panel to select them. In OS X on a Macintosh, you can only change your system alert sound. Put your custom sound file (in AIF format) into ~/System/Library/Sounds, then select it in the Sound preference pane.



MIDI Audio

MIDI (Musical Instrument Digital Interface) is a communications standard developed in the early 1980s for electronic musical instruments and computers. It allows music and sound synthesizers from different manufacturers to communicate with each other by sending messages along cables connected to the devices. MIDI provides a protocol for passing detailed descriptions of a musical score, such as the notes, the sequences of notes, and the instrument that will play these notes. But MIDI data is not digitized sound; it is a shorthand representation of music stored in numeric form. Digital audio is a recording, MIDI is a score—the first depends on the capabilities of your sound system, the other on the quality of your musical instruments *and* the capabilities of your sound system. A MIDI file is a list of time-stamped commands that are recordings of musical actions (the pressing down of a piano key or a sustain pedal, for example, or the movement of a control wheel or slider). When sent to a MIDI playback device, this results in sound. A concise MIDI message can cause a complex sound or sequence of sounds to play on an instrument or synthesizer; so MIDI files tend to be significantly smaller (per second of sound delivered to the user) than equivalent digitized waveform files. Composing your own original score can be one of the most creative and rewarding aspects of building a multimedia project, and MIDI is the quickest, easiest, and most flexible tool for this task. Yet creating an original MIDI score is hard work. Knowing something about music, being able to play a keyboard, and having a lot of good ideas are just the prerequisites to building a good score; beyond that, it takes time and musical skill to work with MIDI.

Happily, you can always hire someone to do the job for you. In addition to the talented MIDI composers who charge substantial rates for their services, many young composers are also available who want to get into multimedia. With a little research, you can often find a MIDI musician to work for limited compensation. Remember, however, that you often get what you pay for. The process of creating MIDI music is quite different from digitizing existing recorded audio. If you think of digitized audio as analogous to a bitmapped graphic image (both use sampling of the original analog medium to create a digital copy), then MIDI is analogous to structured or vector graphics (both involve instructions provided to software to be able to re-create the original on the fly). For digitized audio you simply play the audio through a computer or device that can digitally record the sound.

To make MIDI scores, however, you will need notation software (see Figure 4-5), sequencer software (see Figure 4-6), and a sound synthesizer (typically built into the software of multimedia players in most computers and many handheld devices). A MIDI keyboard is also useful for simplifying the creation of musical scores. Rather than recording the sound of a note, MIDI software creates data about each note as it is played on a MIDI keyboard (or another MIDI device)—which note it is, how much pressure was used on the keyboard to play the note, how long it was sustained, and how long it takes for the note to decay or fade away, for example. This information, when played back through a MIDI device, allows the note to be reproduced exactly. Because the quality of the playback depends upon the end user's MIDI device rather than the recording, MIDI is device dependent.

The sequencer software quantizes your score to adjust for timing inconsistencies (a great feature for those who can't keep the beat), and it may also print a neatly penned copy of your score to paper. An advantage of structured data such as MIDI is the ease with which you can edit the data. Let's say you have a piece of music being played on a honky-tonk piano, but your client decides he wants the sound of a soprano saxophone instead. If you had the music in digitized audio, you would have to re-record and redigitize the music.

When it is in MIDI data, however, there is a value that designates the instrument to be used for playing back the music. To change instruments, you just change that value. Instruments that you can synthesize are identified by a General MIDI numbering system that ranges from 0 to 127 (see Table 4-3). Until this system came along, there was always a risk that a MIDI file originally composed with, say, piano, electric guitar, and bass, might be played back with piccolo, tambourine, and glockenspiel if the ID numbers were not precisely mapped to match the original hardware setup. This was usually the case when you played a MIDI file on a MIDI configuration different from the one that recorded the file.

ID	Sound	ID	Sound
0	Acoustic grand piano	16	Hammond organ
1	Bright acoustic piano	17	Percussive organ
2	Electric grand piano	18	Rock organ
3	Honky-tonk piano	19	Church organ
4	Rhodes piano	20	Reed organ
5	Chorused piano	21	Accordion
6	Harpsichord	22	Harmonica
7	Clarinet	23	Tango accordion
8	Celesta	24	Acoustic guitar (nylon)
9	Glockenspiel	25	Acoustic guitar (steel)
10	Music box	26	Electric guitar (jazz)
11	Vibraphone	27	Electric guitar (clean)
12	Marimba	28	Electric guitar (muted)
13	Xylophone	29	Overdriven guitar
14	Tubular bells	30	Distortion guitar
15	Dulcimer	31	Guitar harmonics

Table 4-3 General MIDI Instrument Sounds

Advantages of MIDI

MIDI is:

- **Compact** - Hours of music can fit on a single 3 1/2" floppy disk, thousands of songs on a CD
- **Editable** - It is an easily edited / manipulated form of data.
- **Efficient** - Just about any computer can handle it
- **Powerful** - A whole orchestra is at your command
- **Quality** – Its of high quality and sound better than digital audio files.
- **Size** – As they are small, MIDI files embedded in web pages, load and play more quickly than their digital equivalents.
- **Versatile** - A click of a button is all it takes to change key, tempo, instrument, etc.
- **Intuitive** - a MIDI file is just an electronic version of a player piano roll for many instruments
- **Portable Industry Standard** - Any MIDI instrument can talk to any other

Disadvantages of MIDI

- As MIDI data does not represent sound but musical instruments, you can be certain that playback will be accurate only if the Midi playback device is identical to the device used for production.
- MIDI cannot easily be used to play back spoken dialog, although expensive and technically tricky digital samplers are available.

Audio File Formats

When you create multimedia, it is likely that you will deal with file formats and translators for text, sounds, images, animations, or digital video clips. A sound file's format is simply a recognized methodology for organizing and (usually) compressing the digitized sound's data bits and bytes into a data file. The structure of the file must be known, of course, before the data can be saved or later loaded into a computer to be edited and/or played as sound. The file name extension identifies which method of storage is used. There are many ways to store the bits and bytes that describe a sampled waveform sound.

The method used for consumer-grade music CDs is Linear Pulse Code Modulation (LPCM), often shortened to PCM. An audio CD provides up to 80 minutes of playing time, which is enough for a slow-tempo rendition of Beethoven's Ninth Symphony. Incidentally, being able to contain Beethoven's Ninth is reported to have been Philips's and Sony's actual size criterion during early research and development for determining the length of the sectors and ultimately the physical size of the compact disc format itself. The CD-ROM/XA (extended architecture) format for reading and writing CDs was developed later so you could put several recording sessions of music or data onto a single CD-R (recordable) disc. LPCM tracks from an audio CD are usually converted and stored on a computer in uncompressed AIFF (Audio Interchange File Format) or wave format (WAV) files when copied from the CD.

AIFF is historically used for Macintosh sound files. The WAV format was introduced by Microsoft when Windows was first released. Both formats contain uncompressed sound data. But there are huge numbers of sound file formats and "multimedia containers" that store sound data (more than three hundred different file name extensions are used for sound files), and often a converter is required to read or write sound files in the format you need. Hoo Technologies (www.hootech.com) offers MP3 to SWF, SWF/FLV to MP3, AIFF to MP3, MIDI to MP3, WMA to MP3, WAV to MP3, and OGG to MP3 converters.

Their AIFF to MP3 converter will read the following formats: 3G2, 3GP, 3GP2, 3GPP, 4XM, AAC, AC3, ADX, AFC, AIF, AIFC, AIFF, ALAW, AMR, AMV, APE, ASF, AU, AVI, AWB, CAF, CDA, CDATA, DIF, DIVX, DTS, DV, DVD, DVR-MS, DXA, FLAC, FLC, FLI, FLIC, FLV, FLX, GSM, GXF, H261, H263, H263+, H264, IT, KAR, M1A, M1V, M2A, M2TS, M2V, M4A, M4B, M4V, MID, MIDI, MJ2, MJPEG, MJPG, MKA, MKV, MLP, MLV, MMF, MO3, MOD, MOV, MP+, MP1, MP2, MP3, MP4, MPA, MPC, MPE, MPEG, MPG, MPGA, MPP, MPV, MTM, MTS, MTV, MVI, MXF, NSA, NSV, NUT, NUV, OGA, OGG, OGM, OGV, OGX, OMA, PSP, PSX, PVA, QCP, QT, RA, RAM, RM, RMI, RMVB, ROQ, RPL, S3M, SDP, SHN, SMK, SND, SOL, SPX, STR, SWF, TS, TTA,
UMX, VFW, VID, VMD, VOB, VOC, VQF, W64, WAV, WAVE64, WM, WMA, WMD, WMV, WV, XA, XM, XVID, XWMV, and YUV. And it will output to MP3, WAV, WMA, AAC, MP4, M4A (MPEG-4 Audio),

M4B (MPEG-4 AudioBook), OGG, AMR, and AWB formats. But rest easy—you will likely only work with a handful of sound file types. The MP3 format was developed by the Moving Picture Experts Group (MPEG) and evolved during the 1990s into the most common method for storing consumer audio. It incorporates a “lossy” compression algorithm to save space. An audio CD, for example, may hold an hour or so of uncompressed LPCM sound. That same CD, using MP3 compression, can store almost seven hours of the same music, but with a slight loss of quality.

WMA (Windows Media Audio) is a proprietary Microsoft format developed to improve MP3. OGG was developed as an open-source and royalty-free “container” for sound compressed using Vorbis algorithms similar to MP3—because the Vorbis sound data resides within an Ogg container, these audio files are normally called “Ogg Vorbis.” MP4 is a format based on Apple’s QuickTime movie (.mov) “container” model and is similar to the MOV format, which stores various types of media, particularly time-based streams such as audio and video.

The mp4 extension is used when the file streams audio and video together.

The m4a extension is used when the file contains only audio data. M4p files contain only audio, but are encrypted for Digital Rights Management (DRM). M4r files are used for ringtones on Apple’s iPhone. Other GSM (Global System for Mobile Communications) mobile phones use 3gp files for their ringtones, a format also based on the MPG container model. The AAC (Advanced Audio Coding) format, which is part of the MP4 model, was adopted by Apple’s iTunes store, and many music files are commercially available in this format. ACC is the default format for iPod, iPhone, PlayStation, Wii, Dsi, and many mobile phones including Motorola, Nokia, Philips, Samsung, Siemens, and Sony Ericsson.

The SWF format is a container for vector-based graphics and animations, text, video, and sound delivered over the Internet. Typically created using Adobe’s Flash, SWF files require a plug-in or player be installed in the user’s browser. Adobe claims that the Flash Player is installed in more than 98 percent of Web users’ browsers and in more than 800 million handsets and mobile devices. Flash video files (FLV) contain both a video stream and an audio stream, and the FLV format has been adopted by YouTube, Google, Yahoo, Reuters.com, BBC.com, CNN.com, and other news providers for Internet delivery of content.

A codec (compressor-decompressor) is software that compresses a stream of audio or video data for storage or transmission, then decompresses it for playback. There are many codecs that do this with special attention to the quality of music or voice after decompression. Some are “lossy” and trade quality for significantly reduced file size and transmission speed; some are “lossless,” so original data is never altered. While editing your audio files, be sure to save your files using a lossless format or codec— with repetitive saves in a lossy format, you will notice a quality degradation each time. A container format such as MP4, MOV, or OGG may encapsulate data streams that use one of many codecs available in that container.

MIDI vs. Digital Audio

In contrast to MIDI data, digital audio data is the actual representation of a sound, stored in the form of thousands of individual numbers *(samples).* The digital data represents the instantaneous amplitude (or loudness) of a sound at discrete slices of time. MIDI data is to digital audio data what vector or drawn graphics are to bitmapped graphics. That is, MIDI data is device dependent; digital data is not. Just as the appearance of vector graphics differs depending on the printer device or display screen, the sounds produced by MIDI music files depend on the particular MIDI device used for playback. Similarly, a roll of perforated player-piano score played on a concert grand would sound different than if played on a honkytonk piano. Digital data, on the other hand, produces sounds that are more or less identical regardless of the playback system. The MIDI standard lets instruments communicate in a well-understood language. MIDI has several advantages over digital audio and two huge disadvantages. First, the advantages:

- MIDI files are much more compact than digital audio files, and the size of a MIDI file is completely independent of playback quality. In general, MIDI files will be 200 to 1,000 times smaller than CD-quality digital audio files. Because MIDI files are small, they don't take up as much memory, disk space, or bandwidth.
- Because they are small, MIDI files embedded in web pages load and play more quickly than their digital equivalents.
- In some cases, if the MIDI sound source you are using is of high quality, MIDI files may sound better than digital audio files.
- You can change the length of a MIDI file (by varying its tempo) without changing the pitch of the music or degrading the audio quality. MIDI data is completely editable—right down to the level of an individual note. You can manipulate the smallest detail of a MIDI composition (often with submillisecond accuracy) in ways that are impossible with digital audio.
- Because they represent the pitch and length of notes, MIDI files can generally be converted to musical notation, and vice versa. This is useful when you need a printed score; in reverse, you can scan a printed score and convert it to MIDI for tweaking and editing. Now for MIDI's disadvantages:
- Because MIDI data does not represent sound but musical instruments, you can be certain that playback will be accurate only if the MIDI playback device is identical to the device used for production. Imagine the emotional humming chorus from *Madame Butterfly* sung by a chorus of Budweiser frogs—same score, wrong instrument. Even with the General MIDI standard (see the General MIDI table of instrument sounds in Table 4-3), the sound of a MIDI instrument varies according to the electronics of the playback device and the sound generation method it uses.
- Also, MIDI cannot easily be used to play back spoken dialog, although expensive and technically tricky digital samplers are available. In general, use MIDI in the following circumstances:
- Digital audio won't work because you don't have enough memory or bandwidth.
- You have a high-quality MIDI sound source.
- You have complete control over the machines on which your program will be delivered, so you know that your users will have high-quality MIDI playback hardware.
- You don't need spoken dialog. The most important advantage of digital audio is its consistent playback quality, but this is where MIDI is the least reliable! With digital audio you can be more confident that the audio track for your multimedia project will sound as good in the end as it did in the beginning when you created it. For this reason, it's no surprise that digital audio is used far more frequently than MIDI data for multimedia sound delivery. There are two additional and often more compelling reasons to work with digital audio:

- A wider selection of application software and system support for digital audio is available for both the Macintosh and Windows platforms.
- The preparation and programming required for creating digital audio do not demand knowledge of music theory, while working with MIDI data usually does require a modicum of familiarity with musical scores, keyboards, and notation, as well as audio production. In general, use digital audio in the following circumstances:
- You don't have control over the playback hardware.
- You have the computing resources and bandwidth to handle digital files.
- You need spoken dialog.

Adding Sound to Your Multimedia Project

The original 128K Macintosh, released in January 1984, was technically a multimedia-capable machine. It displayed bitmapped graphics (albeit in black and white) and, more significantly, boasted 8-bit digital audio capability right on the motherboard. In fact, the very first Macintosh actually introduced itself by voice when it was unveiled by Steve Jobs. Here's a little history: In order to use the Apple moniker, the original founders of Apple Computer, Inc., worked out an arrangement with the Beatles (yes, *those* Beatles). One part of that agreement stipulated that Apple Computer, Inc., would never venture into the music business.

To Steve Jobs and Steve Wozniak, working out of their garage in the late 1970s on a machine that could barely manage a convincing system beep, that clause probably seemed a harmless one. Little did they know that years later their computer and the Apple itself would become the most popular provider of music in the world through its iTunes facility.

The company did finally pay representatives of the Beatles about $30 million to settle the issue once and for all. Whether you're working on a Macintosh or in Windows, you will need to follow certain steps to bring an audio recording into your multimedia project. Here is a brief overview of the process:

1. Determine the file formats that are compatible with your multimedia authoring software and the delivery medium(s) you will be using (for file storage and bandwidth capacity).
2. Determine the sound playback capabilities (codecs and plug-ins) that the end user's system offers.
3. Decide what kind of sound is needed (such as background music, special sound effects, and spoken dialog). Decide where these audio events will occur in the flow of your project. Fit the sound cues into your storyboard (see Chapter 10), or make up a cue sheet.
4. Decide where and when you want to use either digital audio or MIDI data.
5. Acquire source material by creating it from scratch or purchasing it.
6. Edit the sounds to fit your project.

7. Test the sounds to be sure they are timed properly with the project's images. This may involve repeating steps 1 through 4 until everything is in sync. When it's time to import your compiled and edited sounds into your project, you'll need to know how your particular multimedia software environment handles sound data.

Each multimedia authoring program or web browser handles sound a bit differently, but the process is usually fairly straightforward: just tell your software which file you want to play and when to play it. This is usually handled by an importing or “linking” process during which you identify the files to play. Scripting languages such as revTalk (RunRev), Lingo (Director), and ActionScript (Flash) provide a greater level of control over audio playback, but you’ll need to know about the programming language and Environnment.

In multimedia authoring environments, it is usually a simple matter to play a sound when the user clicks a button, but this may not be enough. If the user changes screens while a long file is playing, for example, you may need to program the sound to stop before leaving the current screen. If the file to be played cannot be found, you may need to code an entire section for error handling and file location. Sample code is generally provided in both printed and online documentation for software that includes sound playback. For web pages, you will need to embed a player and point to your sound file using HTML code.

Space Considerations

The substantial amount of digital information required for highquality sound takes up a lot of storage space, especially when the quantity is doubled for two-channel stereo. It takes about 1.94MB to store 11 seconds of uncompressed stereo sound. If monaural sound is adequate for your project, you can cut your storage space requirement in half or get double the playing time in the same memory space. With compression codecs, you might be able to store the sound in one-eighth the space, but you will lose some fidelity. Further, to conserve space you can downsample, or reduce the number of sample slices you take in a second.

Many multimedia developers use 8-bit sample sizes at 22.05 kHz sampling rates because they consider the sound to be good enough (about the quality of AM radio), and they save immense amounts of digital real estate. The following formula will help you estimate your storage needs. If you are using two channels for stereo, double the result. (sampling rate * bits per sample) / 8 = bytes per second If you prefer to solve for kilobytes (KB), not bytes, then try:

sample rate * sample size / 8 * # seconds * 2 (if stereo) = file size in KB

For example, 60 seconds of stereo in Red Book Audio: 44.1 * 16 / 8 * 60 * 2 = 10,584KB 10.59MB This is an approximate result using 1000 instead of 1024 bytes per KB, but yielding the quick handy answer “…about ten and a half megabytes.”

You face important trade-offs when deciding how to manage digitized sound in your multimedia project. How much sound quality can you sacrifice in order to reduce storage? What compression techniques make sense? Will compressed sound work in your authoring platform? What is good enough but not amateurish? Can you get away with 8 bits at 11.025 kHz for voice mail, product testimonials, and voice-overs and then switch to higher sampling rates for music? Many people feel that MP3s files sampled at 128 Kbps provide decent audio quality for music, especially when played through small speakers. For better quality, sample your music at 192 Kbps.

Audio Recording

If your project requires CD-quality digitized sound at 44.1 kHz and 16 bits, you should hire a sound studio. High-fidelity sound recording is a specialized craft, a skill learned in great part by trial and error, much like photography. If you do decide to do it yourself at CD-quality levels, be prepared to invest in an acoustically treated room, high-end amplifiers and recording equipment, and expensive microphones. As already stated, there are many trade-offs involved in making multimedia.

For example, if you are satisfied with 22.05 kHz in your project or are constrained to this rate by storage considerations, any consumer-grade digital or analog recorder of reasonable quality will do fine. This, of course, also applies to conversations recorded from the telephone, where a sampling rate of 11.025 kHz is adequate. Noise reduction circuits and metal tapes are helpful to remove hiss, but at a sampling rate of 22.05 kHz you are only going to be digitizing audio frequencies as high as about 11 kHz, anyway. Both the high and low ends of the audio hearing spectrum are therefore less important to you, and that is OK, because those areas are precisely the add-value focus of very elaborate and expensive consumer equipment.

Digital audio tape (DAT) systems provide a tape-based 44.1 kHz, 16-bit record and playback capability. You may, however, find that DAT is high-fidelity overkill for your needs, because the recordings are too accurate, precisely recording glitches, background noises, microphone pops, and coughs from the next room.

A good editor can help reduce the impact of these noises, but at the cost of time and money. Mobile phones can often record audio (and video), and applications and hardware attachments are available to manage external microphones and file transfer. USB and flash memory recorders range in quality, some suitable for voice only, some generating compressed MP3 files, and some recording in CD-quality stereo. Recordings can be directly downloaded

Keeping Track of Your Sounds

In an elaborate project with many sounds, it is important to maintain a good database, keeping a physical track of your original material—just in case you need to revert to it when your disk drive crashes or you accidentally delete the work file. A database is particularly important because you may need to give your sound files such unhelpful names as janesEyesOpenWide. Aiff or Spanish03.wav; these names contain some clues about the files' actual content, but you may need a more descriptive cross-reference. You don't want to have to load and play many sound files just to find the one you need.

Audio CDs

The method for digitally encoding the high-quality stereo of the consumer CD music market is an international standard, called ISO 10149. This is also known as the Red Book Audio standard (derived simply from the color of the standard's book jacket). Developers of this standard claim that the digital audio sample size and sampling rate of Red Book Audio (16 bits at 44.1 kHz) allow accurate reproduction of all the sounds that humans can hear. Until recently, dedicated professional sound-studio equipment was used for this high-fidelity recording; today most off-the shelf computers will record and play 16-bit sampled sound at 44.1 kHz and at 48 kHz.

Converter and burning software such as Toast and CD-Creator from Roxio can translate the digital files of Red Book Audio found on consumer compact discs directly into a digital sound file formats such as MP3 or WAV. Unlike DVDs, audio CDs do not contain information about artists, titles, or tracklists of songs. But player software such as Apple iTunes and AOL Winamp will automatically link to a database on the Internet when you insert a music CD.

The precise length of your CD's Table of Contents (TOC) is then matched against the known TOC length for more than five million CDs containing more than 60 million songs. When it finds a match, the database service sends back what it knows about the CD you inserted. The database, formerly known as the Compact Disc Database or CDDB, was built up over the years by fans from all over the world submitting information about their favorite CDs. The database is currently maintained by Gracenote Media Recognition Service

Analog display standards

- The following analog broadcast video standards are commonly used around the world:
  1. NTSC
  2. PAL
  3. SECAM
  4. ATSC DTV

NTSC (National Television Standards Committee)

- The United States, Canada, Mexico, Japan, and many other countries used a system for broadcasting and displaying video that is based upon the specifications set forth by the 1992 National Television Standards Committee.
- These standards defined a method for encoding information into the electronic signal that uultimately created a television picture.
- As specified by the NTSC standard, a single frame of video was made up of 525 horizontal scan lines drawn onto the inside face of a phosphor-coated picture tube every 1/30th of a second by a fast-moving electron beam.
- The drawing occurred so fast that your eye would perceive the image as stable.
- The electron beam actually made two passes as it drew a single video frame-first it laid down all the odd-numbered lines, and then all the even-numbered lines.
- Each of these passes, which happens at the rate of 60 per second, or 60 Hz.) painted a field, and the two fields were then combined to create single frame at a rate of 30 frames per second (fs). Technically the speed is actually 29.97 Hz. This process of building a single frame from two fields was called **interlacing.**
- **Interlacing** is a technique that helps to prevent flikker on television screens.

- Computer monitors used a different **progressive-scan** technology, and drew the lines of an entire frame in a single pass, without interlacing them and without fliker.

PAL (Phase Alternate Line)

- The PAL system was used in United Kingdom, Western Europe, Australia, South Africa, China and South Africa.
- PAL increased the screen resolution to 625 horizontal lines, but slowed the scan rate 25 frames per second.
- As with NTSC, the even and odd lines were interlaced, each field taking 1/50 of a second to draw (50 Hz).

SECAM (Sequential Color and Memory)

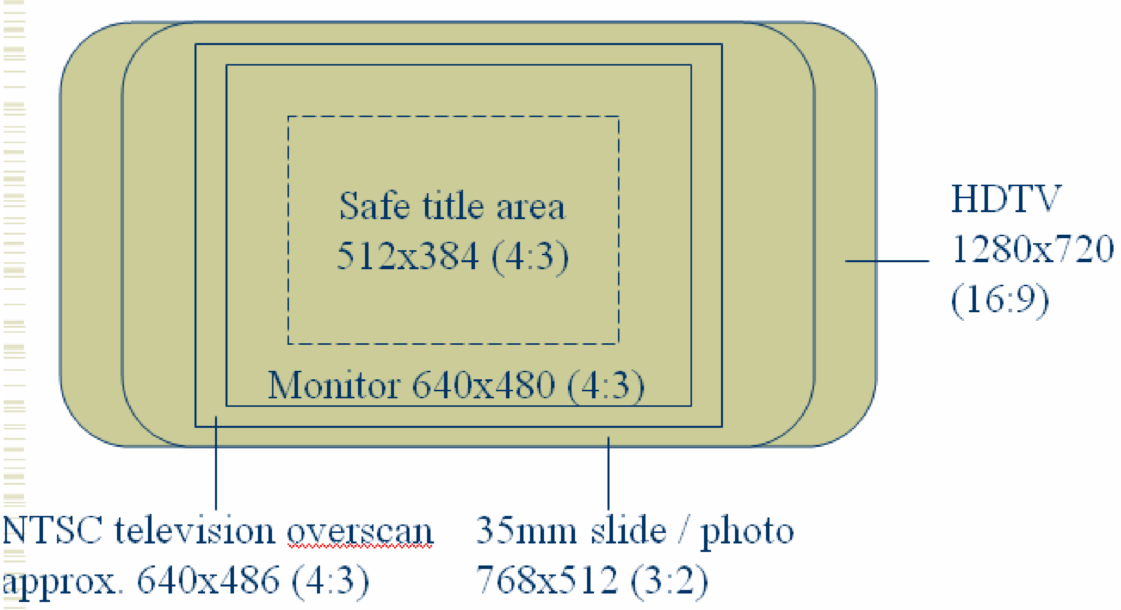
- The SECAM system was used in France, Eastern Europe, the former USSR, and a few other countries.
- Although SECAM is a 625 line, 50 Hz system, it differed greatly from both the NTSC and the PAL color systems in its basic technology and broadcast method.
- However, TV sets sold in Europe utilized dual components and could handle both PAL and SECAM systems.

ATSC DTV (Digital Television)

- **High-Definition Television** (or **HDTV**) is a digital television broadcasting system with higher resolution than traditional television systems (standard-definition TV, or SDTV). HDTV is digitally broadcast; the earliest implementations used analog broadcasting, but today digital television (DTV) signals are used, requiring less bandwidth due to digital video compression.
- This standard, which was slightly modified from both the Digital Television Standard (ATSC Doc. A/53) and the digital Audio Compression Standard (ATSC Doc. A/52), moved U.S. television from an analog to digital standard.
- It also provided TV stations with sufficient bandwidth to present four or five Standard Television (STV, providing the NTSC's resolution of 525 lines with a 3:4 aspect ratio, but in a digital signal) or on HDTV signals (providing 1,080 lines of resolution with a movie screens 16:9 aspect ratio).

- For multimedia producers , this emerging standard allowed for transmission of data to computers and for new ATV interactive services.
- HDTV provides high resolution in a 16:9 aspect ratio.
- This aspect ratio allows the viewing of Cinemascope and Panavision movies.
- The broadcast industry has promulgated an ultra high resolution, 1920x1080 interlaced format to become the corner stone of a new generation of high end entertainment centers, but the computer industry would like to settle on a 1280x720 progressive scan system for HDTV.
- As the 1920x1080 format provides more pixels than the 1280x720 standard, the refresh rates are quite different.
- The higher resolution interlaced format delivers only half the picture every 1/60 of a second, and because of the interlacing, on highly detailed images there is a great deal of screen fliker at 30 Hz.

- The computer people using only 1280x720 standard, saying that the picture quality is superior and steady.
- Both formats have been included in the HDTV standard by the Advanced Television Systems Committee (ATSC).



Digital display standards

- The following digital video standards are commonly used around the world:
  1. ATSC (Advanced Television System Comittee)
  2. DVB (Digital Video Broadcasting)
  3. ISDB (Integrated Services Digital Broadcasting)

ATSC (Advanced Television System Comittee)

- ATSC is the digital television standard for the United States, Canada, Mexico, Taiwan, and South Korea and others.
- It supports wide screen aspect ratio of 16:9 with images upto 1920x1080 pixels in size and number of other images sizes, allowing upto six, standard-definition-television "virtual channels" to be broadcast on a single TV station using the existing 6 MHz channel.
- It provides "theatre quality" because it uses Dolby Digital AC-3 format to provide 5.1 channel surround sound.

DVB (Digital Video Broadcasting)

- DVB is used mostly in Europe where the standards define the physical layer and the data link layer of a distribution system.
- **Digital Video Broadcasting** (**DVB**) is a suite of internationally accepted open standards for digital television.

- DVB systems distribute data using a variety of approaches, including by satellite (DVB-S, DVB-S2); also for distribution via cable (DVB-C); terrestrial television (DVB-T, DVB-T2) and via microwave using DTT(), MMDS(), and / or MVDS standards.
- These standards define the physical layer and data link layer of the distribution system.

ISDB (Integrated Services Digital Broadcasting)

- ISDB is used in Japan to allow radio and television stations to convert to digital format.
- The core standards of ISDB are ISDB-S (satellite television), ISDB-T (terrestrial) ISDB-C (cable) and 2.6 GHz band mobile broadcasting which are all based on MPEG-2 video and audio coding as well as the transport stream described by the MPEG-2 standard, and are capable of high definition television (HDTV).

Digital Audio

Digital audio is created when you represent the characteristics of a sound wave using numbers—a process referred to as digitizing. You can digitize sound from a microphone, a synthesizer, existing recordings, live radio and television broadcasts, and popular CD and DVDs. In fact, you can digitize sounds from any natural or prerecorded source. Digitized sound is sampled sound. Every *n*th fraction of a second, a sample of sound is taken and stored as digital information in bits and bytes. The quality of this digital recording depends upon how often the samples are taken (sampling rate or frequency, measured in kilohertz, or thousands of samples per second) and how many numbers are used to represent the value of each sample (bit depth, sample size, resolution, or dynamic range).

The more often you take a sample and the more data you store about that sample, the finer the resolution and quality of the captured sound when it is played back. Since the quality of your audio is based on the quality of your recording and not the device on which your end user will play the audio, digital audio is said to be device independent. The three sampling rates most often used in multimedia are 44.1 kHz (CD-quality), 22.05 kHz, and 11.025 kHz. Sample sizes are either 8 bits or 16 bits. The larger the sample size, the more accurately the data will describe the recorded sound. An 8-bit sample size provides 256 equal measurement units to describe the level and frequency of the sound in that slice of time.

A 16-bit sample size, on the other hand, provides a staggering 65,536 equal units to describe the sound in that same slice of time. As you can see in Figure 4-1, slices of analog waveforms are sampled at various frequencies, and each discrete sample is then stored either as 8 bits or 16 bits (or more) of data.

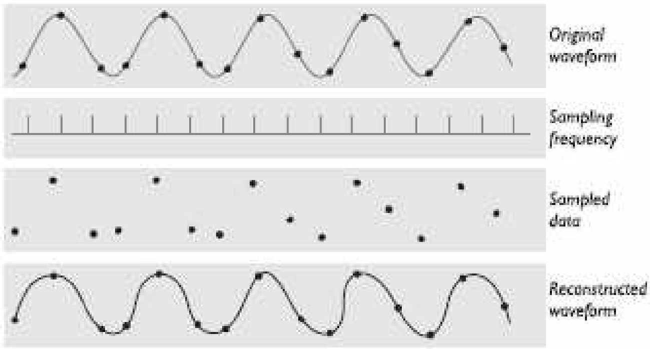


Figure 4-1 It is impossible to reconstruct the original waveform if the sampling frequency is too low.

The value of each sample is rounded off to the nearest integer (quantization), and if the amplitude is greater than the intervals available, clipping of the top and bottom of the wave occurs (see Figure 4-2). Quantization can produce an unwanted background hissing noise, and clipping may severely distort the sound

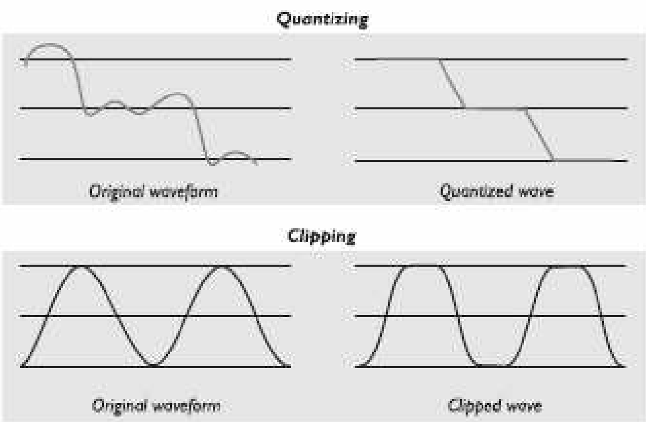


Figure 4-2 Examples of quantizing and clipping

Video Recording & Tape Format

Composite Analog Video:

1. Composite video combines the luminance and chroma information from the video signal.
2. As a result, it produces the lowest quality video and is most susceptible to generation loss, the loss of quality that occurs as you move from original footage to edit master to copy.
3. This recording format was used for consumer analog video recording tape formats and was never adequate for most multimedia productions.

Component video separates the luminance and chroma information in order to improve the quality of the video and to decrease generation loss.

S-Video (Y/C)

- In S-Video, color and luminance information are kept on two separate tracks (Y/C). The result is a definite improvement in picture quality over composite video.
- This standard is used in S-VHS.
- It provides sharper image and slightly less generation loss from master to copy

Three-Channel Component (Y/R-Y/B-Y, Y/U/V)

- Professional quality requires further separation in the video signal, usually divided between luminance (Y) and two channels of chroma, but sometimes divided between the Red/ Green/ Blue primary additive colors.

Composite Digital

- Composite digital recording formats combine the luminance and chroma information.
- They sample the incoming waveforms and encode the information in binary (0/1) digital code.
- It also improves color and image resolution, and having the recording in digital format eliminates generation loss when copies are made.
- The D-2 format was developed primarily by Ampex and Sony.
- The D-3 format was developed primarily by Panasonic.
- The major advantage over the component digital format is the lower cost in equipment.
- Although this equipment is expensive enough to find a market only at the professional level and has not been used much in multimedia production

Component Digital

- Sony released the first equipment in this category with its D-1 format.
- D-1 format is an uncompressed format, so it has very high quality image, and need large storage.
- Because of its high quality, D-1 is used for NTSC video and is the mastering standard of choice among high end editing facilities.
- This quality comes with an extremely high price tag.
- The result is that this format really only fits super-high-end broadcast projects and not your standard multimedia title.
- The other digital component formats, including **DCT** (from Ampex), **D-5** (from Panasonic), and **Digital Betacam** (from Sony) also used.
- Featuring four channels of CD-quality audio, **Digital Betacams** video quality is almost equal to **D-1** digital.
- Although another digital component format **DV** format provided a high quality level and price point that attracted and it compresses the video allowing for the use of smaller tape width.
- DVCPRO and DVCAM, professional formats that use the DV recording format, but provide professional features on the camcorders and tape decks..

ATSC Digital TV

- These standards provide for both digital standard TV (STV) and High Definition TV (HDTV) recordings that can be broadcast by digital TV transmitters to digital TV receivers.

- ATSC standards also provide for Enhanced TV (ETV), potentially bringing the interactivity of multimedia and the web to broadcast television.

Optimizing video files for CD-ROM

- CD-ROMs provide an excellent distribution medium for computer based video, because they are inexpensive to mass produce, they can store grant quantities of information.
- CD-ROM players offer slow data transfer rates, but adequate video transfer can be achieved by taking care to properly prepare your digital video files.
- Without great care, these digital files may display poorly in low-bandwidth / high-compression environments:
  - Limit the amount of synchronization required between the video and audio. With Microsoft's AVI files, the audio and video data are already interleaved, so this is not a necessity, but with QuickTime files, you should "flatten" your movie. Flattening means that you interleaved the audio and video segments together.
  - Use regularly spaced key frames, 10 to 15 frames apart, and temporal compression can correct for seek time delays. Seek time is how long it takes the CD-ROM player to locate specific data on the CD-ROM disc.
  - The size of video window and the frame rate you specify dramatically affect performance. In QuickTime, 20 frames per second played in a 160 x 120 pixel window is equivalent to playing 10 frames per second in 320 x 240 window. The more data that has to be decompressed and transferred from the CD-ROM to the screen, the slower the playback.
  - Although interleaving CD-quality audio into your video production will theoretically yield the highest-quality sound, the volume of data required may be too great to transfer from the CD-ROM in real-time. Try a lower sampling rate and sample size to reduce the quantity of audio data.
  - The software compression algorithm you specify will make a dramatic difference in performance. The Sorenson algorithm or codec, available within both AVI and QuickTime, is optimized for CD-ROM playback.
  - Defragment your files before burning the master.
  - If you are working with QuickTime, consider using a specialized application such as Media Cleaner Pro to automatically optimize your digital video file playback from CD-ROM.

DVD Recordable

- DVD recordable are used for consumer audio and video recording.

- Three formats were: -R/RW (dash), +R/RW (plus), -RAM (random access memory).
- Dual Layer recordings allows DVD-R and DVD+R discs to store more data, upto 8.5 GB per disc, compared with 4.7GB for single layer discs.

Considerations for preparing video for the Web and CD-ROM

Considerations:

1. Codecs are digital video and video compression schemes that compress a video for delivery and then decode it during playback.
2. Streaming audio and video starts playback as soon as enough data has transferred to the user's computer to sustain this playback.
3. The MPEG standards provide good media encoding abilities. MPEG-4 includes numerous multimedia capabilities and may become the preferred standard for video and audio in multimedia.
4. CD-ROMs provide an excellent distribution medium for computer-based video.
5. When preparing video for CD-ROM distribution, interleave the audio tracks with the video track, use key frames every 10 to 15 frames, and keep the size of the video window small. The Sorenson codec is optimized for CD-ROM playback.

4. Multimedia Authoring Tools

Making Instant Multimedia

Making of your Multimedia Project:

1. Use templates that people have already creates to set up your production**.** These can include appropriate styles for sorts of data, font sets, color arrangements, and particular page set ups that will save your time.
2. Use wizards when they are available. They may save you much time and pre-set-up work.
3. Use named styles, because if you take the time to create your own it will really slow you down. Unless your client specifically requests a particular style, you will save a great deal of time using something already created, usable, and legal.
4. Create tables, which you can build with a few keystrokes in many programs, and it makes the production look credible.
5. Help readers find information with tables of contents, running headers and footers and indexes.
6. Improve document appearance with bulleted and numbered lists and symbols.
7. Allow for a quick-change replacement using the global change feature.
8. Reduce grammatical errors by using the grammar and spell checker provided with the software. Do not rely on that feature, though, to set all things right, you still need to proofread everything.
9. Include identifying information in the file name so you can find the file later.

Multimedia Authoring Tools

A multimedia authoring tool is a program that helps you write multimedia applications. A multimedia authoring tool enables you to create a final application merely by linking together objects, such as a paragraph of text, an illustration, or a song. They are used exclusively for applications that present a mixture of textual, graphical, and audio data.
With multimedia authoring software you can make video productions including CDs and DVDs, design interactivity and user interface, animations, screen savers, games, presentations, interactive training and simulations.

There are five distinct stages of multimedia authoring:

1. **Analysis:** What do you need to do and what do you use to do it.
2. **Design:** Create storyboards to tell the story of the project.
3. **Development:** Incorporate data and set it up as a prototype or model.
4. **Evaluation:** When the prototype application works the way you want it to, test it again, fine-tune it and then review your work.
5. **Distribution:** When it is ready to go, make it real. Package and distribute it.

Types of Authoring Tools

Each multimedia project you undertake will have its own underlying structure and purpose and will require different features and functions. E-learning modules such as those seen on PDAs, MP3 players, and intra-college networks may include web-based teaching materials, multimedia CD-ROMs or web sites, discussion boards, collaborative software, wikis, simulations, games, electric voting systems, blogs, computer-aided assessment, simulations, animation, blogs, learning management software, and e-mail. This is also referred to as distance learning or blended learning, where online learning is mixed with face-to-face learning.

The various multimedia authoring tools can be categorized into three groups, based on the method used for sequencing or organizing multimedia elements and events:

- Card- or page-based tools
- Icon-based, event-driven multimedia and game-authoring tools
- Time-based tools

Card- and Page-Based Authoring Tools

Card-based or page-based tools are authoring systems, wherein the elements are organized as pages of a book or a stack of cards. Thousands of pages or cards may be available in the book or stack. These tools are best used when the bulk of your content consists of elements that can be viewed individually, letting the authoring system link these pages or cards into organized sequences.

You can jump, on command, to any page you wish in the structured navigation pattern. Page-based authoring systems such as LiveCode from Runtime Revolution (www.runrev.com) and ToolBook (www.toolbook.org) contain media objects: buttons, text fields, graphic objects, backgrounds, pages or cards, and even the project itself. The characteristics of objects are defined by properties (highlighted, bold, red, hidden, active, locked, and so on). Each object may contain a programming script, usually a property of that object, activated when an event (such as a mouse click) related to that object occurs.

Events cause messages to pass along the hierarchy of objects in the project; for example, a mouse-clicked message could be sent from a button to the background, to the page, and then to the project itself. As the message traveled, it looks for handlers in the script of each object; if it finds a matching handler, the authoring system then executesthe task specified by that handler. Following are some typical messages that might pass along the object hierarchy of the LiveCode and ToolBook authoring systems: Now let's look at specific examples. To go to the next card or page when a button is clicked, place a message handler into the script of that button. An example in RunRev's LiveCode language would be:

```
on mouseUp
go next card
end mouseUp
```

An example in ToolBook's OpenScript language would look like:

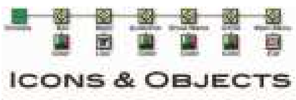
```
to handle buttonUp
go next page
end buttonUp
```

The handler, if placed in the script of the card or page, executes its commands when it receives a “mouseUp” or “buttonUp” event message that occurs at any location on the card or page—not just while the cursor is within the bounds of a button. Card- and page-based systems typically provide two separate layers on each card: a background layer that can be shared among many cards, and a foreground layer that is specific to a single card.

Icon- and Object-Based Authoring Tools

Icon- or object-based, event-driven tools are authoring systems, wherein multimedia elements and interaction cues (events) are organized as objects in a structural framework or process. Icon- or object-based, event-driven
tools simplify the organization of your project and typically display flow diagrams of activities along branching paths. In complicated navigational structures, this charting is particularly useful during development. Icon-based, event-driven tools provide a visual programming approach to organizing and presenting multimedia. First you build a structure or flowchart of events, tasks, and decisions, by dragging appropriate icons from a library.

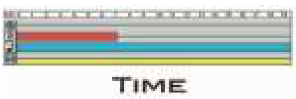
These icons can include menu choices, graphic images, sounds, and computations. The flowchart graphically depicts the project’s logic. When the structure is built, you can add your content: text, graphics, animation, sounds, and video movies. Then, to refine your project, you edit your logical structure by rearranging and fine-tuning the icons and their properties.



with icon-based authoring tools, non-technical multimedia authors can build sophisticated applications without scripting. In Authorware from Adobe, by placing icons on a flow line, you can quickly sequence events and activities, including decisions and user interactions. These tools are useful for storyboarding, as you can change sequences, add options, and restructure interactions by simply dragging and dropping icons. You can print out your navigation map or flowchart, an annotated project index with or without associated icons, design and presentation windows, and a cross-reference table of variables.

Time-Based Authoring Tools

Time-based tools are authoring systems, wherein elements and events are organized along a timeline, with resolutions as high as or higher than 1/30 second. Time-based tools are best to use when you have a message



played back at a speed that you can set. Other elements (such as audio events) are triggered at a given time or location in the sequence of events. The more powerful time-based tools let you program jumps to any location \in a sequence, thereby adding navigation and interactive control. Each tool uses its own distinctive approach and user interface for managing events over time.

Many use a visual timeline for sequencing the events of a multimedia presentation, often displaying layers of various media elements or events alongside the scale in increments as precise as one second. Others arrange long sequences of graphic frames and add the time component by adjusting each frame's duration of play. Flash Flash is a time-based development environment. Flash, however, is also particularly focused on delivery of rich multimedia content to the Web. With the Flash Player plug-in installed in more than 95 percent of the world's browsers, Flash delivers far more than simple static HTML pages. ActionScript, the proprietary, under-the-hood scripting language of Flash, is based upon the international ECMAScript standard (www.ecmainternational. org) derived from Netscape's original JavaScript. Director Adobe's Director is a powerful and complex multimedia authoring tool with a broad set of features to create multimedia presentations, animations, and interactive multimedia applications. It requires a significant learning curve, but once mastered, it is among the most powerful of multimedia development tools. In Director, you assemble and sequence the elements of your project, called a "movie," using a Cast and a Score.

The Cast is a multimedia database containing still images, sound files, text, palettes, QuickDraw shapes, programming scripts, QuickTime movies, Flash movies, and even other Director files. You tie these Cast members together using the Score facility, which is a sequencer for displaying, animating, and playing Cast members, and it is made up of frames that contain Cast members, tempo, a palette, timing, and sound information. Each frame is played back on a stage at a rate specified in the tempo channel. Director utilizes Lingo, a full-featured object-oriented

Storyboarding

Preplanning a video project is a factor that cannot be ignored without costing time loss, lots of unnecessary aggravation, and money that would be better spent elsewhere. Successful video production, of any sort, deserves the time it takes to make a plan to carry it out. It may take a little time at first, but you'll find it to be very helpful in the long run. Storyboards are like any sequential comic you read daily. Every day there are three or four panels showing a progression of story or information.

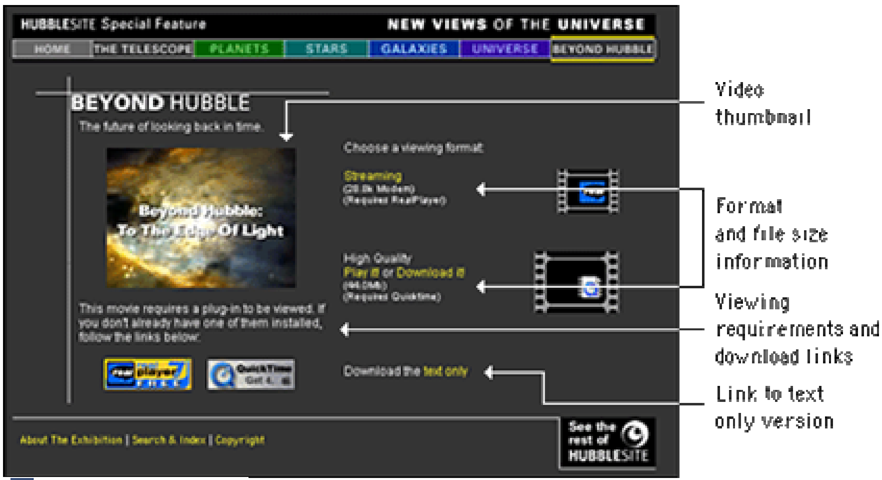
Take the time to structure your production by writing it down and then engineer a sequential group of drawings showing camera and scene, shooting angles, lighting, action, special effects, and how objects move through from start to finish. A storyboard can get everyone on one page quickly.

Media Design

The combination of low-bandwidth considerations and primitive interface options creates interesting design challenges for Web developers who wish to incorporate multimedia elements into their sites. Designers need to inform users when they are entering a high-bandwidth area and give them the tools they need to control their experience once in the area.

Inform your users

One aspect of the Web is that you don't always know where you're going or what you'll find there. For some this uncertainty is exciting. For many, it is annoying, particularly when a long wait is involved. Most frustrating, perhaps, is when you finally receive the requested page only to find that is not what you expected or that it contains materials in a format you are not set up to view. With content that is as technologically demanding as multi-media, it is especially important to give users enough information to make an informed decision *before they click*, so that they know what to expect and are prepared to receive your materials.

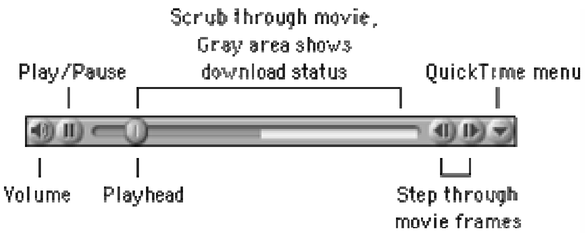


[hubble.stsci.edu]

High-demand content such as large multimedia files should not be part of your basic page design. These materials should appear on secondary pages that are described and can be accessed from the main pages of your site. Make the menu page a plain HTML page that loads quickly and does not require special software. Include descriptive information about the materials along with previews such as still shots from the video. Include the run time for time-based media, and include the file size for materials that download. In addition, fully explain any special software requirements for accessing the materials and provide a download link. Your users should have a clear idea of what your materials are before they begin to download. With a menu interface, users can confirm that their systems are properly configured and that they have enough bandwidth, time, and patience to load the materials.

Provide controls

Be sure to give users status information and controls when you are presenting multimedia materials. The QuickTime controller bar is an extremely effective interface element that provides both controls and status information. It allows users both to adjust the volume control and to play, stop, and scrub through a movie, and it provides information about the movie's download status.



If you don't include controls, users will hit your page with no way to control their viewing environment. For example, if a visitor is looking at your page at a public workstation and you have looping bird calls as a background sound without any control options, the visitor will experience an unsettling (and potentially embarrassing) moment when he or she cannot control interaction with your site. Many users in this situation will simply close the browser window to make the sound stop, which means that they never get to see the page content.

When designing a media interface, let interaction with your media be entirely user-driven. Always include user controls, such as a media controller bar, and make sure that users have a way to turn it off. Avoid prescribed playback options like auto play or looping that take control from the user. With auto play, for example, media files begin playing when a Web page is loaded. If the page has other elements, such as descriptive text, the user who wants just the text will find the video distracting. Design your media interface so that files play only when the user explicitly elects to initiate playback.

Developing Multimedia Packages

With the introduction of computers in the educational and instructional settings the design development and utilization of Multimedia packages is assuming importance. The format required for instructional settings vary from situation to situation and careful planning in terms of storyboard, lesson planning and software application. Gender being a complex subject matter with divergent viewpoints is selected for developing multimedia package for instructional purpose. The design of different screens, color combinations, music and voice over options are discussed in this paper

5. Designing and Producing

Designing the Structure

A multimedia project is no more than an arrangement of text, graphic, sound, and video elements (or *objects*). The way you compose these elements into interactive experiences is shaped by your purpose and messages. How you organize your material for a project will have just as great an impact on the viewer as the content itself. Since the explosive growth of the World Wide Web and proliferation of millions and millions of multimedia- capable HTML documents that can be linked to millions of other similar documents in the cyberspace of the Web, your designs and inventions may actually contribute to the new media revolution:

other creators may discover your work and build upon your ideas and methods. On some projects, you may be both the designer and the programmer. This can work well because you will understand how the design features you choose will actually be implemented. Indeed, your design will be tempered, if not defined, by your programming and coding skills, and you will be less likely to specify features impossible or overly difficult to realize.

Navigation

Mapping the structure of your project is a task that should be started early in the planning phase, because navigation maps outline the connections or links among various areas of your content and help you organize your content and messages. A navigation map (or site map) provides you with a table of contents as well as a chart of the logical flow of the interactive interface. While with web sites a site map is typically a simple hierarchical table of contents with each heading linked to a page, as a more detailed design document your map may prove very useful to your project, listing your multimedia objects and describing what happens when the user interacts.

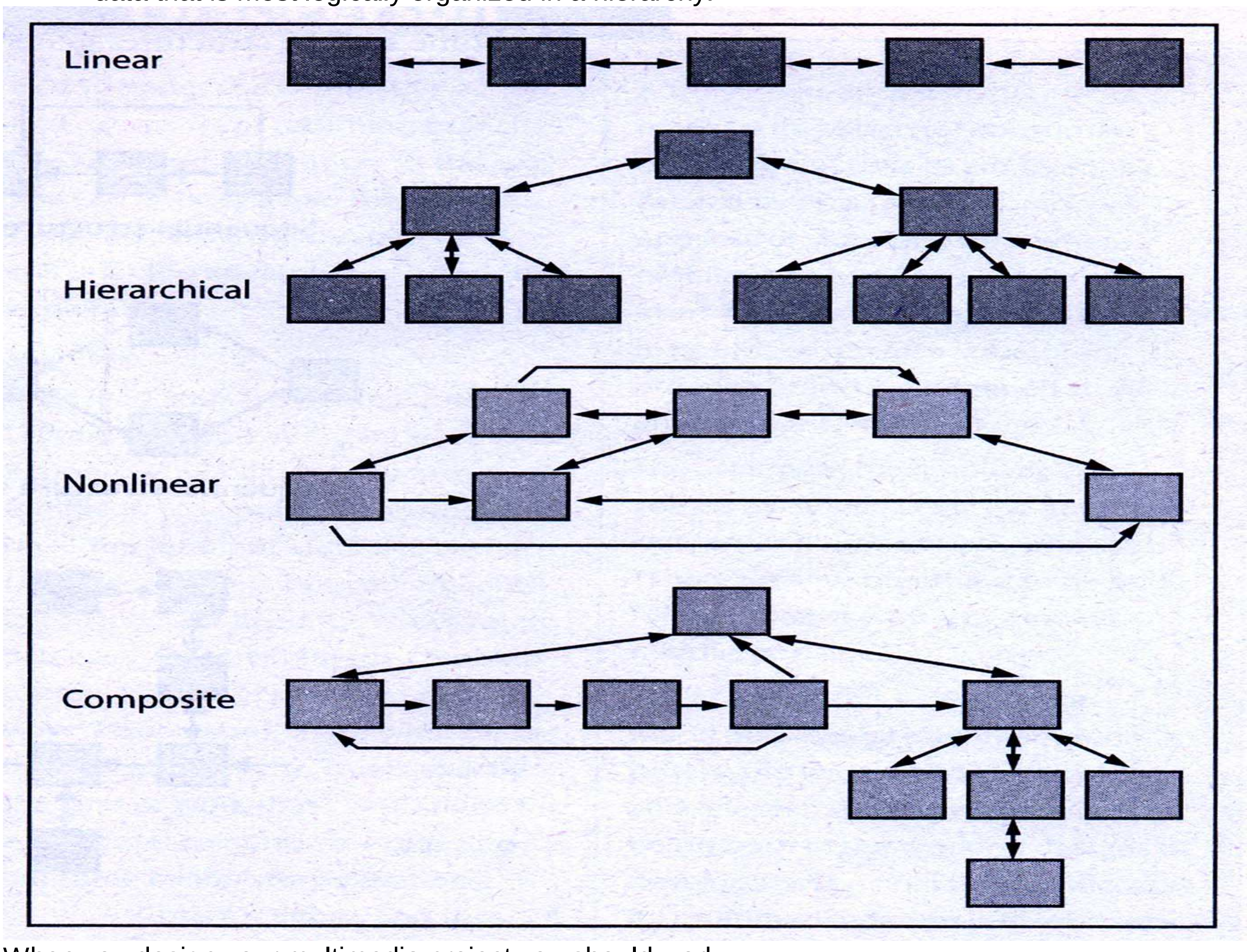
Just as eight story plots might account for 99 percent of all literature ever written (boy meets girl, protagonist versus antagonist, etc.), a few basic structures for multimedia projects will cover most cases: linear navigation, hierarchical navigation, nonlinear navigation, and composite navigation. Figure 10-1 illustrates the four fundamental organizing structures used in multimedia projects, often in combination:

- Linear Users navigate sequentially, from one frame or bite of information to another.
- Hierarchical Also called “linear with branching,” since users navigate along the branches of a tree structure that is shaped by the natural logic of the content.
- Nonlinear Users navigate freely through the content of the project, unbound by predetermined routes.
- Composite Users may navigate freely (nonlinearly) but are occasionally constrained to linear presentations of movies or critical information and/or to data that is most logically organized in a hierarchy.

The method you provide to your viewers for navigating from one place to another in your project is part of the user interface. The success of the user interface depends not only upon its general design and graphic art implementations but also upon myriad engineering details—such as the position of interactive buttons or hot spots relative to the user's current activity, whether these buttons "light up," and whether you use standard Macintosh or Windows pulldown menus. A good user interface is critical to the overall success of your project.

The nature of your interface will vary depending on its purpose: browsing, database access, entertainment, information, instruction, reference, marketing, and gaming projects require different approaches and different navigation strategies.

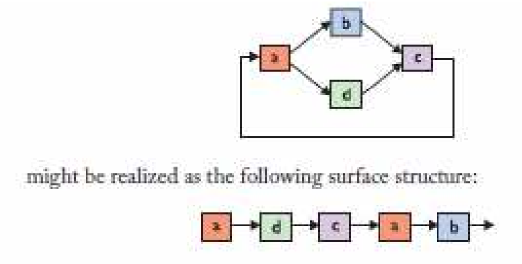
- The basic structures for Multimedia project covers the following cases: **Linear, Hierarchical, Nonlinear, and composite.**
- **Linear:** Users navigate sequentially, from one frame or byte of information to another.
- **Hierarchical:** It is also called "Linear with branching", since users navigate along the branches of a tree structure that is shaped by the natural logic of the content.
- **Nonlinear:** Users navigate freely through the content of the project, unbound by predetermined routes.
- **Composite:** Users may navigate freely (nonlinearly), but are occasionally constrained to linear presentations of movies or critical information's and/ or to data that is most logically organized in a hierarchy.



When you design your multimedia project you should work with **two types of structures:**

1. **Depth Structure**
2. **Surface Structure**

- **Depth Structure:** It represents the complete navigation map and describes all the links between all the components of your project.
- **Surface structure:** It represents the structures actually realized by a user while navigating the depth structure.



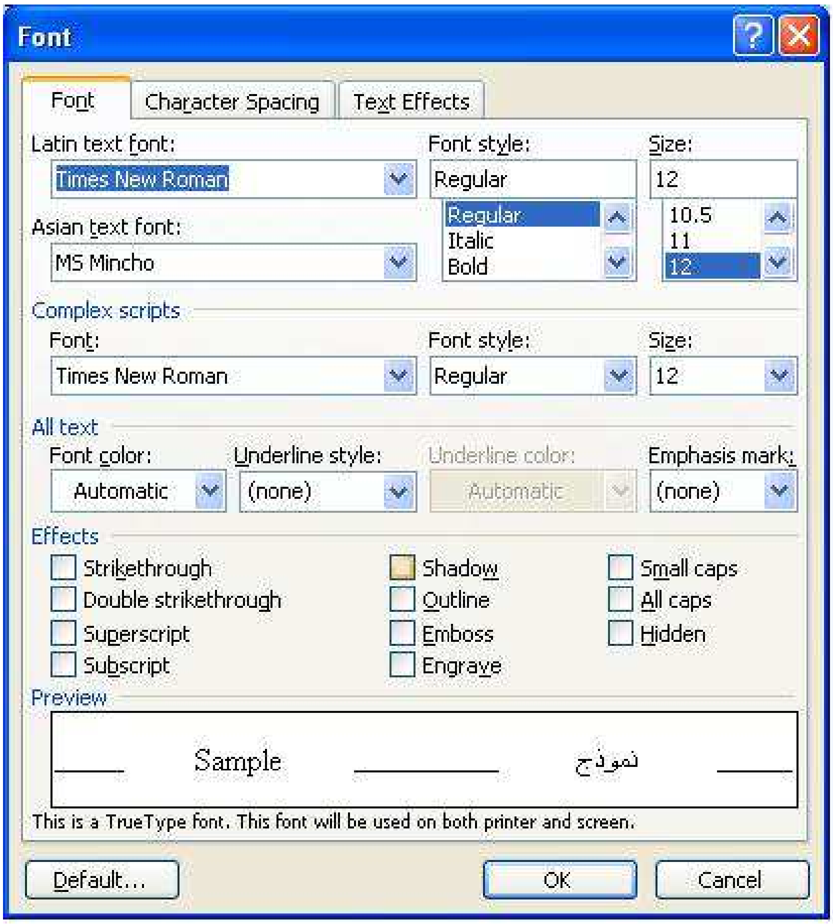
When you design your navigation map, it helps to think about surface structure—to view the product from a user's perspective. Surface structures are of particular interest to marketing firms in tracking users' routes through a web site to determine the effectiveness of the site's design and to profile a user's preferences. When a user's preferences are known, a custom web site experience can be dynamically tailored and delivered to that user.

Acquisition and management of such profiling data is a hot topic, with privacy advocates claiming the personal information revealed in these surface structures is akin to a person's medical and health records.

Hot Spots, Hyperlinks, and Buttons

- **Hot spot** is a region of high or special activity within a larger area of low or normal activity.
- Most multimedia authoring systems allow you to make any part of the screen, or any object, into a **hot spot.**
- When user click a hot spot at that location, something happens, which makes multimedia not just interactive but also exciting.
- Hot spots can be given more specific names based upon either their function or form.
- If the hot spot connects the user to another part of the document or program or to a different program or web site, it is referred to as a **hyperlink.**
- If the hot spot is a graphic image designed to look like a push button or toggle switch, it is called a **button.**
- **A button** is more formally defined as a meaningful graphic image that you click to make something happens.

- The term **“tab”** is not a button but it’s a **hot spot.** The clickable tab labels as the top of the illustration below are hot spots, but are not buttons. The real buttons are along the bottom:



- There are three categories of hot spots based upon the form in which they appear: **Text, graphic, or icon.**
- **Text:** A text is a hypertext. You may use hyperlink to other web pages.
- **Graphics:** You may create graphics, by using authoring tools, which should be used as a hot spot.
- **Icons:** Icons are graphic objects designed specifically to be meaningful buttons and are usually small.
- HTML do not directly support interactive buttons that follow the rules of good interface design.
- But you can make plain and animated buttons for your HTML documents on the web using plug-ins such as Flash or JavaScript.
- A JavaScript in an HTML document may be used to replace one image with another on mouseOver or hover.

- A simplest hot spots on the web are the text anchors that link a document to other document.
- Larger Images should may be sanctioned into hot areas with associated links, these are called image maps.

- Hyperlink to a **text, image, icon** is a hot spot.
- Examples of How text, image and Icon acts as hot spot.

Designing the User Interface

The user interface of your multimedia product is a blend of its graphic elements and its navigation system. If your messages and content are disorganized and difficult to find, or if users become disoriented or bored, your project may fail. Poor graphics can cause boredom. Poor navigational aids can make viewers feel lost and unconnected to the content; or, worse, viewers may sail right off the edge and just give up and quit the program.

- The user interface of your multimedia product is a collection of multimedia elements and navigation system.
- While designing the user interface you need to consider several factors, i.e. designing the Graphical User Interface, Designing the Audio Interface.

Graphical Approaches:

- Designing excellent computer screens requires a special set of fine art skills.
- Graphic artist must upgrade his / her knowledge as the technology changes. They must stay informed about the rapidly changing canvas of new features, techniques, applications, and creative tools.
- Following are **some graphical approaches that get good result:**
- **Things That Work:**
- Neatly executed contrasts: big/ small, heavy/ light, bright/ dark, thin/ thick etc.
- Simple and clean screen with lots of spaces.
- Use shadows in various shapes.
- Gradients
- Reversed graphics to emphasize important text or images.
- Shaded objects and text in 2-D and 3-D.

Following are **some mistakes that you have to avoid:**
Things to Avoid:

- Clashes of color
- Using picture with a lot of contrast.
- Requiring more than two buttons to quit.
- Too many numbers.
- Too many words.

Novice/Expert Modes

Be aware that there are two types of end users: those who are computer literate and those who are not. Creating a user interface that will satisfy both types has been a design dilemma since the invention of computers. The simplest solution for handling varied levels of user expertise is to provide a modal interface, where the viewer can simply click a Novice/ Expert button and change the approach of the whole interface—to be either more or less detailed or complex.

Modal interfaces are common on bulletin boards, for example, allowing novices to read menus and select desired activities, while experts can altogether eliminate the timeconsuming download and display of menus and simply type an activity code directly into an executable command line. Both novices and experts alike may quickly learn to click the mouse and skip the annoying ragtime piece you chose for background music.

Unfortunately, in multimedia projects, modal interfaces are not a good answer. It's best to avoid designing modal interfaces because they tend to confuse the user. Typically, only a minority of users are expert, and so the majority are caught in between and frustrated. The solution is to build your multimedia project to contain plenty of navigational power, providing access to content and tasks for users at all levels, as well as a help system to provide some hand-holding and reassurance. Present all this power in easy-to-understand structures and concepts, and use clear textual cues.
Above all, keep the interface simple! Even experts will balk at a complex screen full of tiny buttons and arcane switches, and will appreciate having neat and clean doorways into your project's content.

GUIs

The Macintosh and Windows graphical user interfaces (GUI, pronounced "gooey") are successful partly because their basic point-and-click style is simple, consistent, and quickly mastered. Both these GUIs offer built-in help systems, and both provide standard patterns of activity that produce standard expected results. The following actions, for example, are consistently performed by similar keystrokes when running most programs on the Macintosh or in Windows:

Action	Macintosh Keystroke	Windows Keystroke
New file	⌘-N	ALT-F-N or CTRL-N
Open file	⌘-O	ALT-F-O or CTRL-O
Save file	⌘-S	ALT-F-S or CTRL-S
Quit	⌘-Q	ALT-F-X or CTRL-Q
Undo	⌘-Z	ALT-E-U or CTRL-Z
Cut	⌘-X	ALT-E-T or CTRL-X
Copy	⌘-C	ALT-E-C or CTRL-C
Paste	⌘-V	ALT-E-R or CTRL-V

For your multimedia interface to be successful, you, too, must be consistent in designing both the look and the behavior of your human interface. Multimedia authoring systems provide you with the tools to design and implement your own graphical user interface from scratch. Be prudent with all that flexibility, however. Unless your content and messages are bizarre or require special treatment, it's best to stick with accepted conventions for button design and grouping, visual and audio feedback, and navigation structure.

Stick with real-world metaphors that will be understood by the widest selection of potential users.

For example, consider using the well-known trash can for deleting files, a hand cursor for dragging objects, and a clock or an hourglass for pauses. If your material is time-oriented, develop metaphors for past, present, and future. If it is topic-oriented, choose metaphors related to the topics themselves. If it is polar (the pros and cons of an issue, for example), choose relevant contrasting images.

Users like to be in control, so avoid hidden commands and unusual keystroke/mouse click combinations. Design your interface with the goal that no instruction manual or special training will be required to move through your project. Users do not like to have to remember keywords or special codes, so always make the full range of options easily available as interactive buttons or menu items.

And finally, users do make mistakes, so allow them a chance to escape from inadvertent or dangerous predicaments ("Do you really want to delete? Delete/Escape"). Keep your interface simple and friendly.

Graphical Approaches

Designing excellent computer screens requires a special set of fine art skills, and not every programmer or graduate in fine arts may be suited to creating computer graphics. Like programmers who must keep up with current operating systems and languages, computer graphic artists must also stay informed about the rapidly changing canvas of new features, techniques, applications, and creative tools. The artist must make broad design choices: cartoon stick figures for a children's game, rendered illustrations for a medical reference, scanned bitmaps for a travel tour of Europe.

The graphic artwork must be appropriate not only for the subject matter, but for the user as well. Once the approach is decided, the artist has to put real pixels onto a computer screen and do the work. A multimedia graphic artist must always play the role of the end user during the design and rendering process, choosing colors that look good, specifying text fonts that "speak," and designing buttons that are clearly marked for what they do.

Audio Interfaces

A multimedia user interface may include important sound elements that reflect the rhythm of a project and may affect the attitude of your audience. Sounds can be background music, special effects for button clicks, voice-overs, effects synced to animation, or they may be buried in the audio track of a video clip. The tempo and style of background music can set the "tone" of a project. Vivaldi or Bach might be appropriate for a banking or investment annual report delivered on DVD.

Comic laughs and screeching effects might be appropriate for a clothing web site aimed at preteens. Choose music that fits the content and the atmosphere you wish to create. In all cases, use special effects sparingly. Always provide a toggle switch to disable sound. (Many AOL users prefer to disable the "You've Got Mail!" voice, for example.) And always test a project that contains sound with potential users.

A Multimedia Design Case History

This section presents an example of the design process for a simple multimedia project about the construction and launch of a 31-foot ocean-going sailboat. This project was initially crafted in SuperCard (a Macintosh-only, page-based, authoring tool), but it was later ported to Adobe's Director (a time-based tool) so that it could be played on both Mac and Windows platforms (see Chapter 7 for details about authoring systems).

Storyboarding a Project

The source material (all that was available) practically sorted itself into logical groups: a pile of old photographs, a magazine article and newspaper clippings, engineering drawings, official documents, and some recorded sounds. The first storyboard was a simple hierarchical structure with branches to each subject area, as shown in Figure 10-10.

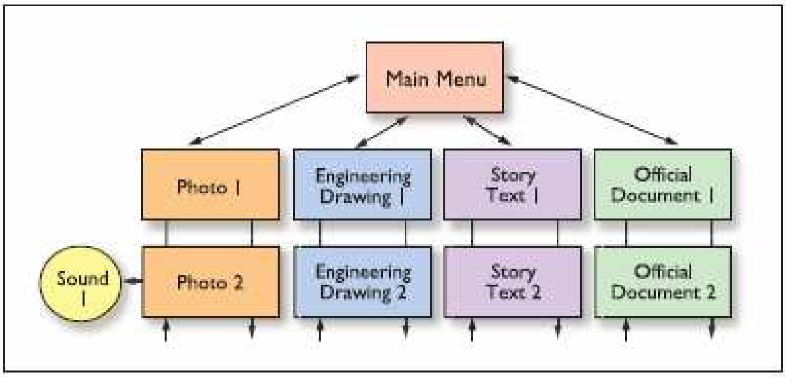


Figure 10-10 The first storyboard

Putting It Together

The most eye-catching photograph was chosen as a background for the main menu, and, as shown in Figure 10-11, the main menu was planned to contain clearly labeled buttons navigating to linear presentations of each topic area. From every screen in the project, users would be able to return to the main menu. Where sound bites were appropriate, clicking buttons on screens would play sounds. Adding a Quit button was necessary, also on the main menu, so that users would never be more than two button clicks from exiting the project (back to the main menu, and then quit).

The 50 or so 4 × 5 photographs were old color prints that offered poor contrast and faded colors (due to a saltwater dunking in a storm off the Central American coastline). Digitized on a flatbed scanner in gray-scale, however, they worked fine, and Photoshop was available to improve contrast. All the prints were scanned, cropped to the same dimension, optimized, and stored as bitmapped objects within Director. While at the scanner, merchant marine licenses and documents were also digitized, and the magazine article was scanned using OCR software to bring it into ASCII format. The story text was imported into the project.

After all the content was in Director's Cast (see Chapter 11), and work on the navigation system was under way, several issues emerged. First, it is terrifically boring to read a 3,000-word story by scrolling a long text field. Second, the photos were too small to be placed alone on a single screen. So it made sense to combine the story line with the images, even though they were not directly related; the story about launching the boat would progress from beginning to end as the boat was slowly built in the pictures. The storyboard changed to that shown in Figure 10-12.

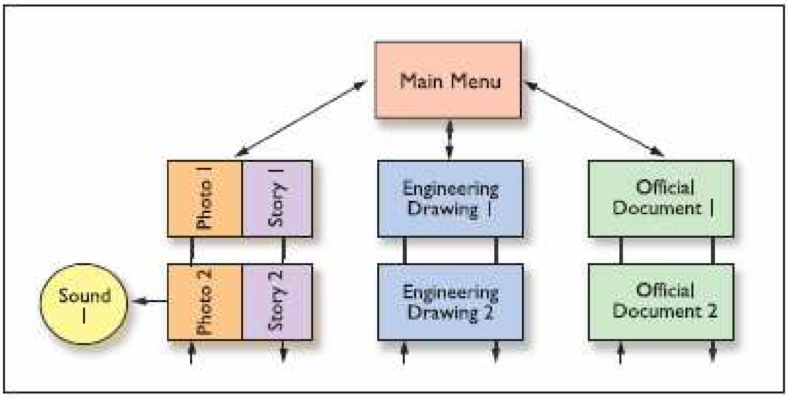


Figure 10-12 The second storyboard

The photo-essay-and-story combination worked out to 28 screens. The photos were placed into the score, and the text was cut and pasted into fields (see Figure 10-13). It became clear that users might want to scan rapidly through the photographs to watch the boat being built, ignoring the text of the story. So a special button was programmed to scan through the images until the mouse was clicked.

Images that did not fit into the photo essay about building the boat—for instance, the launching party with its roast pig, the long haul to the beach by trailer, and setting the mooring—were withdrawn from the pile of construction photographs; but because they were interesting, they were attached as separate branches accessible by button from the main menu. This was the third time the navigation changed, proving that you can continue to hang elements on a menu until the menu screen is too busy (and then you use submenus), or until you run out of material, as shown in Figure 10-14.

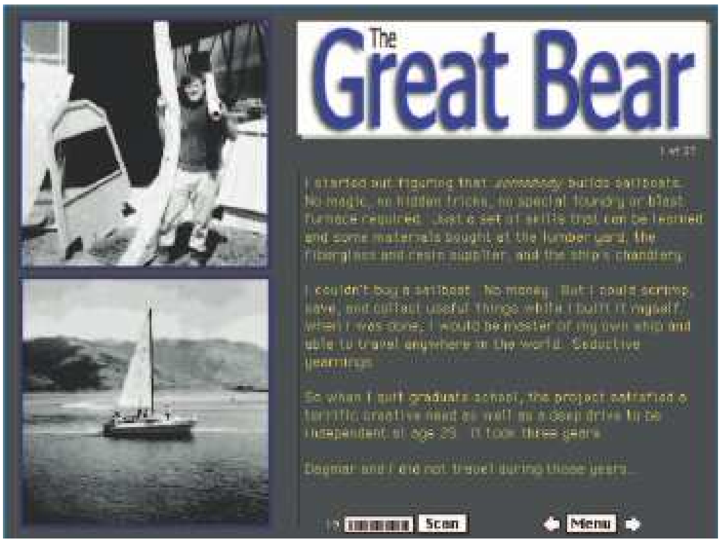


Figure 10-13 Snapshots are combined with text to form a 28-screen story line.

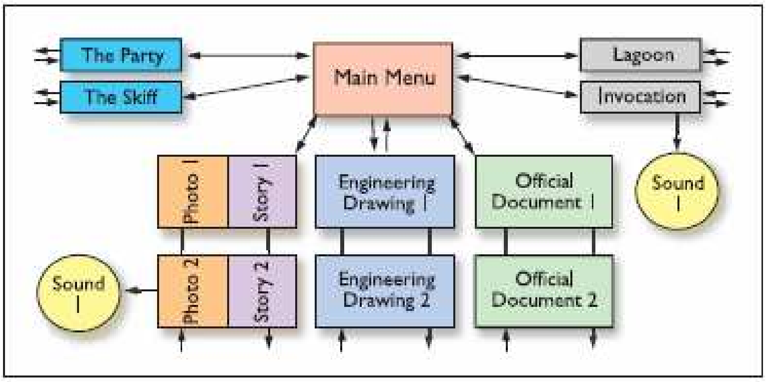


Figure 10-14 The third storyboard

Next, the sound bites were recorded, digitized, and added to the project. Figure 10-15, the screen where the sounds were to play, shows the special button installed to play sound bites. It's simply a picture of a loudspeaker. The documents for the project included engineering drawings, highway permits, and licenses.

The highway permit, for example, was 8.5 by 11 inches (portrait); but after some experimentation, once it was scaled to 480 pixels in height, it was (barely) readable and acceptable for this project. The licenses and drawings were in landscape orientation and fit more easily on a 640 × 480-pixel screen (see Figure 10-16).

Figure 10-15 Sound is played when the loudspeaker icon is clicked.

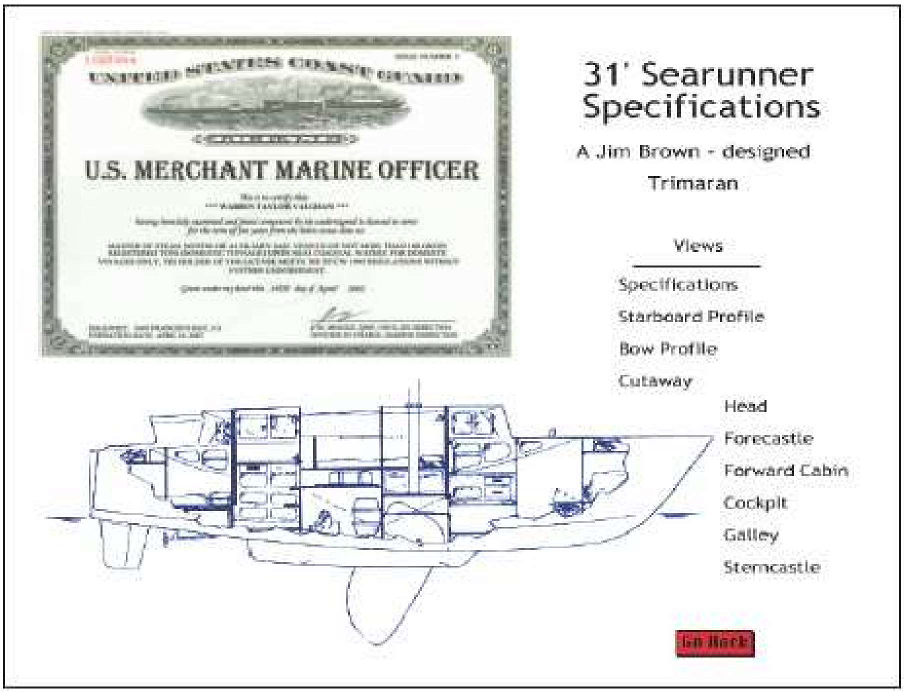


Figure 10-16 Larger fonts of some scanned documents can be read at 72 dpi resolution, and engineering drawings in landscape orientation can be resized to fit.

The buttons on the main menu were the wrong color, so they were changed a few times until the color worked. Helvetica title text wasn't fancy enough, so it was reworked and a drop shadow was laid in. A special slider button was built and programmed to allow the construction sequence to go immediately to any of the pictures in the sequence. The backgrounds were tweaked a little, and the order of images changed somewhat. A small red car was animated to drive along the edge of the lagoon.

The project described here was simple and straightforward. With the exception of designing a few custom buttons for auto-scanning through some of the images and designing the animations, the entire project was a progression of screens of information, with links activated by clicking buttons.

Testing

The terms alpha and beta are used by software developers to describe levels of product development when testing is done and feedback is sought. **Alpha releases** are typically for internal circulation only and are passed among a select group of mock users—often just the team working on the project. These versions of a product are often the first working drafts of your project, and you can expect them to have problems or to be incomplete**.** **Beta releases**, on the other hand, are sent to a wider but still select audience with the same caveat: this software may contain errors, bugs, and unknown alligators that slither out of the swamp at day's end to bite startled designers from behind. Because your product is now being shown and used outside the privacy of its birth nest, its reputation will begin to take form during beta phase. Thankfully though, beta-level bugs are typically less virulent than alpha bugs.

Alpha Testing

You should remain flexible and amenable to changes in both the design and the behavior of your project as you review the comments of your alpha testers. Beware of alpha testing groups made up of kindly friends who can provide positive criticism. Rather, you need to include aggressive people who will attack all aspects of your work. The meaner and nastier they are, the more likely they will sweat out errors or uncertainties in your product's design or navigation system. In the testing arena, learn to skillfully utilize friend and enemy alike. You will undoubtedly discover aspects of your work that, despite even the most insightful planning, you have overlooked.

Beta Testing

The beta testing group should be representative of real users and should not include persons who have been involved in the project's production. Beta testers must have no preconceived ideas. You want them to provide commentary and reports in exchange for getting to play with the latest software and for recognition as part of this "inside" process. Managing beta test feedback is critical. If you ignore or overlook testers' comments, the testing effort is a waste.

Ask your beta testers to include a detailed description of the hardware and software configuration at the time the problem occurred, and a step-by-step recounting of the problem, so that you can recreate it, analyze it, and repair it. You should also solicit general comments and suggestions. Figure 14-1 presents the search page from a web-based bug reporting system that is database driven and capable of managing thousands of reports about a complicated application in a meaningful way. (Apache is the most widely used HTTP daemon for
Serving web pages.)

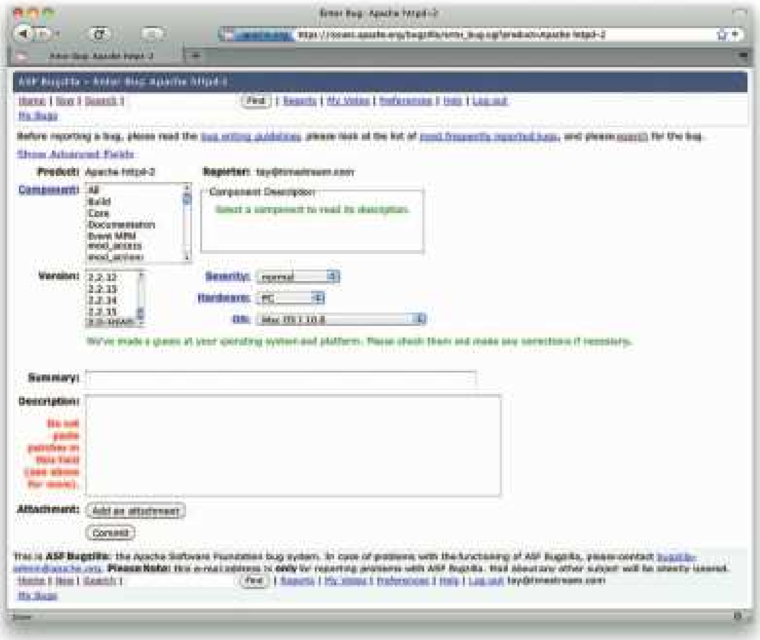


Figure 14-1
This page from a web-based bug reporting system seeks precise and reproducible descriptions of problems. In testing complex applications, thousands of bug reports may be received, and a dedicated quality control team may be tasked to deal with them.

Polishing to Gold

As you move through alpha and beta testing, and then through the debugging process toward a final release, you may want to use terms that indicate the current version status of your project. For example, bronze when you are close to being finished, gold when you have determined there is nothing left to change or correct and are ready to reproduce copies from your golden master. Some software developers also use the term release candidate (with a version number) as they continue to refine the product and approach a golden master. Going gold, or announcing that the job is finished, and then shipping, can be a scary thing. Indeed, if you examine the file creation time and date for many software programs, you will discover that many went gold at two o'clock in the morning.

Preparing for Delivery

If your completed multimedia project will be delivered to consumers or to a client who will install the project on many computers, you will need to prepare your files so they can be easily transferred from your media to the user's platform. Simply copying a project's files to the user's hard disk is often not enough for proper installation; frequently, you will also need to install special system and run-time files. So that end users can easily and automatically set up your project or application on their own computers, you may need to provide a single program that acts as an installer.

It is important to provide well-written documentation about the installation process so that users have a clear step-by-step procedure to follow. That documentation must include a discussion of potential problems and constraints related to the full range of your target platforms. Because you likely will not have control over the specification and configuration of the user's platform, it is critical that you include appropriate warnings in your installation document, like these examples:

- Must have at least 1GB of RAM
- Will not run unless QuickTime is installed
- 3MB available disk drive space
- Disable all screen savers before running
- Back up older versions before installing this update Often a file named README.TXT or Read.Me is a useful thing to include on the distribution disc of your project. This file can be a simple ASCII text file accessible by any text editor or word processing application. It should contain a description of changes or bugs reported since the documentation was printed and may also contain a detailed description of the installation process. If your project will be deployed on the Web, provide special directions in a FAQ or Help page, where you can describe required plug-ins and browser compatibility and other issues. You can often do much dynamically using JavaScript "sniffer" routines to alert users when their browser is not compatible, and you can even serve a different version of your project tailored to that user's system or browser.

The clearer and more detailed your installation instructions are, the fewer frustrated queries you will receive from your project's users. If your project is designed for wide distribution, installation problems can cause you many headaches and a great deal of time and expense in providing answers and service over the telephone. Set up a product-related web site with pages for software registration, bug reporting, technical support, and program upgrades.

From: Christopher Yavelow <Christopher@yav.com>
Subject: The case of the keyboardless kiosk
About ten days ago I posted an announcement to the list
about our interactive kiosk installation at the new
Netherlands Museum of Science and Technology.
Now I've discovered that science museums at this level are
the target of bands of teenage hackers that try to crash
all the exhibits. Our exhibit fell prey to such a band last Friday.
Although the software is running inside of a kiosk built
into a larger "The Music is the Message" exhibit housing
AND the museum visitor has only a trackball and single push
button to operate the exhibit AND there is *no* way to quit
the software without issuing a command-Q from a keyboard
which is double-locked inside the guts of the exhibit
housing, some kids were able to get back to the desktop and
delete the 60 MB of files associated with the exhibit...
and they did so in such a way that Norton Utils (3.5) could
not find them for un-erasing (I had to bring over a CD-ROM
version and re-install the entire exhibit).
How did they do it? Is there a way to get back to the
desktop in such a scenario: a (for all practical purposes)
keyboardless kiosk with only a trackball and single button
interface and no on-screen option to quit the application?
There are no menus, the menubar is hidden, there are no quit
buttons, AllowInterrupts is set to false, etc., etc.
Christopher Yavelow
YAV Interactive Media
Brederodestraat 47
2042 BB Zandvoort

The Netherlands
eMail: Christopher@yav.com
wSite: http://www.yav.com

Scary message found at an Internet newsgroup for multimedia programmers

Delivering on CD-ROM

Many multimedia projects are delivered on CD-ROM or DVD. While the very first users of CD-ROMs were owners of large databases like library catalogs, reference systems, and parts lists, today most computers are shipped with a CD/DVD drive, and software that is not downloaded from the Internet is typically packaged on a disc.

Compact Disc Technology

A compact disc, or CD, is a thin wafer of clear polycarbonate plastic and metal measuring 4.75 inches (120 mm) in diameter, with a small hole, or hub, in its center. The metal layer is usually pure aluminum, sputtered onto the polycarbonate surface in a thickness measurable in molecules. As the disc spins in the CD player, the metal reflects light from a tiny infrared laser into a light-sensitive receiver diode. These reflections are transformed into an electrical signal and then further converted to meaningful bits and bytes for use in digital equipment.

Pits on the CD, where the information is stored, are 1 to 3 microns long, about 1/2 micron wide, and 1/10 micron deep. (By comparison, a human hair is about 18 microns in diameter.) A CD can contain as many as three miles of these tiny pits wound in a spiral pattern from the hub to the edge. A layer of lacquer is applied to protect the surface, and artwork from the disc's author or publisher is usually silk-screened on the back side.

Compact discs are made in what is generally referred to as a family process. The glass master is made using the well-developed, photolithographic techniques created by the microchip industry: First an optically ground glass disc is coated with a layer of photo-resistant material 1/10 micron thick. A laser then exposes (writes) a pattern of pits onto the surface of the chemical layer of material. The disc is developed (the exposed areas are washed away) and is silvered, resulting in the actual pit structure of the finished master disc. The master is then electroplated with layers of nickel one molecule thick, one layer at a time, until the desired thickness is reached.

The nickel layer is separated from the glass disc and forms a metal negative, or father. In cases where low runs of just a few discs are required, the father is used to make the actual discs. Most projects, though, require several mothers, or positives, to be mad by plating the surface of the father. In a third plating stage, sons, or stampers, are made from the mother, and these are the parts that are used in the injection molding machines. Plastic pellets are heated and injected into the mold or stamper, forming the disc with the pits in it.

The plastic disc is coated with a thin aluminu layer for reflectance and lacquer for protection, given a silk-screened label for marketing, and packaged for delivery. Most of these activities occur in a particle-free clean room, because one speck of dust larger than a pit can ruin many hours of work. The mastering process alone takes around 12 hours.

CD-R

CD-R (compact disc-recordable) is an excellent method for distributing multimedia projects. CD-R writers and blank CD-R discs are inexpensive, and for short runs of a product, it is more cost effective to burn your work onto CD-Rs and custom-label them with your own printer than to have the discs mastered and pressed using the expensive father and son method described previously. Many services with auto-loading equipment and 24-hour turnarounds can make short runs.

CD-R blanks that can hold as much as 84 minutes of Red Book sound (see the next section) or more than 700MB of data are made of a polycarbonate core coated with layers of reflective metals and special photosensitive organic dyes (see Figure 14-3). During the burning process, laser light hits the layer of dye, bakes it, and forms a pit. A 74-minute CD-R disc contains 333,000 sectors * 2048 bytes / sector for a capacity of 650.4MB. An 80-minute disc contains 360,000 sectors * 2048 bytes / sector for a data capacity of 703.1MB.

Principles for successful project management of multimedia productions

1. Production is the phase when your multimedia project is actually rendered.
2. Provide a time-accounting system for everyone working on the project.
3. Check your development hardware and software and review your organizational and administrative setup.
4. Have a system for communication between you, the client, and the people actually building the project in place.
5. Provide management oversight and control the client review process to avoid endless feedback loops.
6. Establish a process in which your client is continually informed and formally approves the project as you develop it.
7. Organize a method for tracking the receipt of material that you will incorporate into your multimedia project.
8. Develop a file-naming conventions specific to your project’s structure.
9. Version control of your file (tracking editing changes s is critically important, especially in large projects.

The Process of Making Multimedia

Usually something will click in your mind or in the mind of a client that says, “Hey, wouldn’t it be neat if we could…” Your visions of sound and music, flashy images, and perhaps a video will solve a business need, provide an attention-grabbing product demo, or yield a slick front end to an otherwise drab computer database. You might want to spark a little interest or a laugh in an otherwise dull meeting, build an interactive photo album for Christmas greetings to your family, or post your company’s annual report in a new set of pages on the Web.

Plan for the entire process: beginning with your first ideas and ending with completion and delivery of a finished product. Think in the overview. The stepwise process of making multimedia is illustrated in Figure 9-1. Use this chart to help you get your arms around a new web site or DVD production! Note the feedback loops for revisions based upon testing and experiment. Note also the constant presence of an “evaluation committee” (who could be simply a project manager) to oversee the whole.

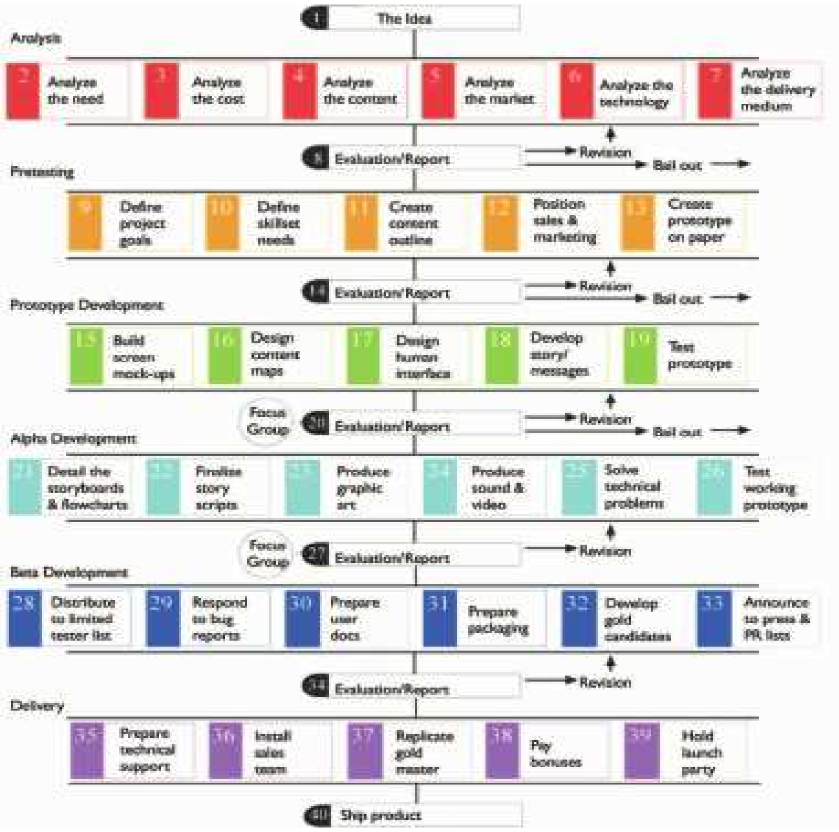


Figure 9-1 The process of making multimedia

It is, of course, easiest to plan a project using the experience you have accumulated in similar past projects. Over time, you can maintain and improve your multimedia-planning format, just like a batch of sourdough starter. Just keep adding a little rye and water every time you do a project, and the starter for your next job gets a bit more potent as your estimates become tempered by experience.

Idea Analysis

The important thing to keep in mind when you are toying with an idea is balance. As you think through your idea, you must continually weigh your purpose or goal against the feasibility and cost of production and delivery. Use whiteboard, notepaper, and scratch pads as you flesh out your idea, or use a note-taking or outlining program on your computer. Start with broad brushstrokes, and then think through each constituent multimedia element. Ultimately, you will generate a plan of action that will become your road map for production. Who needs this project? Is it worthwhile?

Do you have the materials at hand to build it? Do you have the skills to build it? Your idea will be in balance if you have considered and weighed the proper elements:

- What is the essence of what you want to do? What is your purpose and message?
- Who is your intended audience? Who will be your end users? What do they already know about the subject? Will they understand industry terms (jargon), and what information do they need your project to communicate to them? What will their multimedia playback platforms be, and what are the minimal technical capabilities of those platforms?
- Is there a client, and what does the client want?
- How can you organize your project?
- What multimedia elements (text, sounds, and visuals) will best deliver your message?
- Do you already have content material with which you can leverage your project, such as old videotapes or video files, music, documents, photographs, logos, advertisements, marketing packages, and other artwork?
- Will interactivity be required?
- Is your idea derived from an existing theme that can be enhanced with multimedia, or will you create something totally new?
- What hardware is available for development of your project? Is it enough?
- How much storage space do you have? How much do you need?
- What multimedia software is available to you?
- What are your capabilities and skills with both the software and the hardware?
- Can you do it alone? Who can help you?
- How much time do you have?
- How much money do you have?
- How will you distribute the final project?
- Will you need to update and/or support the final product?

You can maintain balance between purpose and feasibility by dynamically adding and subtracting multimedia elements as you stretch and shape your idea. You can start small and build from minimum capabilities toward a satisfactory result in an additive way. Or you can shoot the moon with a heavy list of features and desired multimedia results, and then discard items one by one because they are just not possible. Both additive and subtractive processes can work in concert and can yield very useful cost estimates and a production road map. Consider the following scenario: You have a video clip with four headand- shoulders testimonials that will be perfect for illustrating your message.

So add motion video to your list. You will need to purchase digitizing software, so add that item and its cost to your list as well. But you want to make your product available at a web site frequented by rural students without high-speed connections who will wait minutes for the video to play. Subtract motion video, but add tiny framed still images of the four talking heads (captured with your new video software) using short, one-sentence voice-overs of the speakers (recorded from the video clip).

Subtract one of the four testimonials because you discover that particular executive is no longer with the firm and you don’t have a signed release. Add animation instead. Subtract. Add. Subtract. In this manner, you will flesh out your idea, adding and subtracting elements within the constraints of the hardware, software, and your budget of cost and expertise.
The time you spend defining your project in this way—reality-testing it against technology and your abilities—might be your most valuable investment, even before you boot up a computer. At any point, you can decide to go forward or bail out.

Idea Management Software

Software such as dotProject, kForge, OpenProj, GanttProject (see Figure 9-2), outlining programs, and spreadsheets such as Excel can be useful for arranging your ideas and the many tasks, work items, employee resources, and costs required of your multimedia project. Project management tools provide the added benefit of built-in analysis to help you stay within your schedule and budget during the rendering of the project itself.

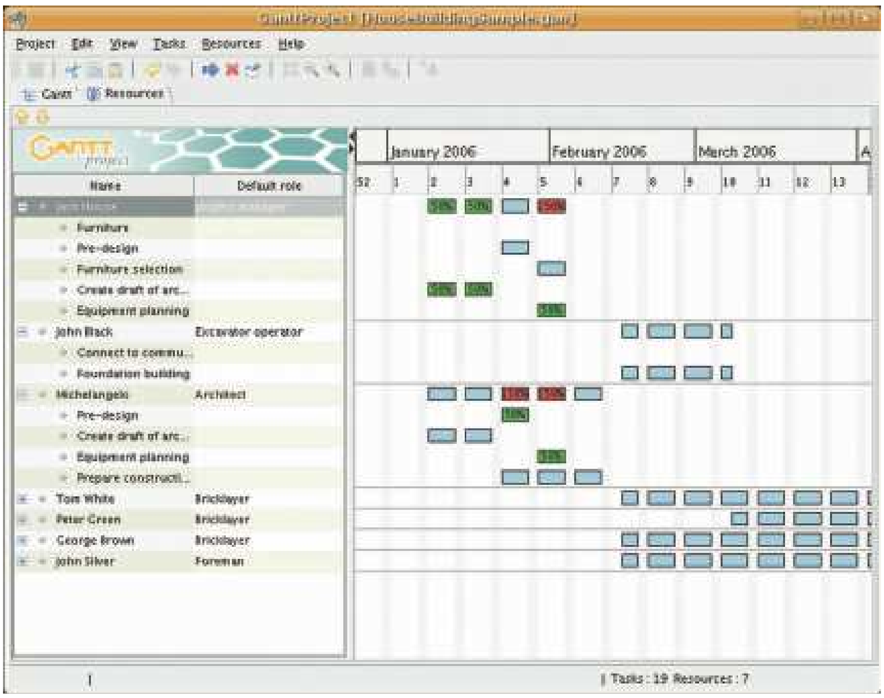


Figure 9-2 GanttProject, an open-source, web-based integrated project scheduling and management tool, generates helpful documents for instructional designers.

Project management software typically provides Critical Path Method (CPM) scheduling functions to calculate the total duration of a project based upon each identified task, earmarking tasks that are critical and that, if lengthened, will result in a delay in project completion. Program Evaluation Review Technique (PERT) charts provide graphic representations of task relationships, showing prerequisites, the tasks that must be completed before others can commence. Gantt charts depict all the tasks along a timeline.

Pre testing

If you decide that your idea has merit, take it to the next step. Define your project goals in greater detail and spell out what it will take in terms of skills, content, and money to meet these goals. If you envision a commercial product, sketch out how you will sell it. Work up a prototype of the project on paper, with an explanation of how it will work. All of these steps help you organize your idea and test it against the real world.

Task Planning

There may be many tasks in your multimedia project. Here is a checklist of action items for which you should plan ahead as you think through your project: In a white paper about producing educational software, elearnity (www.elearnity.com) has allocated percentages of effort, as shown to below:

Task	Percentage of Effort
Analyze need	3%
Draft mission statement	1%
Create audience profile	2%
Write objectives	2%
Analyze and outline content	6%
Lay out course map	2%
Define treatment	2%
Select learner activities	2%
Storyboard the course	19%
Author the course	28%
Evaluate the course	20%
Produce media	13%

Building a Team

Multimedia is an emerging technology requiring a set of skills so broad that multimedia itself remains poorly defined. Players in this technology come from all corners of the computer and art worlds as well as from a variety of other disciplines, so if you need to assemble a team, you need to know the people and skills it takes to make multimedia. (Refer to Chapter 8 for a description of the various skills and talents needed and how others have built successful teams.) Building a matrix chart of required skills is often helpful to describe the makeup of your team. The skills and software capabilities available to you are not as limiting as your list of required hardware–you can always budget for new and more powerful software and for the learning curve (or consultant fees) required to make use of it. Indeed, authoring software is usually necessary only for development of the project, not its playback or delivery, and should be a cost or learning burden not directly passed to end users.

Figure 9-4 shows a skill matrix developed when four medium-sized multimedia development companies came together to bid on a single, large CD-ROM project. If you are building a complex web site, substitute Java/Ruby programmer, HTML/CSS programmer, and Server Specialist into the proper row. Staying at the leading edge is important.

If you remain knowledgeable about what's new and expected, you will be more valuable to your own endeavors, to your team, and to your employer or prospective clients. But be prepared for steep learning curves and difficult challenges in keeping your own skills (and those of your employees) current and in demand. And don't neglect team morale as hours grow long, deadlines slip, and tempers flare.

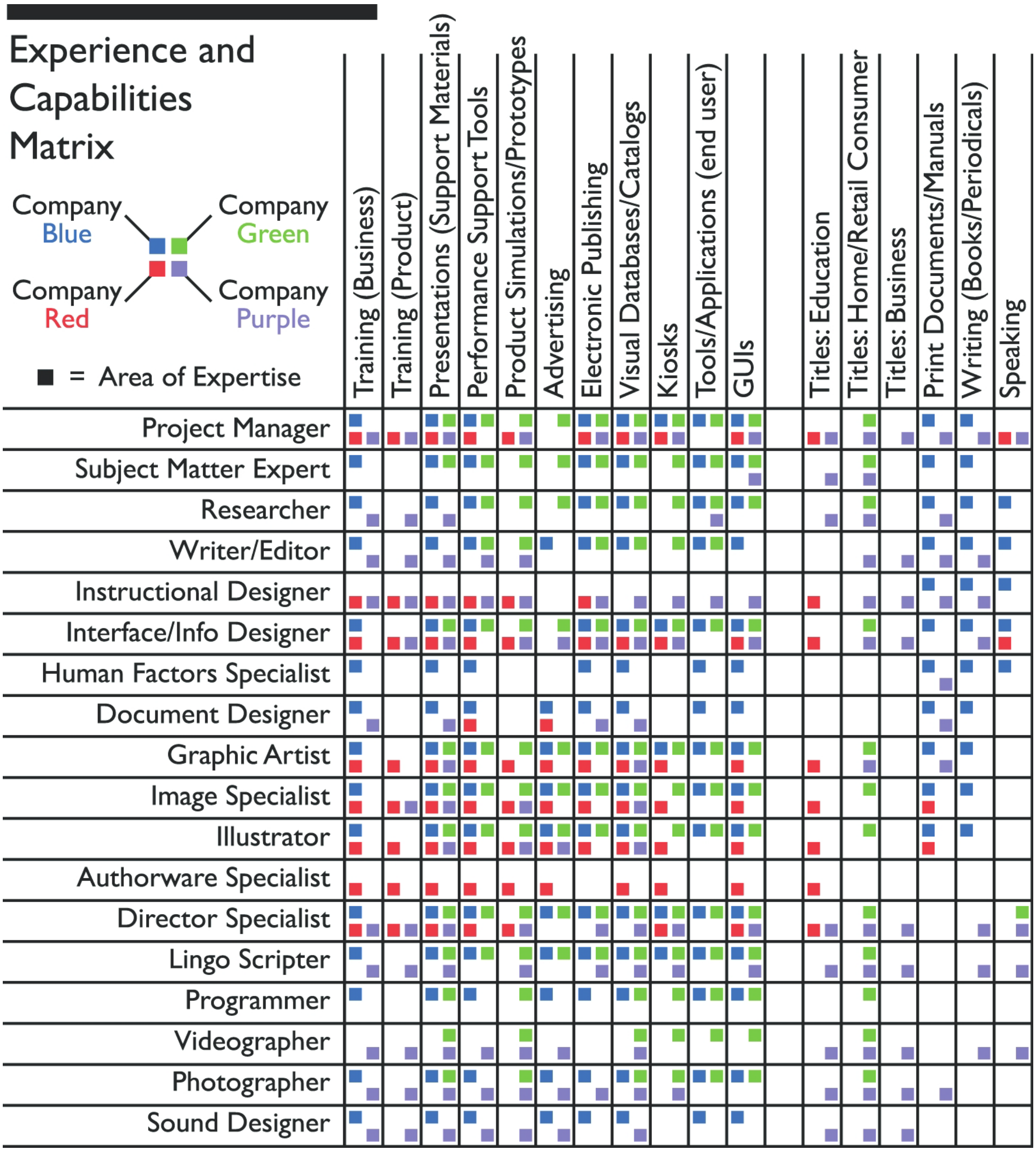


Figure 9-4 A matrix of available skills can assist you in planning for your project.

Prototype Development

Once you have decided that a project is worth doing, you should develop a working prototype. This is the point at which you begin serious work at the computer, building screen mock-ups and a human interface of menus and button clicks. Your messages and story lines will take shape as you explore ways of presenting them. For the prototype, sometimes called a proof-of-concept or feasibility study, you might select only a small portion of a large project and get that part working as it would in the final product. Indeed, after trying many different approaches in the course of prototyping, you may end up with more than one viable candidate for the final product. During this phase you can test ideas, mock up interfaces, exercise the hardware platform, and develop a sense about where the alligators live.

These alligators are typically found in the swampy edges of your own expertise; in the dark recesses of software platforms that almost-butnot- quite perform as advertised and in your misjudgment of the effort required for various tasks. The alligators will appear unexpectedly behind you and nip at your knees, unless you explore the terrain a little before you start out. Test your prototype along several fronts: technology (will it work on your proposed delivery platform or platforms?), cost (can you do this project within budget constraints?), market (can you sell it, or will it be properly used if it is an in-house project?), and human interface (is it intuitive and easy to use?). At this point you may wish to arrange a focus group, where you can watch potential end users experiment with your prototype and analyze their reactions.

The purpose of any prototype is to test the initial implementation of your idea and improve on it based upon test results. So you should never feel committed or bound to any one option, and you should be ready and willing to change things! Persuade the client to spend a small amount of money and effort up front to let you build a skeletal version of the project, including some artwork, interactive navigation, and performance checks. Indeed, there may be some very specific technology issues that need thorough examination and proof before you can provide a realistic estimate of the work and cost required. The focused experience of this proof will allow both you and the client to assess the project's goals and the means to achieve them. Include your experimental pilot as the first phase of your project.

At the pilot's conclusion, prepare a milestone report and a functional demo. You will be paid for the work so far, and the client will get real demonstration material that can be shown to bosses and managers. If your demo is good, it will be a persuasive argument within the client's management hierarchy for completing the full-scale project. Figure 9-5 is excerpted from trial calculations that were the result of a prototype five-language CD-ROM project. In the prototyping, office staff read the voice-over script as a "scratch track," like using a stand-in for the real thing; later, professional talent was used in the recording studio. As a result of building a prototype, accurate estimates of required storage space on the disc were possible.

Calculation Sheet

CD-ROM Project

Allocation of Disc Space

Note 1: The following trial calculations are based upon the file sizes yielded by an early voice rendering of the project's English script.

Note 2: File sizes for low-resolution images (72dpi) of 640×480 and 768×512 pixel dimensions are estimated at 768KB each.

Note 3: File sizes for high-resolution images (300dpi) may range from 3.7MB to 4.5MB, depending upon image complexity and compression rates. The conservative figure of 4.5MB per high-resolution image is used in these estimates.

Note 4: More accurate real estate estimates will be available following finalization of the script and recording of the English version narration.

Note 5: Firm count of low-resolution images and their pixel dimensions will be calculated upon script freeze.

SUMMARY: There is adequate room on the disc for both sound and images if each language recording is limited to no more than 9 minutes.

Scratch Track File

(English)	Duration	(English)	Duration
SNDE01A	18.369	SNDE10A	5.658
SNDE01B	9.180	SNDE11A	23.856
SNDE01C	9.295	SNDE12A	14.314
SNDE02A	17.609	SNDE13A	14.193
SNDE03A	17.932	SNDE14A	7.487
SNDE04A	11.156	SNDE15A	16.172
SNDE05A	18.035	SNDE16A	19.450
SNDE06A	8.050	SNDE17A	5.830
SNDE07A	12.790	SNDE18A	21.443
SNDE08A	16.218	SNDE19A	12.295
SNDE09A	27.468	Total	306.800 Seconds
			5.113 Minutes
		plus Intro Fanfare (Shared by all languages)	30.0 Seconds

Figure 9-5 Trial calculations are possible after prototyping.

As part of your delivery at the end of the pilot phase, reassess your estimates of the tasks required as well as the cost. Prepare a written report and analysis of budgets and anticipated additional costs. This is also the proper time to develop a revised and detailed project plan for the client. It allows the client some flexibility and provides a reality check for you. At this point you can also finalize your budget and payment schedule for the continuation of the project, as well as ink a contract and determine overrun procedures.

Difficulties may arise if your client is disappointed in the quantity of material delivered or is otherwise not satisfied with your work. If you have kept good records of the time and effort spent during prototyping, you may be able to smooth the rough waters. Remember that developing multimedia is a “trying” experience—try this, try that, then try this again a bit differently—and the creative process soaks up a lot of hours and cost. Listen carefully to the client's reaction to your prototype, because many problems can be quickly fixed, and all constructive comments can certainly be woven into the next phase of development.

Alpha Development

As you go forward, you should continually define the tasks ahead, because just as if you were navigating a supertanker, you should be aware of the reefs and passages that will appear along your course and prepare for them. With an alpha stage prototype in hand and a commitment to proceed, the investment of effort will increase and, at the same time, become more focused. More people may become involved as you begin to flesh out the project as a whole.

Beta Development

By the time your idea reaches the beta stage of development, you will have committed serious time, energy, and money, and it is likely too late to bail out. You have gone past the point of no return and should see it through. But by now you have a project that is looking great! Most of the features are working, and you are distributing it to a wider arena of testers. In fact, you are on the downhill slope now, and your concern should be simply successfully steering the project to its well-defined goal.

Delivery

By the time you reach the delivery stage, you are going gold—producing the final product. Your worries slide toward the marketplace: how will your project be received by its intended audience? You must also deal with a great many practical details, such as who will answer the support hotline and run the live chat desk, or whether to co-locate a server or trust the current ISP to handle the predicted increased volume of hits.

7. Coding and Compression

Introduction to coding and compression techniques

Entropy encoding run length

Run-length encoding (RLE) is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could greatly increase the file size.

RLE may also be used to refer to an early graphics file format supported by CompuServe for compressing black and white images, but was widely supplanted by their later Graphics Interchange Format. RLE also refers to a little-used image format in Windows 3.x, with the extension role, which is a Run Length Encoded Bitmap, used to compress the Windows 3.x startup screen.

Example

For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with B representing a black pixel and W representing white:
WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW
If we apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, we get the following:
12W1B12W3B24W1B14W
This is to be interpreted as twelve Ws, one B, twelve Ws, three Bs, etc.
The run-length code represents the original 67 characters in only 18. Of course, the actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same. Even binary data files can be compressed with this method; file format specifications often dictate repeated bytes in files as padding space. However, newer compression methods such as DEFLATE often use LZ77-based algorithms, a generalization of run-length encoding that can take advantage of runs of strings of characters (such as BWWBWWBWWBWW).

Applications

Run-length encoding performs lossless data compression and is well suited to palette-based iconic images. It does not work well at all on continuous-tone images such as photographs, although JPEG uses it quite effectively on the coefficients that remain after transforming and quantizing image blocks.

Common formats for run-length encoded data include Truevision TGA, PackBits, PCX and ILBM.
Run-length encoding is used in fax machines (combined with other techniques into Modified Huffman coding). It is relatively efficient because most faxed documents are mostly white space, with occasional interruptions of black.

Huffman

Huffman coding is an entropy encoding algorithm used for lossless data compression. The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It was developed by David A. Huffman while he was a Ph.D. student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes".

Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code (sometimes called "prefix-free codes", that is, the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol) that expresses the most common source symbols using shorter strings of bits than are used for less common source symbols. Huffman was able to design the most efficient compression method of this type: no other mapping of individual source symbols to unique strings of bits will produce a smaller average output size when the actual symbol frequencies agree with those used to create the code. A method was later found to design a Huffman code in linear time if input probabilities (also known as weights) are sorted.[citation needed]

For a set of symbols with a uniform probability distribution and a number of members which is a power of two, Huffman coding is equivalent to simple binary block encoding, e.g., ASCII coding. Huffman coding is such a widespread method for creating prefix codes that the term "Huffman code" is widely used as a synonym for "prefix code" even when such a code is not produced by Huffman's algorithm.

Although Huffman's original algorithm is optimal for a symbol-by-symbol coding (i.e. a stream of unrelated symbols) with a known input probability distribution, it is not optimal when the symbol-by-symbol restriction is dropped, or when the probability mass functions are unknown, not identically distributed, or not independent (e.g., "cat" is more common than "cta"). Other methods such as arithmetic coding and LZW coding often have better compression capability: both of these methods can combine an arbitrary number of symbols for more efficient coding, and generally adapt to the actual input statistics, the latter of which is useful when input probabilities are not precisely known or vary significantly within the stream.

However, the limitations of Huffman coding should not be overstated; it can be used adaptively, accommodating unknown, changing, or context-dependent probabilities. In the case of known independent and identically distributed random variables, combining symbols reduces inefficiency in a way that approaches optimality as the number of symbols combined increases.

Huffman coding is based on the frequency of occurance of a data item (pixel in images). The principle is to use a lower number of bits to encode the data that occurs more frequently. Codes are stored in a ***Code Book*** which may be constructed for each image or a set of images. In all cases the code book plus encoded data must be transmitted to enable decoding.

The Huffman algorithm is now briefly summarized:

- A bottom-up approach

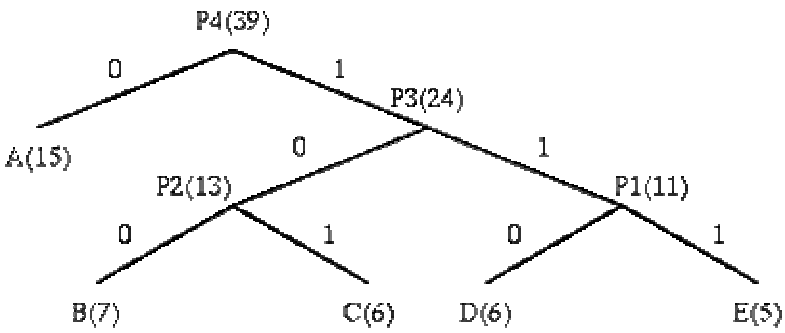
1. Initialization: Put all nodes in an OPEN list, keep it sorted at all times (e.g., ABCDE).

2. Repeat until the OPEN list has only one node left:

(a) From OPEN pick two nodes having the lowest frequencies/probabilities, create a parent node of them.

(b) Assign the sum of the children's frequencies/probabilities to the parent node and insert it into OPEN.

(c) Assign code 0, 1 to the two branches of the tree, and delete the children from OPEN.



Symbol	Count	log(1/p)	Code	Subtotal (# of bits)
A	15	1.38	0	15
B	7	2.48	100	21
C	6	2.70	101	18
D	6	2.70	110	18
E	5	2.96	111	15
		TOTAL (# of bits): 87		

The following points are worth noting about the above algorithm:

- Decoding for the above two algorithms is trivial as long as the coding table (the statistics) is sent before the data. (There is a bit overhead for sending this, negligible if the data file is big.)
- **Unique Prefix Property**: no code is a prefix to any other code (all symbols are at the leaf nodes) -> great for decoder, unambiguous.
- If prior statistics are available and accurate, then Huffman coding is very good.

In the above example:

Number of bits needed for Huffman Coding is: 87 / 39 = 2.23

JPEG compression process

One of the hottest topics in image compression technology today is JPEG. The acronym JPEG stands for the Joint Photographic Experts Group, a standards committee that had its origins within the International Standard Organization (ISO). In 1982, the ISO formed the Photographic Experts Group (PEG) to research methods of transmitting video, still images, and text over ISDN (Integrated Services Digital Network) lines. PEG's goal was to produce a set of industry standards for the transmission of graphics and image data over digital communications networks.

In 1986, a subgroup of the CCITT began to research methods of compressing color and gray-scale data for facsimile transmission. The compression methods needed for color facsimile systems were very similar to those being researched by PEG. It was therefore agreed that the two groups should combine their resources and work together toward a single standard.

In 1987, the ISO and CCITT combined their two groups into a joint committee that would research and produce a single standard of image data compression for both organizations to use. This new committee was JPEG.

Although the creators of JPEG might have envisioned a multitude of commercial applications for JPEG technology, a consumer public made hungry by the marketing promises of imaging and multimedia technology are benefiting greatly as well. Most previously developed compression methods do a relatively poor job of compressing continuous-tone image data; that is, images containing hundreds or thousands of colors taken from real-world subjects. And very few file formats can support 24-bit raster images.

GIF, for example, can store only images with a maximum pixel depth of eight bits, for a maximum of 256 colors. And its LZW compression algorithm does not work very well on typical scanned image data. The low-level noise commonly found in such data defeats LZW's ability to recognize repeated patterns.

Both TIFF and BMP are capable of storing 24-bit data, but in their pre-JPEG versions are capable of using only encoding schemes (LZW and RLE, respectively) that do not compress this type of image data very well.

JPEG provides a compression method that is capable of compressing continuous-tone image data with a pixel depth of 6 to 24 bits with reasonable speed and efficiency. And although JPEG itself does not define a standard image file format, several have been invented or modified to fill the needs of JPEG data storage.

JPEG in Perspective

Unlike all of the other compression methods described so far in this chapter, JPEG is not a single algorithm. Instead, it may be thought of as a toolkit of image compression methods that may be altered to fit the needs of the user. JPEG may be adjusted to produce very small, compressed images that are of relatively poor quality in appearance but still suitable for many applications. Conversely, JPEG is capable of producing very high-quality compressed images that are still far smaller than the original uncompressed data.

JPEG is also different in that it is primarily a lossy method of compression. Most popular image format compression schemes, such as RLE, LZW, or the CCITT standards, are lossless compression methods. That is, they do not discard any data during the encoding process. An image compressed using a lossless method is guaranteed to be identical to the original image when uncompressed.

Lossy schemes, on the other hand, throw useless data away during encoding. This is, in fact, how lossy schemes manage to obtain superior compression ratios over most lossless schemes. JPEG was designed specifically to discard information that the human eye cannot easily see. Slight changes in color are not perceived well by the human eye, while slight changes in intensity (light and dark) are. Therefore JPEG's lossy encoding tends to be more frugal with the gray-scale part of an image and to be more frivolous with the color.

JPEG was designed to compress color or gray-scale continuous-tone images of real-world subjects: photographs, video stills, or any complex graphics that resemble natural subjects. Animations, ray tracing, line art, black-and-white documents, and typical vector graphics don't compress very well under JPEG and shouldn't be expected to. And, although JPEG is now used to provide motion video compression, the standard makes no special provision for such an application.

The fact that JPEG is lossy and works only on a select type of image data might make you ask, "Why bother to use it?" It depends upon your needs. JPEG is an excellent way to store 24-bit photographic images, such as those used in imaging and multimedia applications. JPEG 24-bit (16 million color) images are superior in appearance to 8-bit (256 color) images on a VGA display and are at their most spectacular when using 24-bit display hardware (which is now quite inexpensive).

The amount of compression achieved depends upon the content of the image data. A typical photographic-quality image may be compressed from 20:1 to 25:1 without experiencing any noticeable degradation in quality. Higher compression ratios will result in image files that differ noticeably from the original image but still have an overall good image quality. And achieving a 20:1 or better compression ratio in many cases not only saves disk space, but also reduces transmission time across data networks and phone lines.

An end user can "tune" the quality of a JPEG encoder using a parameter sometimes called a *quality setting* or a *Q factor*. Although different implementations have varying scales of Q factors, a range of 1 to 100 is typical. A factor of 1 produces the smallest, worst quality images; a factor of 100 produces the largest, best quality images. The optimal Q factor depends on the image content and is therefore different for every image. The art of JPEG compression is finding the lowest Q factor that produces an image that is visibly acceptable, and preferably as close to the original as possible.

The JPEG library supplied by the Independent JPEG Group uses a quality setting scale of 1 to 100. To find the optimal compression for an image using the JPEG library, follow these steps:

1. Encode the image using a quality setting of 75 (-Q 75).
2. If you observe unacceptable defects in the image, increase the value, and re-encode the image.
3. If the image quality is acceptable, decrease the setting until the image quality is barely acceptable. This will be the optimal quality setting for this image.
4. Repeat this process for every image you have (or just encode them all using a quality setting of 75).

JPEG isn't always an ideal compression solution. There are several reasons:

- As we have said, JPEG doesn't fit every compression need. Images containing large areas of a single color do not compress very well. In fact, JPEG will introduce "artifacts" into such images that are visible against a flat background, making them considerably worse in appearance than if you used a conventional lossless compression method. Images of a "busier" composition contain even worse artifacts, but they are considerably less noticeable against the image's more complex background.
- JPEG can be rather slow when it is implemented only in software. If fast decompression is required, a hardware-based JPEG solution is your best bet, unless you are willing to wait for a faster software-only solution to come along or buy a faster computer.
- JPEG is not trivial to implement. It is not likely you will be able to sit down and write your own JPEG encoder/decoder in a few evenings. We recommend that you obtain a third-party JPEG library, rather than writing your own.
- JPEG is not supported by very many file formats. The formats that do support JPEG are all fairly new and can be expected to be revised at frequent intervals.

Baseline JPEG

The JPEG specification defines a minimal subset of the standard called baseline JPEG, which all JPEG-aware applications are required to support. This baseline uses an encoding scheme based on the Discrete Cosine Transform (DCT) to achieve compression. DCT is a generic name for a class of operations identified and published some years ago. DCT-based algorithms have since made their way into various compression methods.

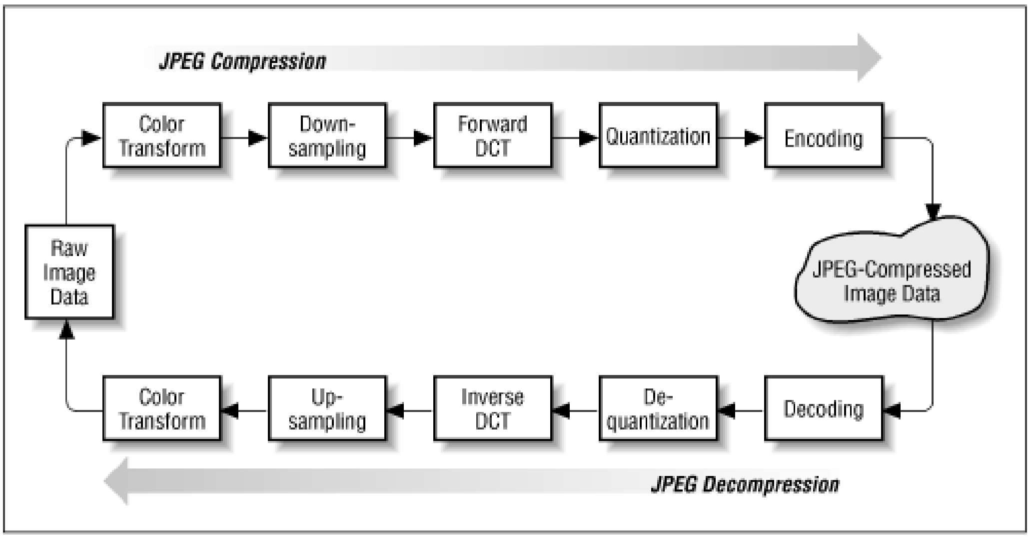
DCT-based encoding algorithms are always lossy by nature. DCT algorithms are capable of achieving a high degree of compression with only minimal loss of data. This scheme is effective only for compressing continuous-tone images in which the differences between adjacent pixels are usually small. In practice, JPEG works well only on images with depths of at least four or five bits per color channel. The baseline standard actually specifies eight bits per input sample. Data of lesser bit depth can be handled by scaling it up to eight bits per sample, but the results will be bad for low-bit-depth source data, because of the large jumps between adjacent pixel values. For similar reasons, colormapped source data does not work very well, especially if the image has been dithered.

The JPEG compression scheme is divided into the following stages:

1. Transform the image into an optimal color space.
2. Downsample chrominance components by averaging groups of pixels together.
3. Apply a Discrete Cosine Transform (DCT) to blocks of pixels, thus removing redundant image data.
4. Quantize each block of DCT coefficients using weighting functions optimized for the human eye.
5. Encode the resulting coefficients (image data) using a Huffman variable word-length algorithm to remove redundancies in the coefficients.

Figure 9-11 summarizes these steps, and the following subsections look at each of them in turn. Note that JPEG decoding performs the reverse of these steps.

Figure 9-11: JPEG compression and decompression



Transform the image

The JPEG algorithm is capable of encoding images that use any type of color space. JPEG itself encodes each component in a color model separately, and it is completely independent of any color-space model, such as RGB, HSI, or CMY. The best compression ratios result if a luminance/chrominance color space, such as YUV or YCbCr, is used.

Most of the visual information to which human eyes are most sensitive is found in the high-frequency, gray-scale, luminance component (Y) of the YCbCr color space. The other two chrominance components (Cb and Cr) contain high-frequency color information to which the human eye is less sensitive. Most of this information can therefore be discarded.

In comparison, the RGB, HSI, and CMY color models spread their useful visual image information evenly across each of their three color components, making the selective discarding of information very difficult. All three color components would need to be encoded at the highest quality, resulting in a poorer compression ratio. Gray-scale images do not have a color space as such and therefore do not require transforming.

Downsample chrominance components

The simplest way of exploiting the eye's lesser sensitivity to chrominance information is simply to use fewer pixels for the chrominance channels. For example, in an image nominally 1000x1000 pixels, we might use a full 1000x1000 luminance pixels but only 500x500 pixels for each chrominance component. In this representation, each chrominance pixel covers the same area as a 2x2 block of luminance pixels. We store a total of six pixel values for each 2x2 block (four luminance values, one each for the two chrominance channels), rather than the twelve values needed if each component is represented at full resolution. Remarkably, this 50 percent reduction in data volume has almost no effect on the perceived quality of most images. Equivalent savings are not possible with conventional color models such as RGB, because in RGB each color channel carries some luminance information and so any loss of resolution is quite visible.

When the uncompressed data is supplied in a conventional format (equal resolution for all channels), a JPEG compressor must reduce the resolution of the chrominance channels by*downsampling*, or averaging together groups of pixels. The JPEG standard allows several different choices for the sampling ratios, or relative sizes, of the downsampled channels. The luminance channel is always left at full resolution (1:1 sampling). Typically both chrominance channels are downsampled 2:1 horizontally and either 1:1 or 2:1 vertically, meaning that a chrominance pixel covers the same area as either a 2x1 or a 2x2 block of luminance pixels. JPEG refers to these downsampling processes as 2h1v and 2h2v sampling, respectively.

Another notation commonly used is 4:2:2 sampling for 2h1v and 4:2:0 sampling for 2h2v; this notation derives from television customs (color transformation and downsampling have been in use since the beginning of color TV transmission). 2h1v sampling is fairly common because it corresponds to National Television Standards Committee (NTSC) standard TV practice, but it offers less compression than 2h2v sampling, with hardly any gain in perceived quality.

Apply a Discrete Cosine Transform

The image data is divided up into 8x8 blocks of pixels. (From this point on, each color component is processed independently, so a "pixel" means a single value, even in a color image.) A DCT is applied to each 8x8 block. DCT converts the spatial image representation into a frequency map: the low-order or "DC" term represents the average value in the block, while successive higher-order ("AC") terms represent the strength of more and more rapid changes across the width or height of the block. The highest AC term represents the strength of a cosine wave alternating from maximum to minimum at adjacent pixels.

The DCT calculation is fairly complex; in fact, this is the most costly step in JPEG compression. The point of doing it is that we have now separated out the high- and low-frequency information present in the image. We can discard high-frequency data easily without losing low-frequency information. The DCT step itself is lossless except for roundoff errors.

Quantize each block

To discard an appropriate amount of information, the compressor divides each DCT output value by a "quantization coefficient" and rounds the result to an integer. The larger the quantization coefficient, the more data is lost, because the actual DCT value is represented less and less accurately. Each of the 64 positions of the DCT output block has its own quantization coefficient, with the higher-order terms being quantized more heavily than the low-order terms (that is, the higher-order terms have larger quantization coefficients). Furthermore, separate quantization tables are employed for luminance and chrominance data, with the chrominance data being quantized more heavily than the luminance data. This allows JPEG to exploit further the eye's differing sensitivity to luminance and chrominance.

It is this step that is controlled by the "quality" setting of most JPEG compressors. The compressor starts from a built-in table that is appropriate for a medium-quality setting and increases or decreases the value of each table entry in inverse proportion to the requested quality. The complete quantization tables actually used are recorded in the compressed file so that the decompressor will know how to (approximately) reconstruct the DCT coefficients.

Selection of an appropriate quantization table is something of a black art. Most existing compressors start from a sample table developed by the ISO JPEG committee. It is likely that future research will yield better tables that provide more compression for the same perceived image quality. Implementation of improved tables should not cause any compatibility problems, because decompressors merely read the tables from the compressed file; they don't care how the table was picked.

Encode the resulting coefficients

The resulting coefficients contain a significant amount of redundant data. Huffman compression will losslessly remove the redundancies, resulting in smaller JPEG data. An optional extension to the JPEG specification allows arithmetic encoding to be used instead of Huffman for an even greater compression ratio At this point, the JPEG data stream is ready to be transmitted across a communications channel or encapsulated inside an image file format.

MPEG audio and video compression

MPEG standards

The MPEG standards were developed by the Moving Picture Experts Group (MPEG, www.mpeg.org), a working group convened by the International Organization for Standardization (ISO) and the International Electro-technical Commission (IEC), which created standards for the digital representation of moving pictures as well as associated audio and other data.

Using MPEG-1 (specifications released in 1992), you could deliver 1.2 Mbps (megabits per second) of video and 250 Kbps (kilobits per second) of two-channel stereo audio using CD-ROM technology.

MPEG-2 (specifications released in 1994), a completely different system from MPEG-1, required higher data rates (3 to 15 Mbps) but also delivered higher image resolution, improved picture quality, interlaced video formats, multiresolution scalability, and multichannel audio features.

MPEG-2 became the video compression standard required for digital television (DTV) and for making DVDs.

The MPEG specifications since MPEG-2 include elements beyond just the encoding of video. As a container, MPEG-4 (specifications released in 1998 and 1999) provides a content-based method for assimilating multimedia elements. It offers indexing, hyperlinking, querying, browsing, uploading, downloading, and deleting functions, as well as “hybrid natural and synthetic data coding,” which will enable harmonious integration of natural and synthetic audiovisual objects. With MPEG-4, multiple views, layers, and multiple sound tracks of a scene, as well as stereoscopic and 3-D views, are available, making virtual reality workable.

MPEG-4 can adjust to varied download speeds, making it an attractive option for delivery of video on the Web. The MPEG-4 AVC standard (Advanced Video Coding, Part 10) requires the H.264 codec for Blu-ray discs. Because the software behind MPEG-4 is patented by more than two dozen companies, developers who build video editors and players that read and write MPEG-4 files must purchase licenses and make royalty payments.

CD formats

Name	Description	Comment
CD-Audio or CD-DA	Digital audio	Consumer audio discs
CD-ROM High Sierra	Read-only memory	Vestigial standard, seldom used
CD-ROM ISO 9660	Read-only memory	MS-DOS and Macintosh files
CD-ROM HFS	Read-only memory	Macintosh HFS files
CD-ROM/XA	Read-only memory	Extended Architecture
CD-I or CD-RTOS	Interactive	Philips Interactive motion video
CD-I Ready	Interactive/Ready	Audio CD with features for CD-I player
CD-Bridge	Bridge	Allows XA track to play on CD-I player
CD-MO	Magneto-optical	Premastered area readable on any CD player
CD-WO or CD-R	Write-once recordable	May use multiple sessions to fill disc
CD+G	Mixed mode	CD+Graphics—MTV on disc
CDTV	ISO 9660 variant	Commodore proprietary system
PhotoCD	Compressed images	Kodak multisession XA system
Video CD or Karaoke CD	Bridge	Karaoke full-motion MPEG video

Table 14-2 Compact Disc Formats

JPEG Images compression process,

JPEG (Joint Photographic Experts Group) images may contain 24 bits of color depth (millions of colors). JPEG uses a powerful but lossy compression method that produces files as much as ten times more compressed than GIF. Lossy means that information in the original image is lost in the compression process and cannot be retrieved. A lossless compression method does not irretrievably discard the original data.

The JPEG compression scheme compresses about 20:1 before visible image degradation occurs. Test the amount of compression acceptable for your JPEG image; stay inside the "threshold of visible error." To compress an image with JPEG, the image is divided into 8 × 8–pixel blocks, and the resulting 64 pixels (called a "search range") are mathematically described relative to the characteristic of the pixel in the upper-left corner.

The binary description of this relationship requires far less than 64 pixels, so more information can be squeezed into less space. JPEG compresses slowly, about one to three seconds for a 1MB image, depending upon computer speed, but JPEG can compress images as much as 75:1, with loss.