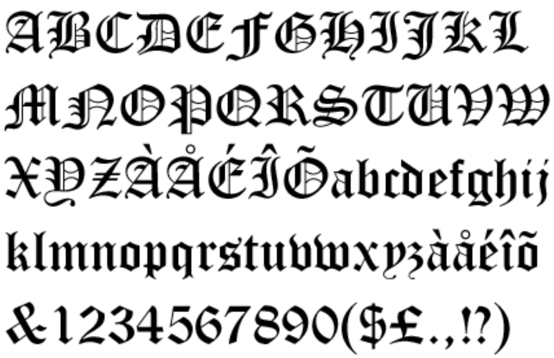
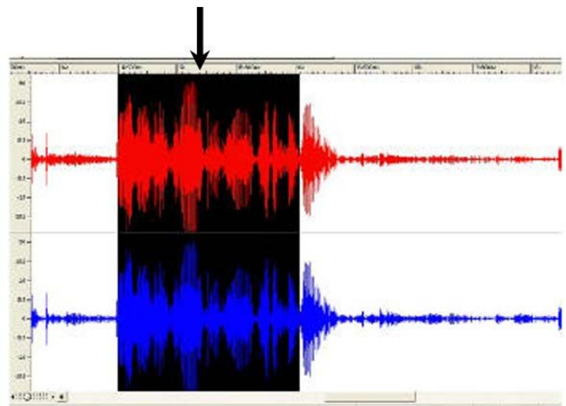
Introduction to Multimedia

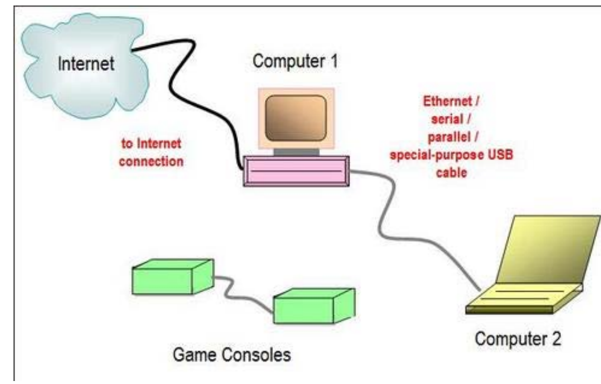
Introduction

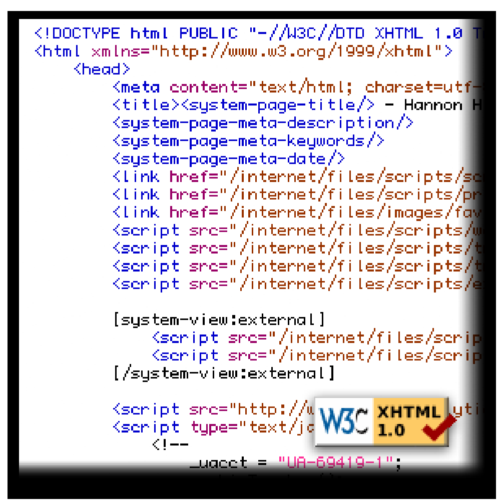
- **What is Multimedia?**
- Multimedia Software Tools
- Interactivity
- Design Issues
- Source Coding
- Legal Issues

What is Multimedia?





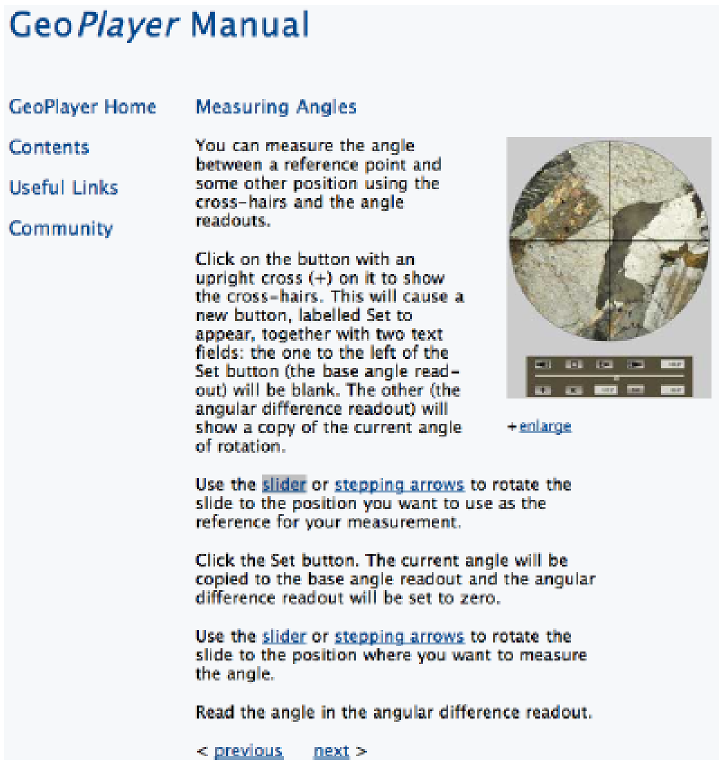




What is Multimedia?

- Media
  - the storage and transmission channels or tools used to store and deliver information or data.
  - http://en.wikipedia.org/wiki/Media_%28communication%29
  - Another definition: A set of coordinated channels spanning one or more modality which have come, by convention, to be referred to as a unitary whole, and which possess a cross-channel language of interpretation.
- Key concept:
  - Media is a carrier: storage or transmission channel.
  - This carrier is used to carry information.

What is Multimedia?


GeoPlayer Manual

GeoPlayer Home
Contents
Useful Links
Community

Measuring Angles

You can measure the angle between a reference point and some other position using the cross-hairs and the angle readouts.

Click on the button with an upright cross (+) on it to show the cross-hairs. This will cause a new button, labelled Set to appear, together with two text fields: the one to the left of the Set button (the base angle read-out) will be blank. The other (the angular difference readout) will show a copy of the current angle of rotation.

+ enlarge

Use the slider or stepping arrows to rotate the slide to the position you want to use as the reference for your measurement.

Click the Set button. The current angle will be copied to the base angle readout and the angular difference readout will be set to zero.

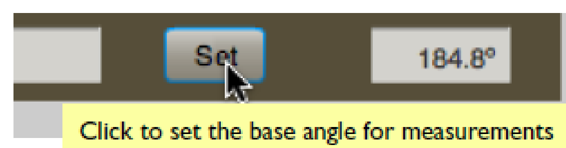
Use the slider or stepping arrows to rotate the slide to the position where you want to measure the angle.

Read the angle in the angular difference readout.

< previous next >

A Web page

Video



A tool tip

What is Multimedia?

- Time-based media: changes over time
  - Video, animation, sound …
- Static media: do NOT change over time
  - Still images, text
- Each medium has its own characteristics, leading to distinctive strengths and weakness
- Choose the most appropriate medium for your purpose

What is Multimedia?

- How about multimedia?
  - media and content that uses a combination of different content forms.
  - E.g. includes a combination of text, audio, still images, animation, video, and interactivity content forms.
  - http://en.wikipedia.org/wiki/Multimedia
- Key Components
  - Modality
  - Channel of communication
  - Media (see discussions above)

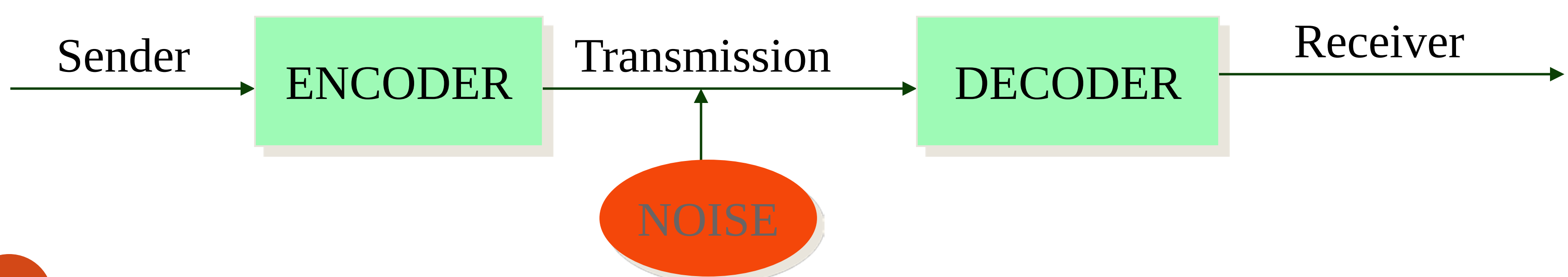
What is Multimedia?

- Modality: sensory systems available to human beings

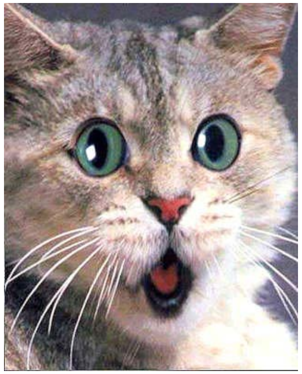
Modality	Sensory	Sense Organ
Tactile	Touching	Skin
Gustatory	Tasting	Tongue
Visual	Seeing	Eyes
Auditory	Hearing	Ears
Olfactory	Smelling	Nose

What is Multimedia?

- Channel of Communication
  - A connection between an encoder and a decoder such that information is encoded by the encoder, transmitted along the channel and decoded by the decoder to produce the same information at the other end of the channel.
  - There are various types of channels of communication.
  - A channel of communication exists within a single modality; A modality may include many channels of communication.



What is Multimedia?

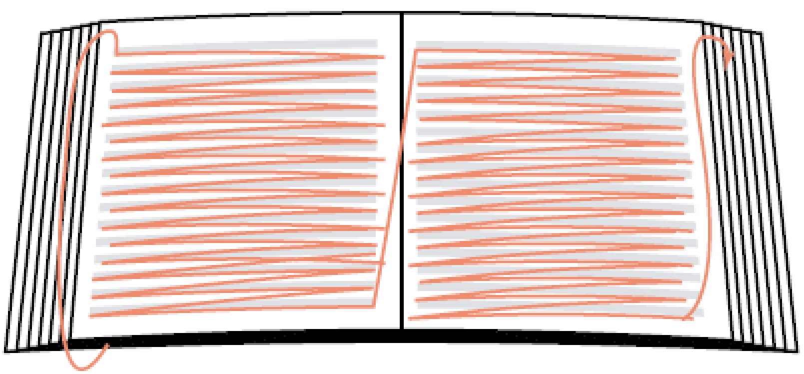


- What is digital multimedia?
  - "digitalized" multimedia
- An important relation between multimedia and computer.
  - Computer processes digitalized information
  - We only discuss digitalized multimedia, including how to digitalize
- Why digitalized?
  - Easy to represent, store, transmit, process…
  - Can you think more?

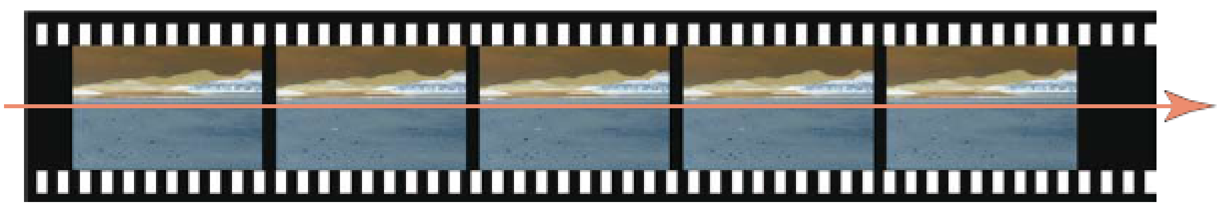
What is Multimedia?

- Linear Structure in conventional media

Book: physical arrangement of text and pages implies a linear reading order.

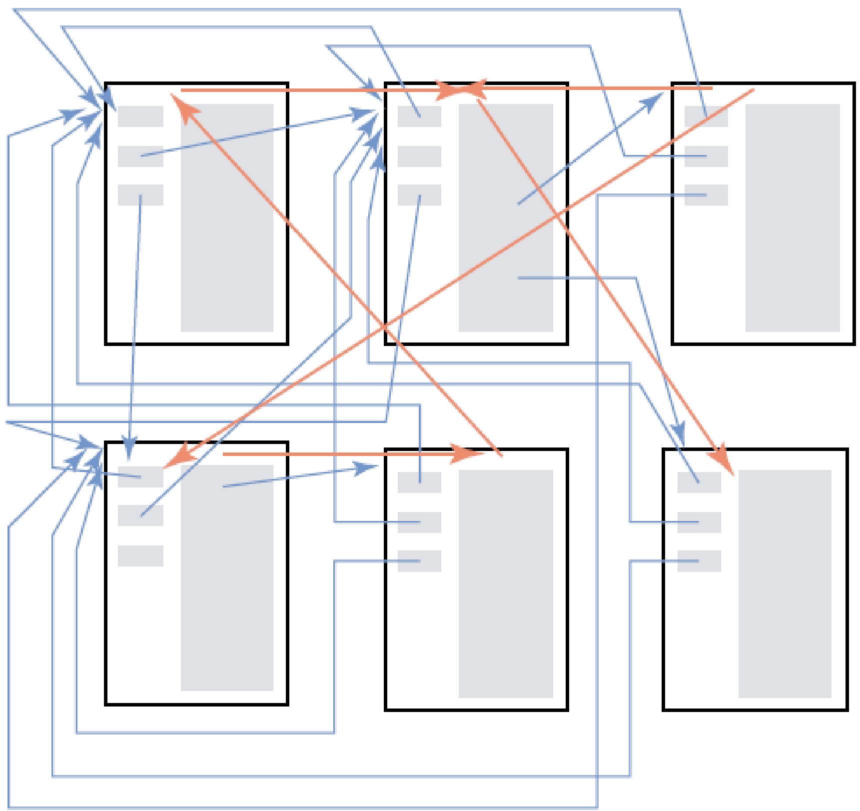


Film: fixed order of frames defines a single playback sequence.



What is Multimedia?

- Users can interact with digital multimedia in novel ways, leading to non-linear structures



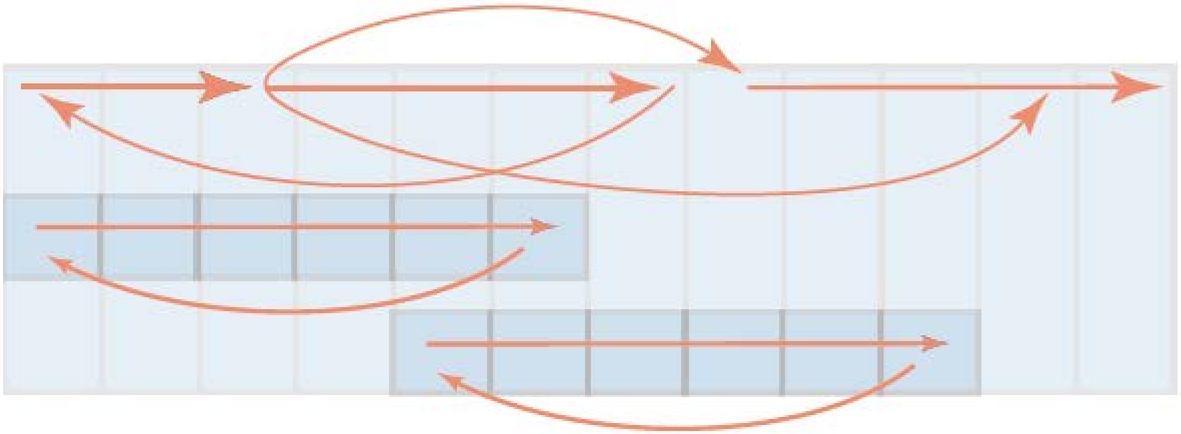
Hypermedia: links between pages permit multiple arbitrary reading orders.

What is Multimedia?

- Hypermedia
  - An extension of the term hypertext.
  - A **non-linear** medium of information which includes graphics, audio, video, plain text and hyperlinks.
  - Multimedia=Hypermedia+ non-interactive linear presentations.
    - World Wide Web (WWW) is a classic example of hypermedia.
    - Film in cinema is multimedia but not hypermedia.

What is Multimedia?

- Nonlinear Structure (cont.)



Flash: jumps between frames controlled by interactivity; independent movie clips play in parallel.

What is Multimedia?

- Some comments:
  - Digital multimedia can interact with other sorts of data and computation, serving as a user interface to databases and applications.
  - Multimedia is a relatively immature technology, although its adoption is accelerating with the increasing power of computer systems.
  - The history of the development of film demonstrates that it takes much more time than multimedia has existed for new media forms to develop fully.
  - Most multimedia adapt the forms of older media, but unique new forms are beginning to emerge.

Introduction

- What is Multimedia?
- **Multimedia Software Tools**
- Interactivity
- Design Issues
- Source Coding
- Legal Issues

Multimedia Software Tools

Digital Audio	Macromedia Sound Edit, CoolEdit
Music Sequencing and Notation	Cakewalk, Cubase Logic Audio, Marc of the Unicorn Performer
Image/Graphics Editing	Adobe Photoshop, Adobe Premiere Macromedia Freehand, GIMP ……
Animation	Avid SoftImage, OpenGL
Multimedia Authoring	Macromedia Director, Authorware
Website Design	Adobe Fireworks, Dreamweaver…
Flash	Macromedia Flash
……	……

Some Multimedia Applications

- Digital video editing and production systems
- Electronic newspapers/magazines
- World Wide Web
- Online reference works: e.g. encyclopedia, etc
- Home shopping
- Interactive TV
- Multimedia courseware
- Video conferencing
- Video-on-demand
- Interactive movies

Exercises

- Question
  - Do you think a lecture to be an example of linear(time-based) multimedia?
  - In what context do you encounter linear structures on a web?

Introduction

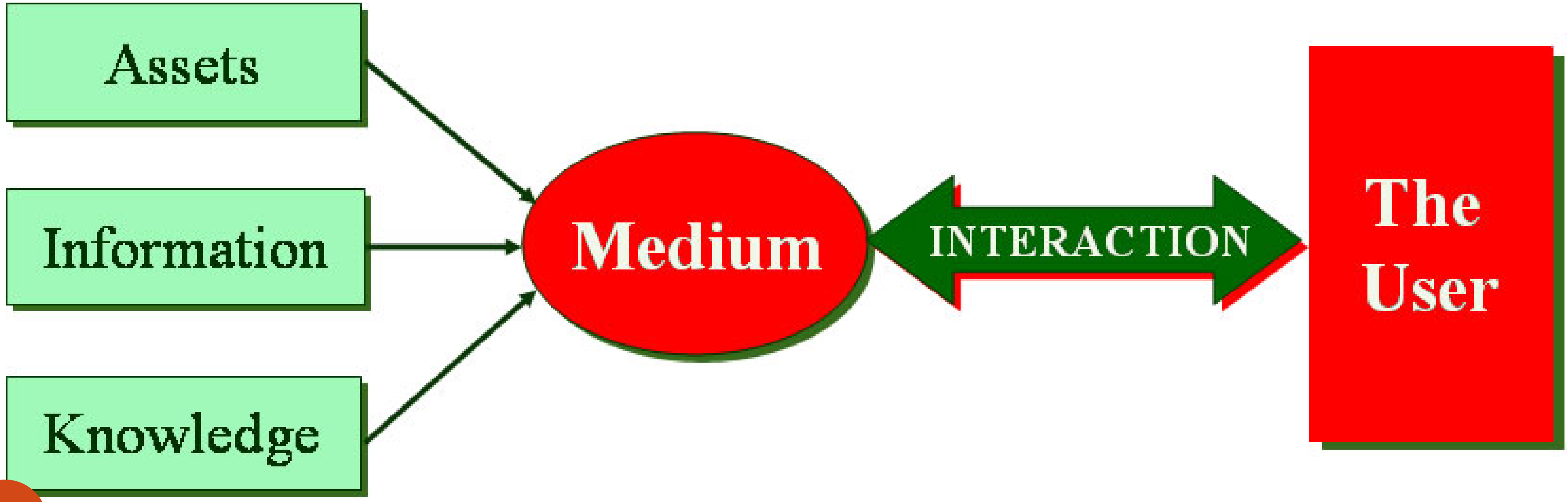
- What is Multimedia?
- Multimedia Software Tools
- **Interactivity**
- Design Issues
- Source Coding
- Legal Issues

Interactivity

- Interactive Multimedia
  - An integrating relationship between multimedia and interaction
  - A multimedia system in which related items of information are connected and can be presented together
- A combination of:
  - knowledge and information
  - a set of technologies
  - a set of multimedia components (modality, channels of communication, medium)
  - a set of collaborative systems
  - virtual environments

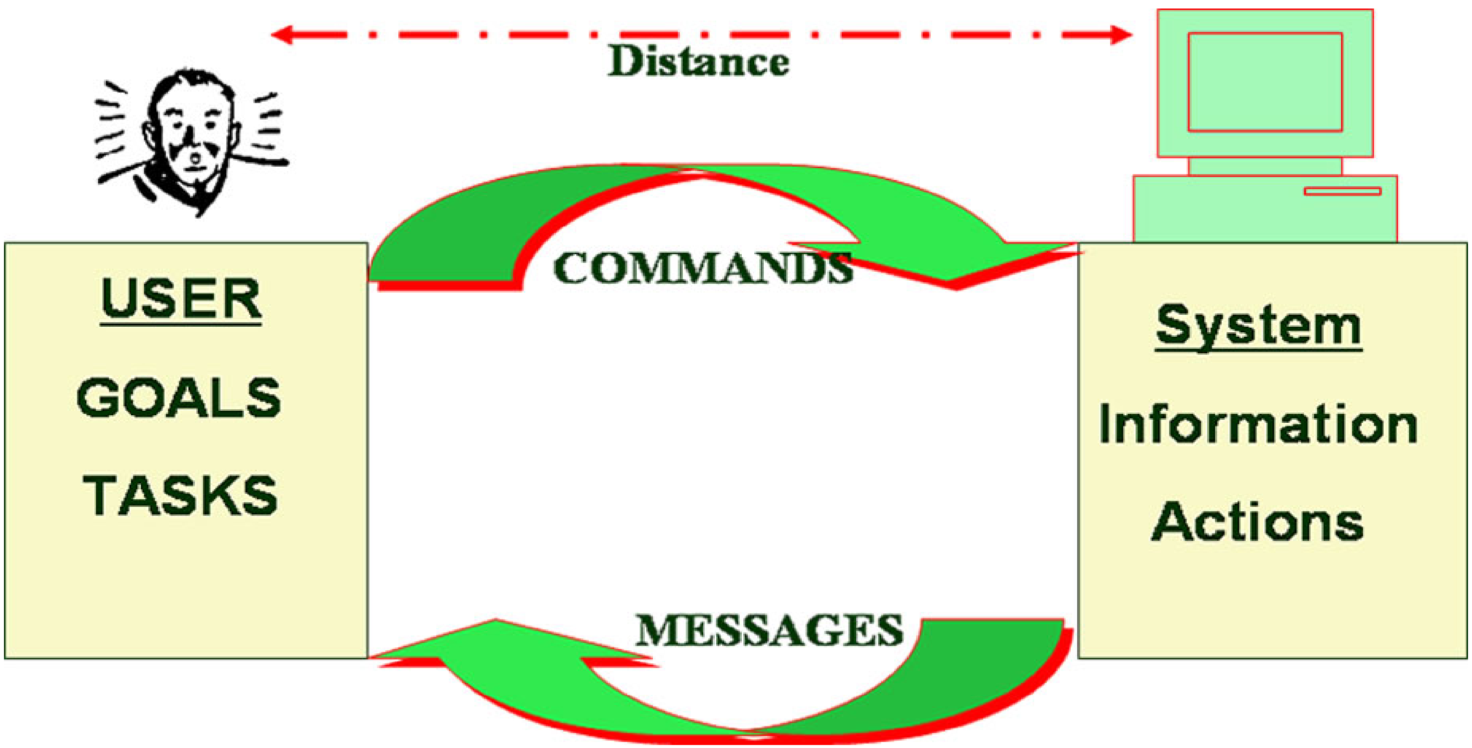
Interactivity

- Components of Interactive Multimedia
  - Asset – an object which encapsulates a single piece of 'media' (e.g. video, sound clip, graphic)
  - Information – the collection of data by a particular encoding
  - Knowledge – the interpretation and understanding of information



Interactivity

- Human Computer Interface (HCI)
  - HCI – the field concerned with the interface between humans and all forms of computer systems



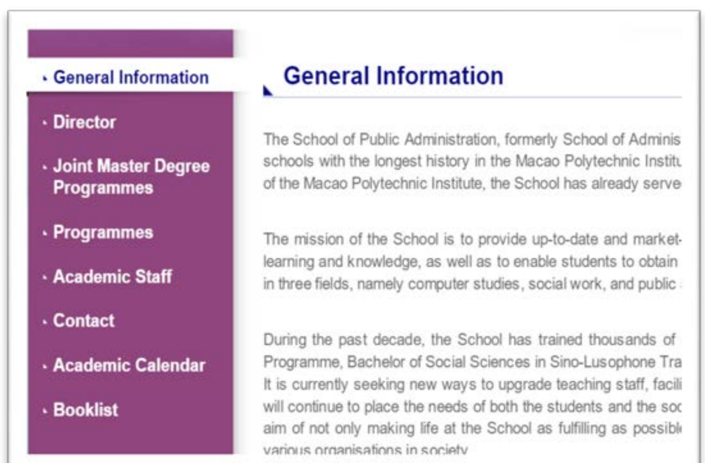
Interactivity with Multimedia

- Example: Hyperlinks
  - In hypertext, some means of indicating that a link can be followed by clicking is required.
  - There is no precedent for this requirement in traditional media. The nearest thing is cross-reference in print.
  - Underlining and blue colouring is the most common signifier for links. It may be implemented in CSS (Cascading Style Sheets) as a text decoration or a bottom border on a elements.

- Visit Macao Polytechnic Institute's website for more details
- Visit Macao Polytechnic Institute's website for more details
- Visit ***Macao Polytechnic Institute's website*** for more details

Interacting with Multimedia

- It is common practice to add some highlight to links when the cursor moves over them, to indicate that something will happen if a user clicks.
- Navbars and other collections of links may be moved to a separate area, where their function is evident without additional decoration.
- Users often expect images to have links on them. Provide a link where possible, but avoid using an image as the only link to a destination.

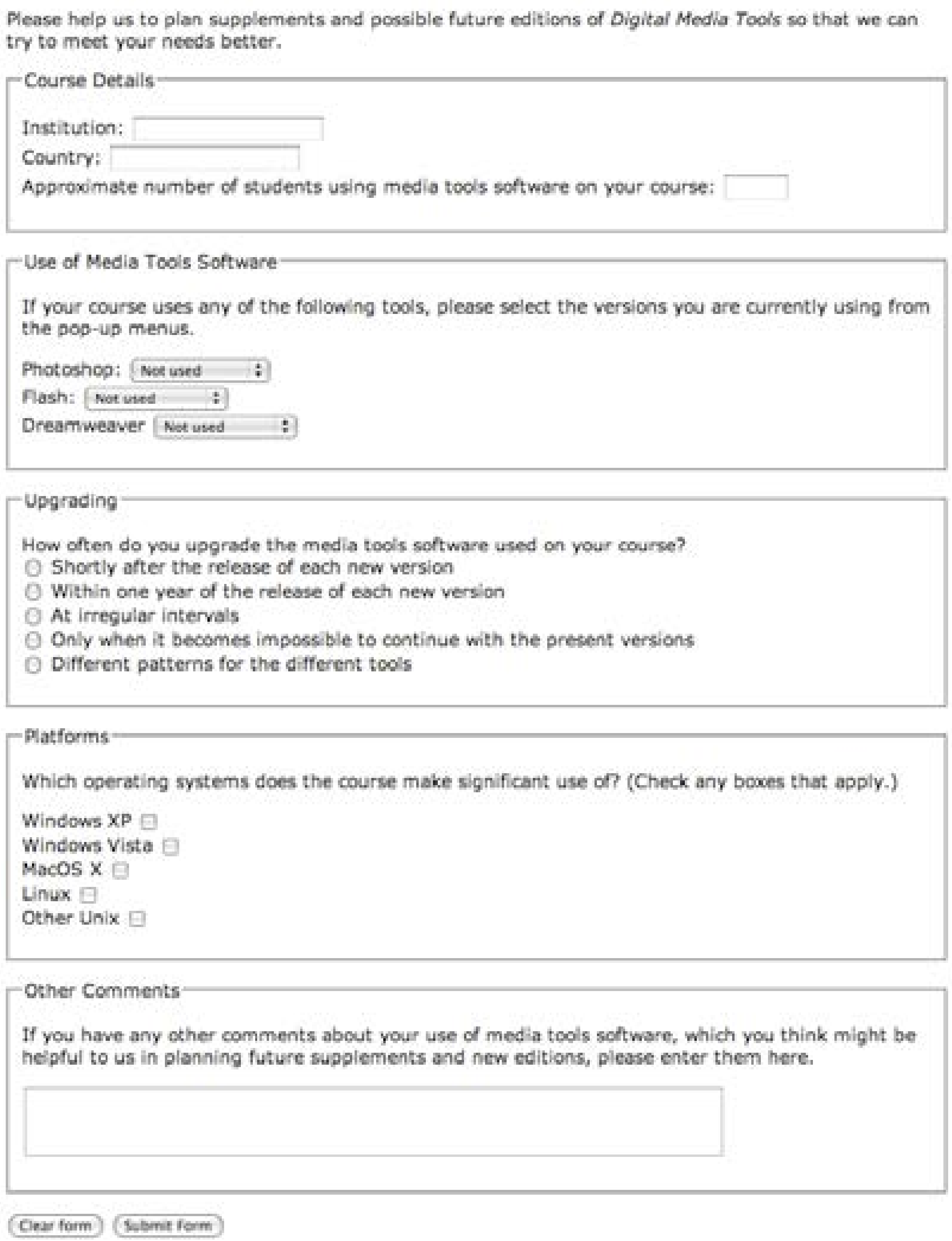


Interactivity with Multimedia

- Interacting through Multimedia
  - By embedding controls, we can provide ways of interacting with data or computation. Standard dialogue controls can be used for such purposes.
  - XHTML provides input elements, for text fields, check boxes, radio buttons, etc., text area for multiple lines of text and select and option elements for pop-up menus and lists. These elements are used within a form to send data to a script on the server.

type Attribute	Control	Type-specific Attributes
text	text input field	maxlength
checkbox	check box	checked, value
radio	radio button	
submit	submit button	
reset	reset button	
button	push button	
file	file selector	

Principal types of input element



Please help us to plan supplements and possible future editions of *Digital Media Tools* so that we can try to meet your needs better.

Course Details

Institution:

Country:

Approximate number of students using media tools software on your course:

Use of Media Tools Software

If your course uses any of the following tools, please select the versions you are currently using from the pop-up menus.

Photoshop: Not used

Flash: Not used

Dreamweaver Not used

Upgrading

How often do you upgrade the media tools software used on your course?

- Shortly after the release of each new version
- Within one year of the release of each new version
- At irregular intervals
- Only when it becomes impossible to continue with the present versions
- Different patterns for the different tools

Platforms

Which operating systems does the course make significant use of? (Check any boxes that apply.)

Windows XP

Windows Vista

MacOS X

Linux

Other Unix

Other Comments

If you have any other comments about your use of media tools software, which you think might be helpful to us in planning future supplements and new editions, please enter them here.

Clear form | Submit form

Text fields

Pop-up menus

Radio buttons

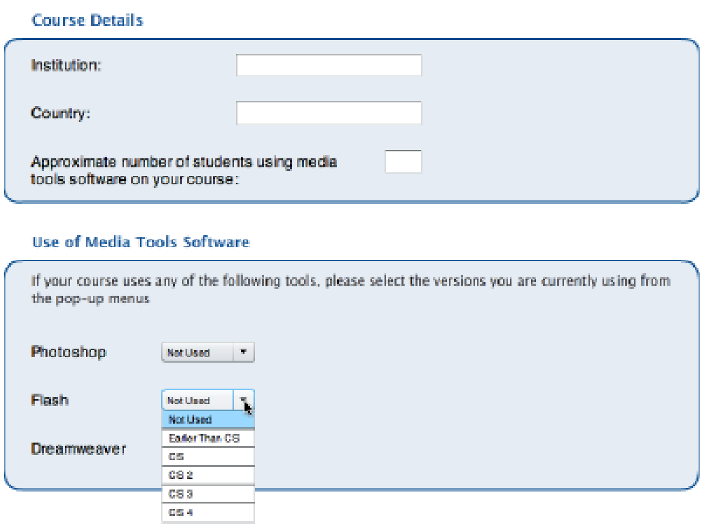
Check boxes

Text area

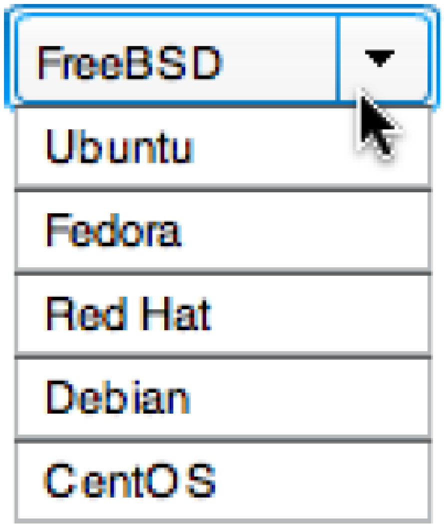
button

Interactivity with Multimedia

- Interacting through Multimedia
  - Flash UI components provide the same controls, plus a few others. They must be combined with some ActionScript to do anything useful.



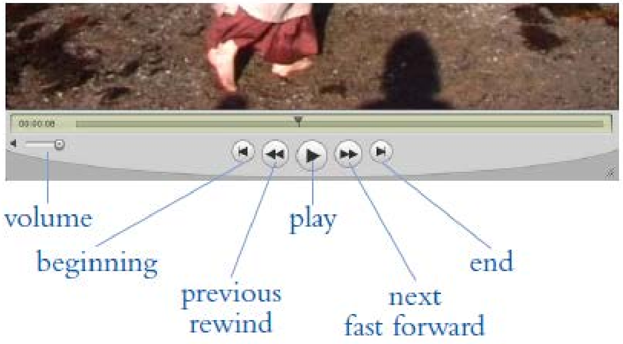
Flash text field, label and combo box components



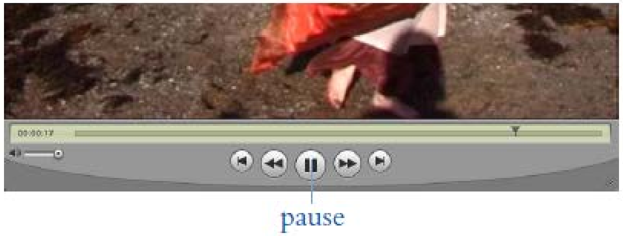
A ComboBox

Interactivity with Multimedia

- Example:
  - Controls used by software for playing time-based media are derived from an established set of buttons used by physical media players: play, pause, stop, rewind, fast forward.
  - Software allows controls to be more flexible than physical equivalents.



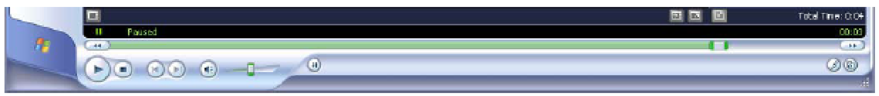
QuickTime Player controls



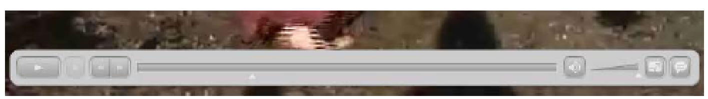
QuickTime Player controls while playing

Interactivity with Multimedia

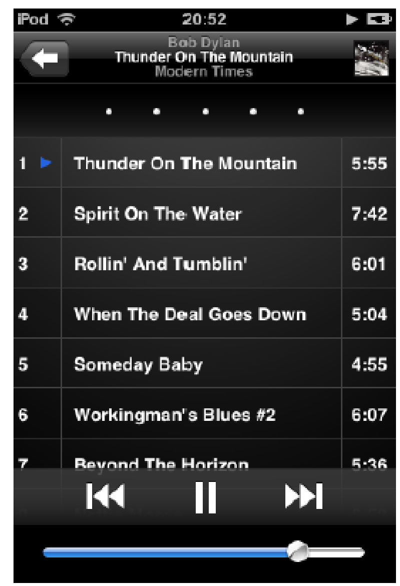
- Example (cont.):
  - Media player controls use semiotic and gestalt principles: a set of standard icons are arranged so that they are perceived as a unit.



Windows Media Player controls



Flash video playing controls



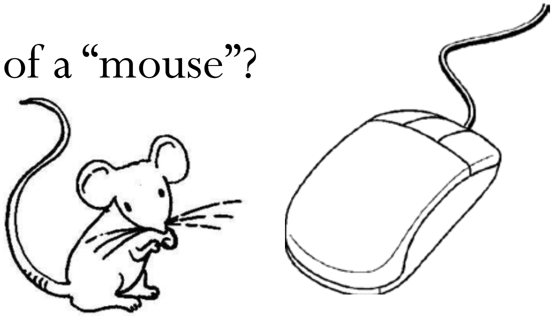
iPod touch controls

Interactivity with Multimedia

- Interacting through Multimedia (Further examples)
  - Flash movie clips and ActionScript can be used to create interfaces that are not restricted to using standard controls.
  - Flash-based interfaces can be used in a web browser or in desktop applications.
  - Multimedia applications which do something new may require innovative interface design.
  - Users will draw on existing experience when trying to use new interfaces, so familiar features and ideas should be used where possible.
  - JavaScript and other web-standard technology can be used to program multimedia interfaces, but the possibilities are less extensive than those which Flash offers. JavaScript libraries are used to make the task simpler.

Interactivity

- Question
  - Do you think microsoft windows provide a good HCI to its users?
  - Or, which HCI is the best? DOS? Windows? OS/2? MAC OS? UNIX?
  - How do you rate the operating systems on mobile phones?
  - Can you provide any hint in further improving their ***usability***?
  - Suppose you are in charge of a software system, are you willing to devote much effort to HCI design?
  - How do you think about the contribution of a "mouse"?



Introduction

- What is Multimedia?
- Multimedia Software Tools
- Interactivity
- **Design Issues**
- Source Coding
- Legal Issues

Visual Design

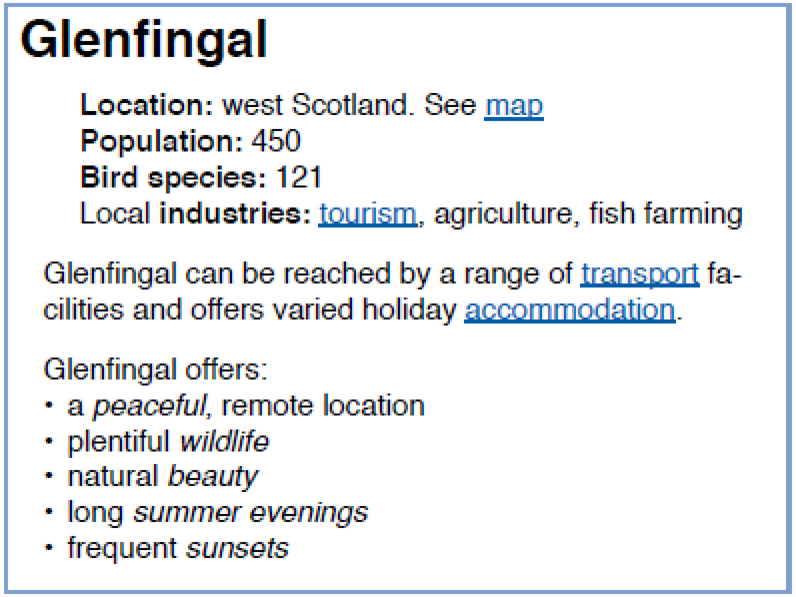
- Visual Communication
  - Semiotics
- Gestalt Principles
  - Grouping, Visual Hierarchy
- Colour and Tone
  - Colour in Multimedia design
  - Combining Colours
- Layout Grids
  - Alignment
  - Grids

Design Issues

- Poor visual design is one of the main factors that lead to multimedia applications and Web sites being difficult to use.
- Visual communication depends upon a range of psychological, cultural and physical factors which cannot easily be quantified or systematized.
- Images and colors immediately convey impressions which cannot necessarily be adequately described in words.

Design Issues

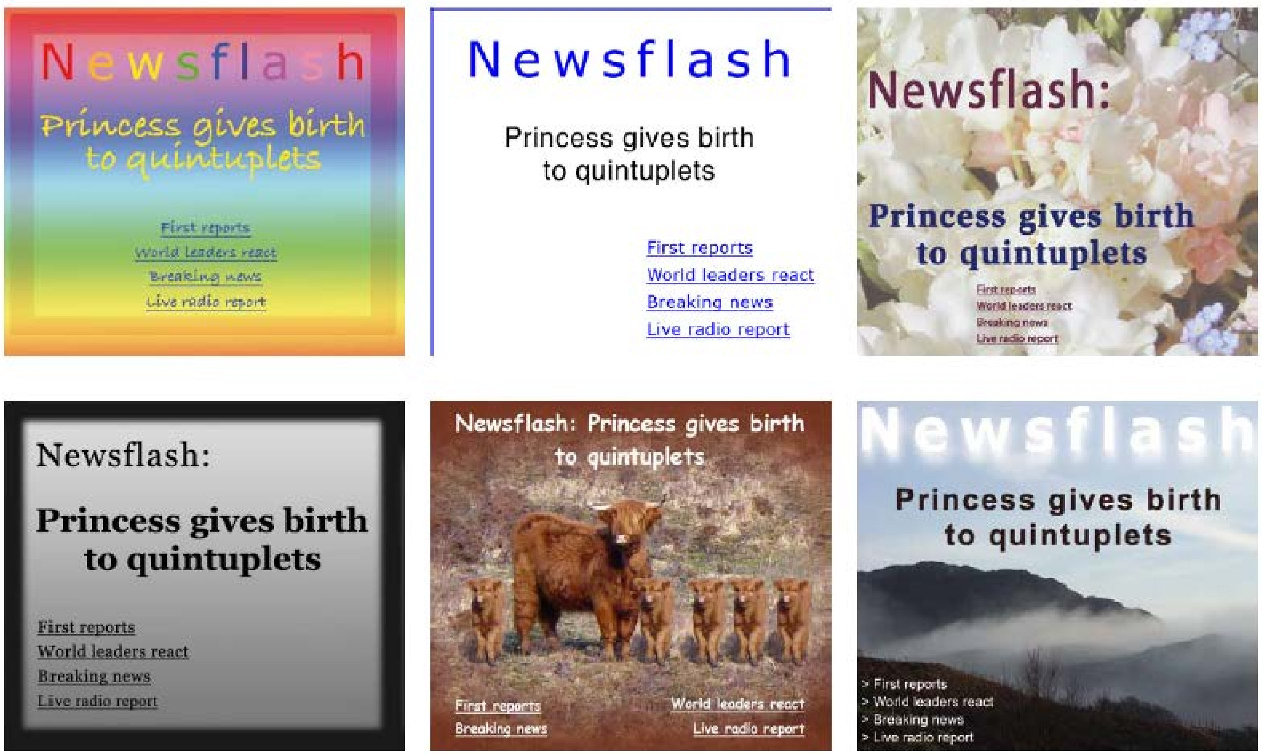
- Which of the pages would make you want to visit Glenfingal?



Combination of photo, warm colour, font and navbar

Design Issues

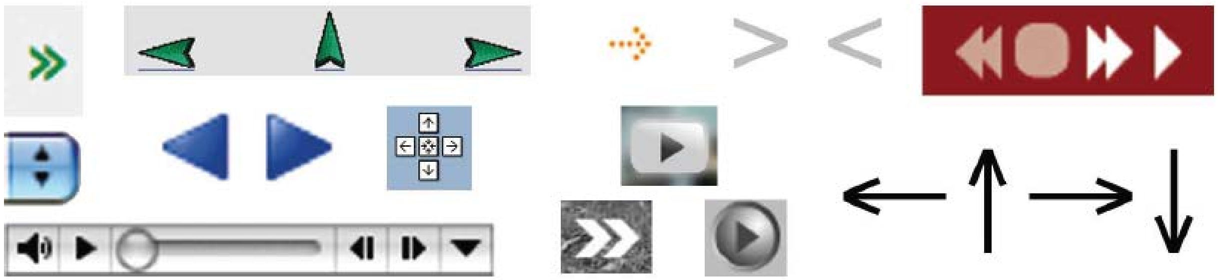
- Visual communication may alter the meaning of what is being communicated by words, in a gross or subtle way.



Interaction of textual and visual communication

Design Issues

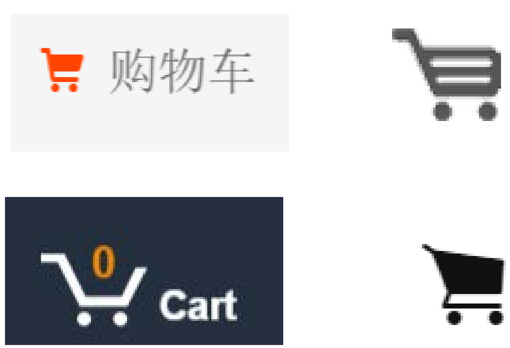
- Semiotics is the study of systems of signs and the relationship between signifier and signified within them.
- Signifier: a sign's form e.g., a word or a graphic symbol.
- Signified: specific meaning or concept which a sign refers to.
- The relationship between the signifier and the signified is arbitrary and can only be understood within a particular system of signs.

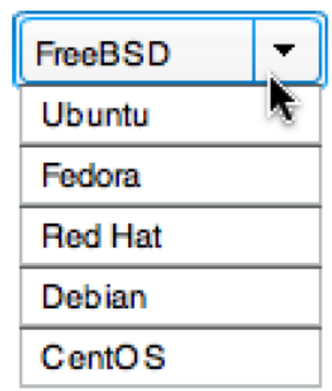


Arrows as signifiers

Design Issues

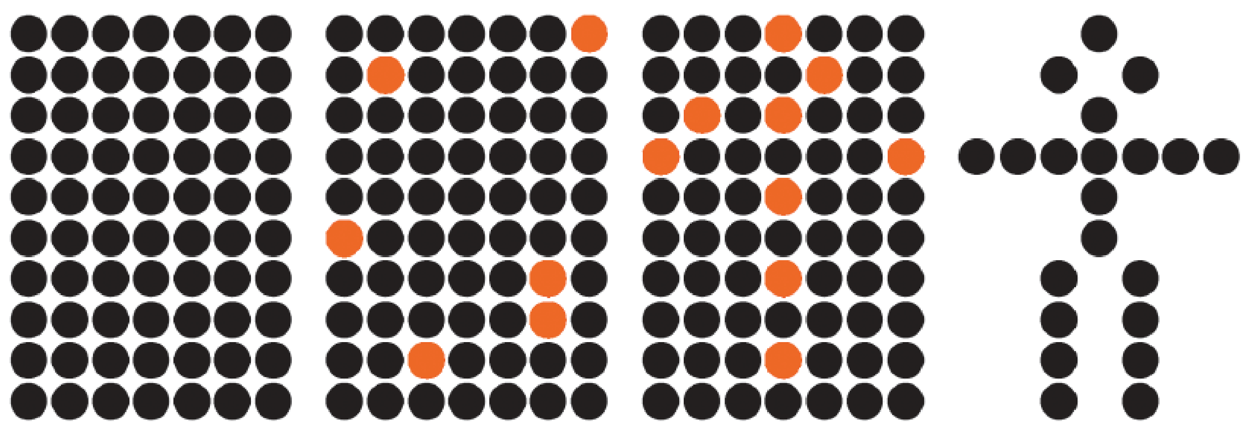
- Convention, context and users' experience determine whether a user will understand a sign correctly.
- Advantages of using graphic symbols
  1. Graphic symbols transcend language barriers.
  2. Graphic symbols are capable of conveying complex meanings succinctly and there may sometimes be no sensible alternative to using them.





Design Issues – Gestalt Principle

- Gestalt principles of visual perception are derived from the theories of gestalt psychology.
- They are concerned with how the human brain tends to organize the visual information that reaches it through eyes.



Gestalt principles of visual perception

http://en.wikipedia.org/wiki/Gestalt_psychology

Design Issues – Gestalt Principle

- Perception of patterns and structures may be determined by the grouping of objects in a visual field.
- The gestalt principles of proximity, similarity, symmetry, figure/ground and closure determine our recognition of grouping.

Design Issues

- Law of Proximity — Spatial or temporal proximity of elements may induce the mind to perceive a collective or totality.

Design Issues – Gestalt Principle

- Law of Similarity — The mind groups similar elements into collective entities or totalities. This similarity might depend on relationships of form, color, size, or brightness.

Gestalt Principle

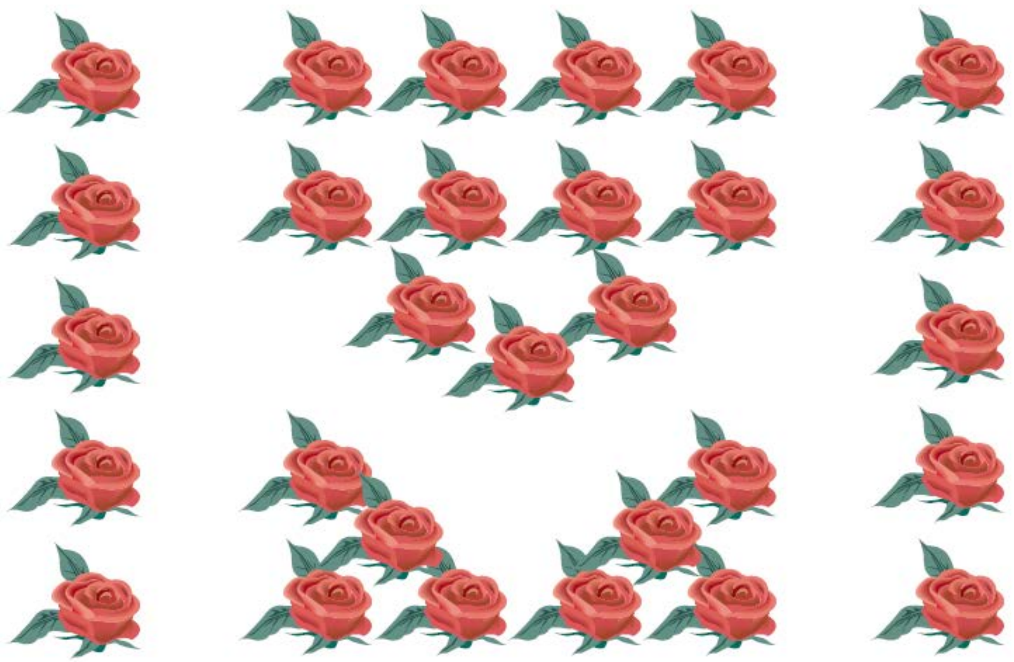
Synthetic image (different brightness and resolution)

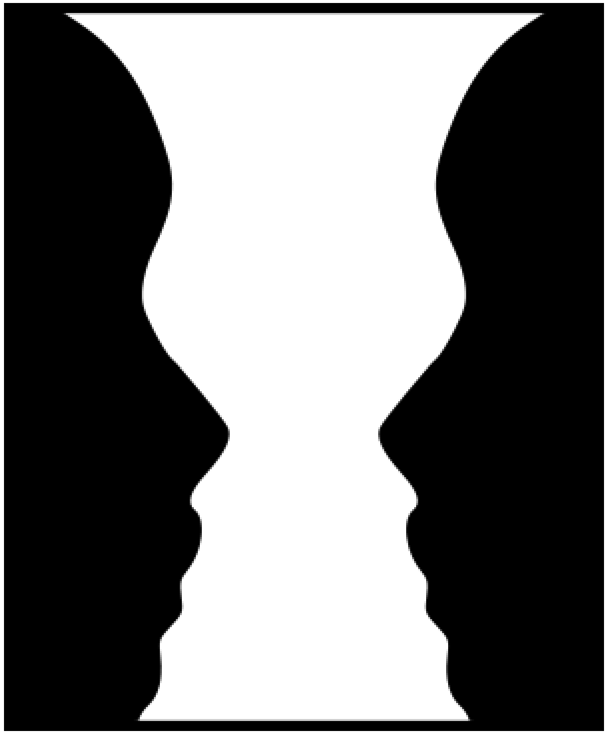
Gestalt Principle

- Law of Symmetry — Symmetrical images are perceived collectively, even in spite of distance.

Gestalt Principle

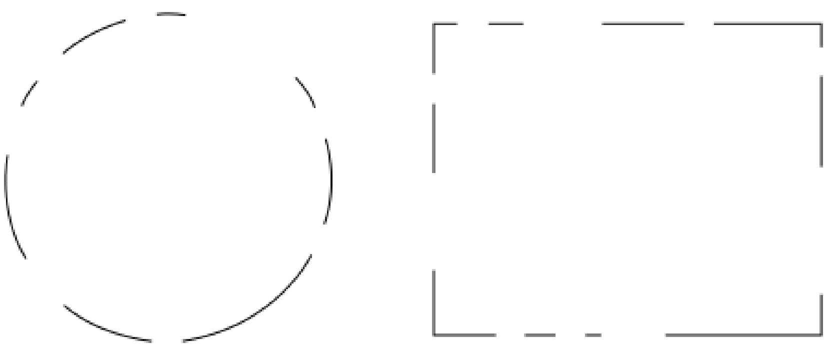
- Figure/Ground — convey two visual ideas at the same time





Design Issues – Gestalt Principle

- Law of Closure: The mind may experience elements it does not perceive through sensation, in order to complete a regular figure (that is, to increase regularity).

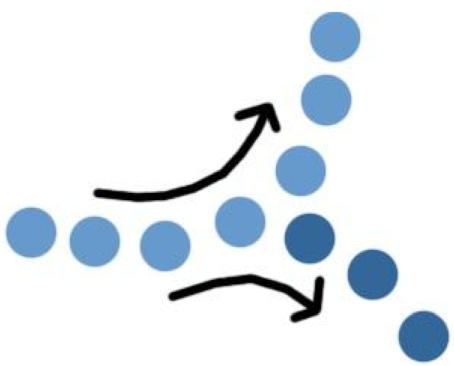


The famous use of Law of Closure

- Who bites the apple?

Design Issues – Gestalt Principle

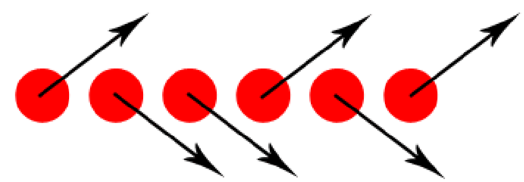
- Law of Continuity —The mind continues visual, auditory, and kinetic patterns.
- Law of Common Fate — Elements with the same moving direction are perceived as a collective or unit.



Law of Continuity:

Lines are seen as following the smoothest path.

In the image above, the top branch is seen as continuing the first segment of the line. This allows us to see things as flowing smoothly without breaking lines up into multiple parts.



Design Issues – Gestalt Principle

- Non-symbolic ordering based on the gestalt principles is the foundation of structure in visual design.
- The precise appearance and arrangement of objects may lead to one principle dominating the others.
- Ignoring gestalt principles often results in a confusing design.

Monday	Heavy Rain
Tuesday	Heavy Rain
Wednesday	HEAVY RAIN
Thursday	Heavy Rain
Friday	Heavy Rain
Saturday	*Heavy Rain*
Sunday	*Heavy Rain*

A confusing absence of similarity

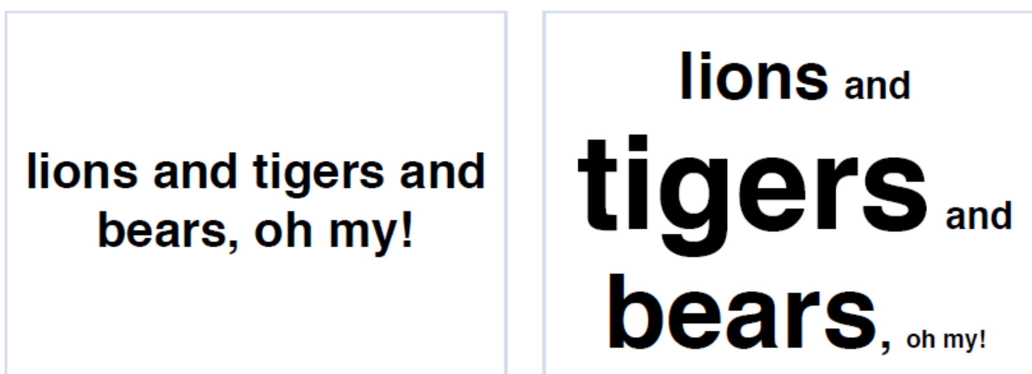
Design Issues – Gestalt Principle

- The component parts of an interface or Web page should usually be organized according to gestalt principles.
- Navbars on Web pages, and the conventional arrangement of controls on media players illustrate the application of gestalt principles to multimedia design.

Using gestalt principles in Web page design

Design Issues – Gestalt Principle

- Visual hierarchy describes the dominance of one or more elements in the visual field.
- Like other hierarchies, it may have many levels.
- Visual hierarchy may be achieved by applying gestalt principles "inversely", in order to disrupt grouping and make one or more elements appear dominant.



Expressing hierarchical emphasis through type size

Design Issues – Gestalt Principle

- Visual hierarchy is not necessarily determined by size.

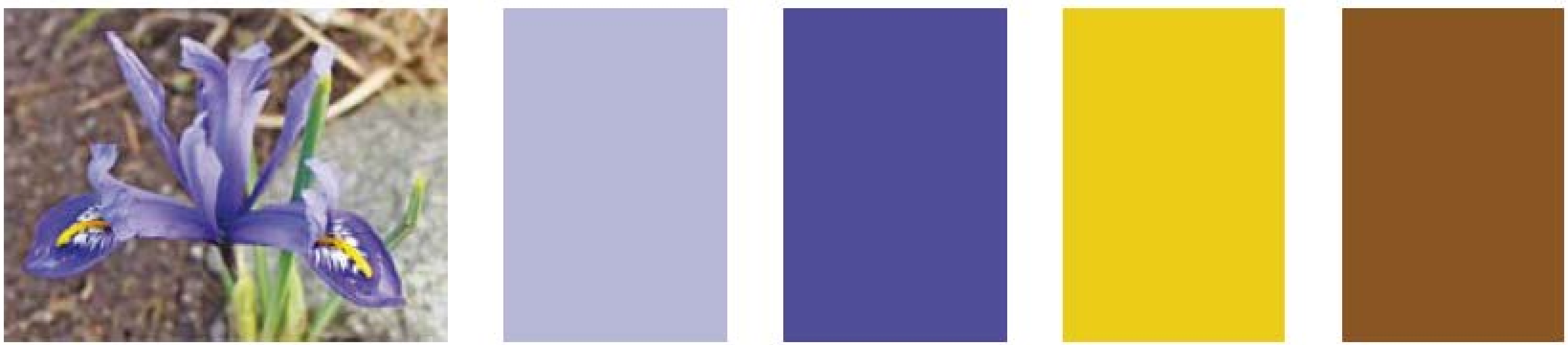
Design Issues – Color & Tone

- Color plays many roles in visual design, affecting visual hierarchy, perception of structure, and even meaning.
- Individuals' responses to color may be emotive, or determined by fashion or culture.
- When colors are juxtaposed, the perceived appearance is modified.
- Hue and brightness may appear to be modified when the same color is placed against different backgrounds.

The effect of colour combinations on perception of colour

Design Issues – Color & Tone

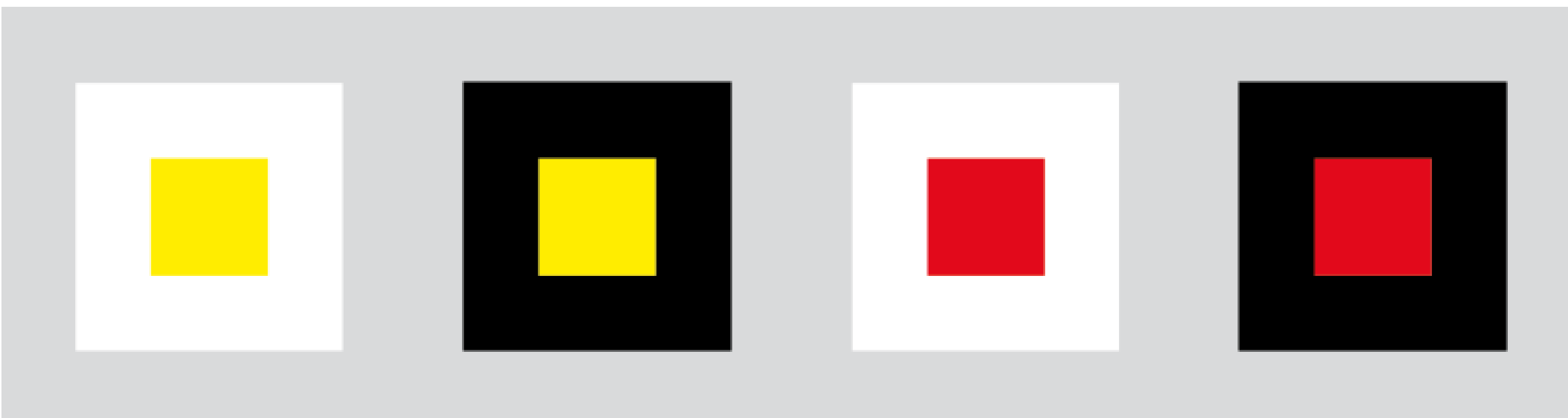
- Large flat areas of a single color look quite different from pixels of the same color within an image.



Colours removed from their pictorial context

Design Issues – Color & Tone

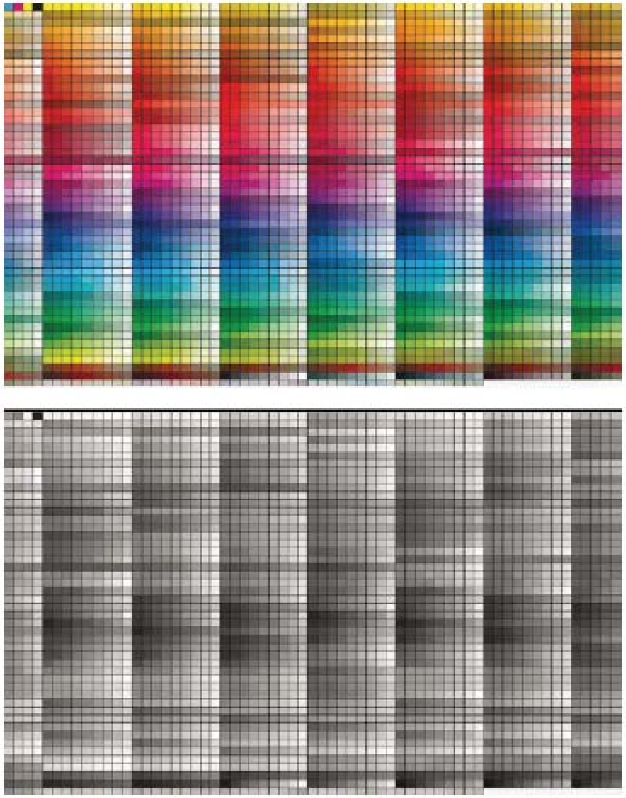
- Colors may seem to advance or recede when placed on different-colored backgrounds.



The effect of colour combinations on perception of size

Design Issues – Color & Tone

- Tonal contrast affects perception of the distinction between figure and ground.
- Contrast therefore affects the legibility of text.
- It can be difficult to judge the contrast between colors of different hues.



Colours and tones

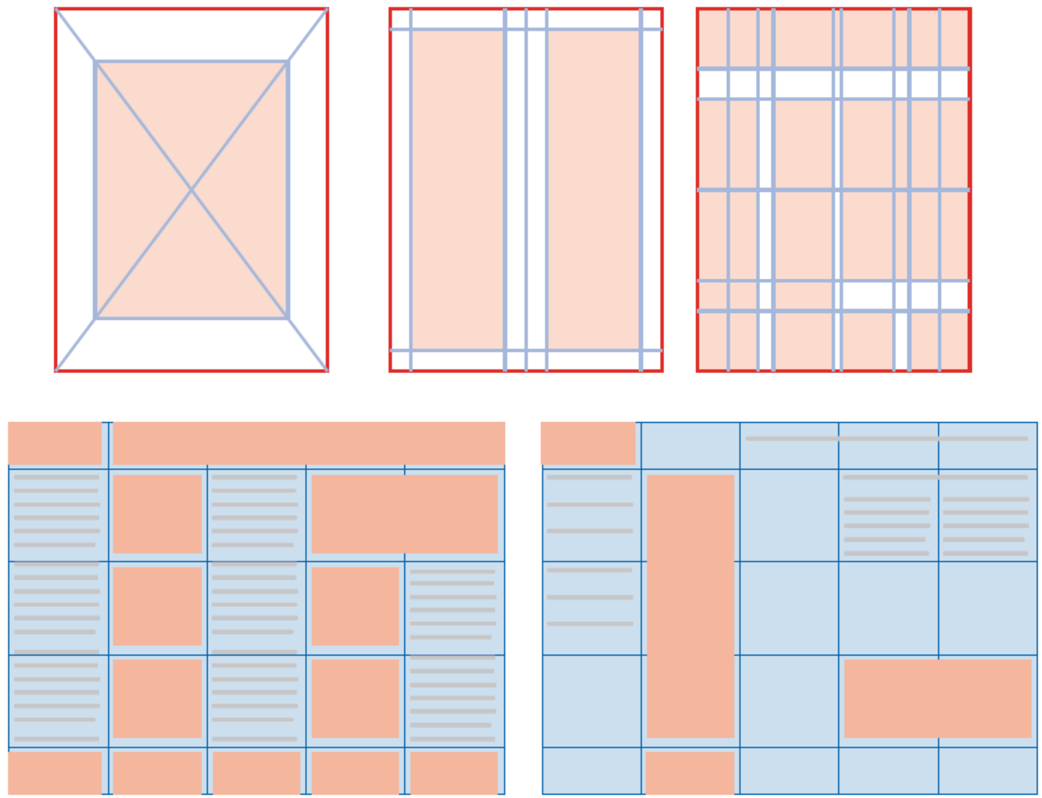
Design Issues – Color & Tone

- As people age, less light enters the eye, so a smaller range of tonal values is perceived.
- Many people suffer from defective color vision, most commonly an inability to distinguish between red and green.
- The tonal contrast of colored designs should be tested by converting to greyscale.

Different colours with similar tonal values (red–green confusability)

Design Issues – Layout Grids

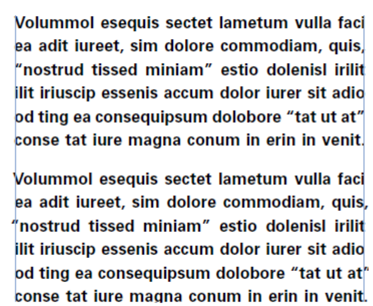
- A layout grid is a geometrical division of the page that can be used to structure the placement of text blocks and images.
- The grid itself is simply an aid to layout and remains invisible.



Design Issues – Layout Grids

- Alignment can give an appearance of coherence and visual order to a Web page or multimedia interface.
- It is sometimes necessary to misalign irregular objects slightly to make them look as if they are perfectly aligned.
- Hanging punctuation is used to prevent punctuation marks at the ends of lines of text appearing misaligned.

Aligned irregular shapes



Hanging punctuation

Design Issues – Layout Grids

- Centred layouts may be problematic – images and headings may look untidy unless one of the aligned elements is significantly narrower than the other.

Centred alignment

Design Issues – Layout Grids

- Modified grids may be used to accommodate the dynamic dimensions of Web pages while maintaining a framework for vertical and horizontal alignments.
- Arbitrary grid layouts may be created in Flash.

Introduction

- What is Multimedia?
- Multimedia Software Tools
- Interactivity
- Design Issues
- **Source Coding**
- Legal Issues

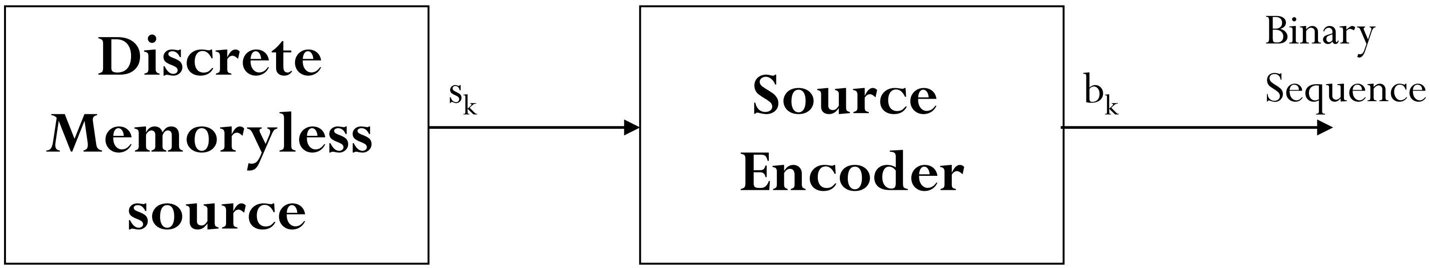
Source Coding

- In computer science and information theory, data compression or source coding is the process of encoding information using fewer bits (or other information-bearing units) than an un-encoded representation would use, through use of specific encoding schemes.
- http://en.wikipedia.org/wiki/Source_coding

Source Coding

- Source coding is a process by which data that is generated by a discrete source is represented efficiently.
- The knowledge of the statistics of the source is required in order to develop an efficient source code.
- In particular, we may assign short code-word to frequent source symbols and long codewords to rare source symbols.
- Such a source code is referred to as variable-length code.

Source Coding



Source Coding

- Consider a source that has an alphabet of K different symbols such that $\boldsymbol{s_k}$ is the k-th symbol in the alphabet.
- Also let $\boldsymbol{s_k}$ occur with probability $\boldsymbol{p_k}$ and let the binary codeword assigned to $\boldsymbol{s_k}$ by the encoder have length $\boldsymbol{l_k}$ measured in bits.
- The average codeword length of the source encoder is defined as

$$\overline{L} = \sum_{k=0}^{K} p_k l_k$$

Source Coding

- $\overline{L}$ represents the average number of bits per source symbol used in the source encoding process.
- If $\boldsymbol{L_{min}}$ denotes the minimum possible codeword length, the coding efficiency of the source encoder is defined as

$$\eta = \frac{L_{\min}}{\overline{L}}$$

- With $\overline{L} \geq L_{\min}$, then $\eta \leq 1$.
- The encoder is said to be efficient when η approaches 1.

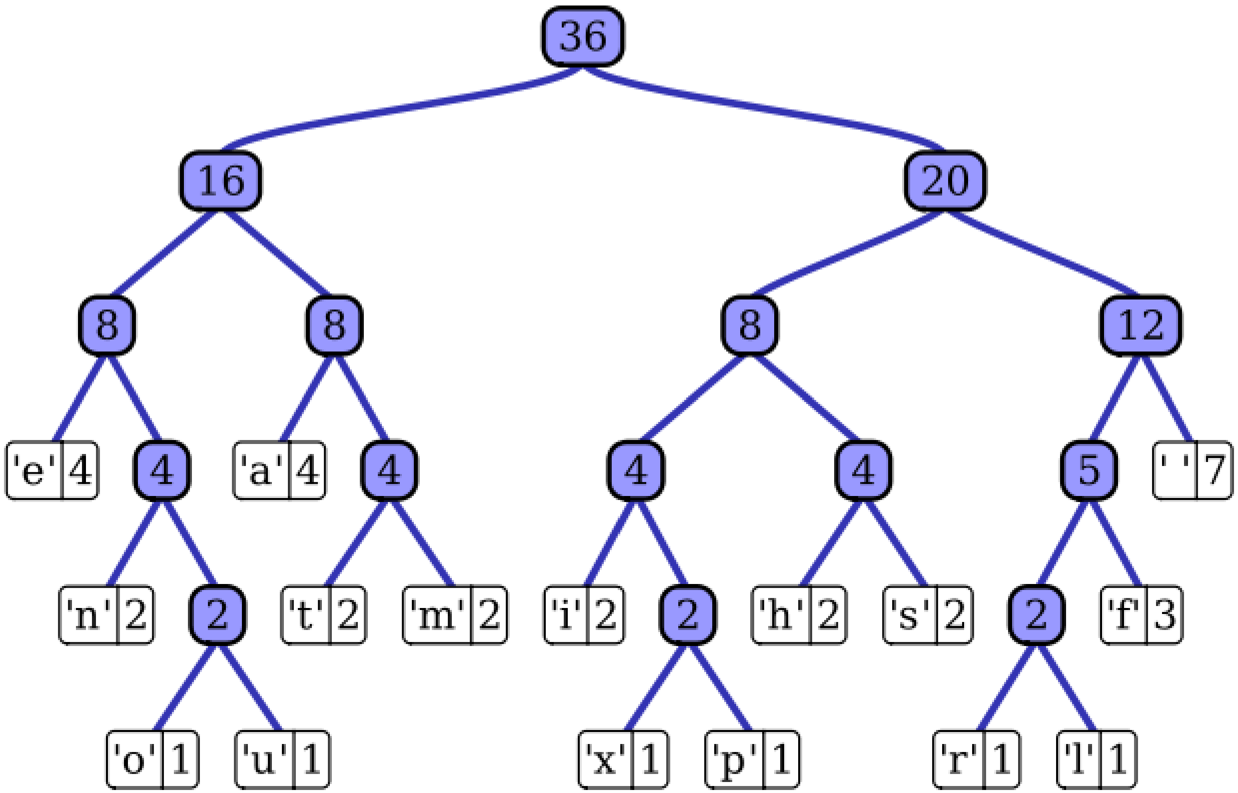
Source Coding

- Examples: The Huffman code
- The Huffman code is an efficient source code whose η approaches 1 (ie: average word length approaches the fundamental limit).
- Huffman code uses the statistics of the source to generate a binary sequence that represents the source symbols.

Source Coding

- The Huffman Algorithm proceeds as follows:
  1. The source symbols is listed in order of decreasing probability. The two source symbols with the lowest probability are assigned a 0 and a 1 (Splitting) .
  2. The two source symbols are regarded as being combined into a new source symbol with probability equal to the sum of the two original probabilities. The probability of the new symbol is placed in the list in accordance with its value.
  3. The procedure is repeated until we are left with a final list of source statistics of only two for which a 0 and a 1 is assigned.
- The code for each original source symbol is found by working back and tracing the sequence of 0s and 1s assigned to that symbol as well as its successors.

Source Coding



Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110
p	1	10011
r	1	11000
u	1	00111
x	1	10010

"this is an example of a huffman tree"

Introduction

- What is Multimedia?
- Multimedia Software Tools
- Interactivity
- Design Issues
- Source Coding
- **Legal Issues**

Legal Issues

- Different countries/regions have different rules

- General guidance:
  - Respect others' work/privacy/assets…
  - Protect one's own work/privacy/assets…

- Distinguish the different types of licenses
  - E.g. GNU license, shareware, freeware, academic license, commercial license, student license…

Legal Issues

- Is it legal if:
  - Copy commercial software instead of buying it, then use it for educational purposes?
  - Use information from the internet without giving recognition?
  - Use information from another source without giving recognition?
  - Download a movie using P2P networks?

Graphics and Image Data Representations

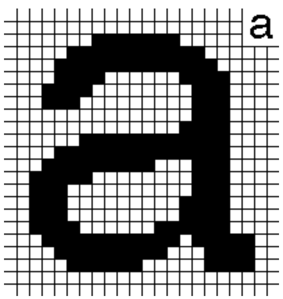
- Font Technologies
- 2-D Computer Graphics
- Graphics/Image Data Types
- Popular File Formats
- Further Exploration

Font Technologies

- Two techniques for displaying text on computer:
  - Bitmapped fonts
  - Outline fonts.

Bitmapped Fonts

- Pixels that make the letter are described by a binary code.
  - Every character is stored as a bitmapped letter, number, or symbol.
  - Require large memory and storage capacity.



Bitmapped Fonts

- Advantages
  - Precise control over letter appearance.
  - Letters can be edited at pixel level.

- Disadvantages
  - Letters can't be easily scaled.
  - Requires separate bitmaps for each typeface, style, and point size to be used.
  - Requires large storage capacities.

Outline Fonts

- Store a description of the character to be displayed.
  - Description is a series of commands to create the letter on the computer display.
- A set of instructions to draw the letter rather than a mapping of pixels.

- Outline font technology:
  - Adobe Postscript
  - TrueType.

Outline Fonts

- Advantages
  - Fonts are easily scaled.
  - Requires smaller storage capacity.

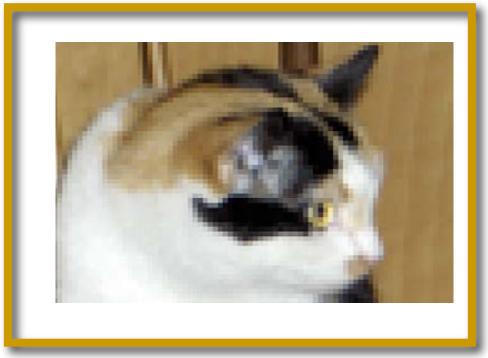
- Disadvantages
  - Cannot edit the look of the letter at pixel level
  - Limited creativity in overall design of character

2-D Computer Graphics

BITMAPPED IMAGES &
VECTOR DRAWN GRAPHICS

Bitmapped Graphics

- Bitmapped graphics
  - Created as a pattern of discrete elements.
  - Each element is a pixel or "picture element."
- Pixels
  - Small squares.
  - Assigned a binary code to define color.
    - More bits = more color possibilities



Bitmapped File Formats

- Compression of bitmapped graphics are:
  - Lossy
  - Lossless.
- Common graphic file formats are:
  - PICT
  - BMP
  - TIFF
  - JPEG
  - GIF
  - PNG.

Vector-Drawn Graphics

- Vector: a line with length, curvature, and direction.
- Vector graphics: images created from mathematically defined shapes.
- Draw programs: software used to create vector graphics.
- Main advantages:
  - Images can be enlarged without distortion
  - Small file size.

Vector Graphics File Formats

- Files are saved in native format or general purpose formats.
  - Native format: dependent on the application.
  - General purpose: can be used in many applications.
    - Vector-only:

      EPS—Encapsulated Postscript

      PDF— Portable Document Format.
    - Metafiles:

      SVG—Scalable Vector Format.

Advantages

BITMAPPED IMAGES

- Represent complex contones.
- Full-featured photo editing.
- Wide range of artistic effects.
- Precise editing.

VECTOR IMAGES

- Smooth scaling and reshaping.
- Ease of editing objects in layers.
- Low file size.

Disadvantages

BITMAPPED IMAGES

- Large file sizes.
- Loss of precise shapes when scaled or rotated.

VECTOR IMAGES

- Inaccurate, incomplete representation of complex contone images.
- No photo-editing capability.
- Limited artistic control.

Graphics/Image Data Types

- The number of file formats used in multimedia continues to proliferate.
- For example, Table 2.1 shows a list of some file formats used in the popular product adobe Director.

Table 2.1: Adobe Director File Formats

File Import					File Export		Native
Image	Palette	Sound	Video	Anim.	Image	Video	
.BMP, .DIB, .GIF, .JPG, .PICT, .PNG, .PNT, .PSD, .TGA, .TIFF, .WMF	.PAL .ACT	.AIFF .AU .MP3 .WAV	.AVI .MOV	.DIR .FLA .FLC .FLI .GIF .PPT	.BMP	.AVI .MOV	.DIR .DXR .EXE

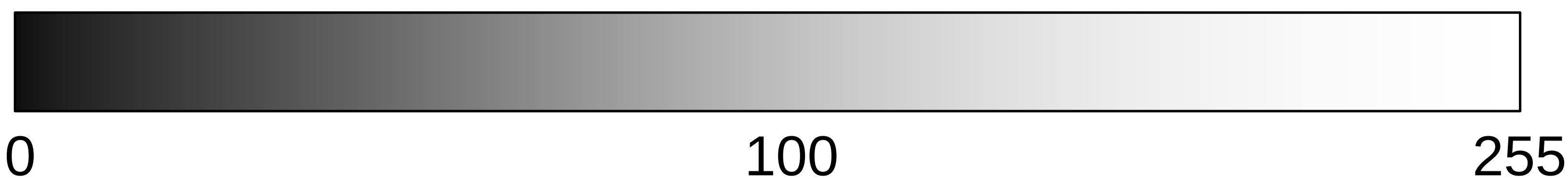
1-bit Images

- Each pixel is stored as a single bit (0 or 1), so also referred to as **binary image**.
- Such an image is also called a 1-bit **monochrome** image since it contains no color.
- Fig. 2.1 shows a 1-bit monochrome image (called "Lena" by multimedia scientists — this is a standard image used to illustrate many algorithms).

Fig. 2.1: Monochrome 1-bit Lena image.

8-bit Gray-level Images

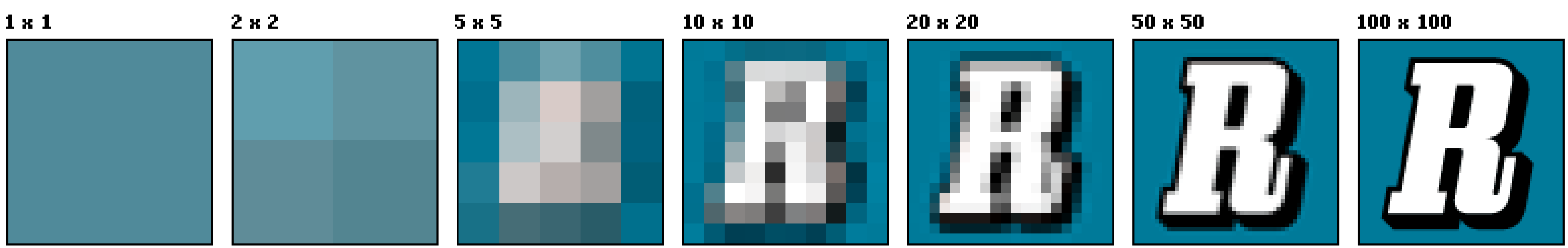
- Each pixel has a gray-value between 0 and 255. Each pixel is represented by a single byte; e.g., a dark pixel might have a value of 10, and a bright one might be 230.
- **Bitmap**: The two-dimensional array of pixel values that represents the graphics/image data.



8-bit Gray-level Images

- **Image resolution** refers to the number of pixels in a digital image (higher resolution always yields better quality).

  - Fairly high resolution for such an image might be 1,600 x1,200, whereas lower resolution might be 640 x 480.

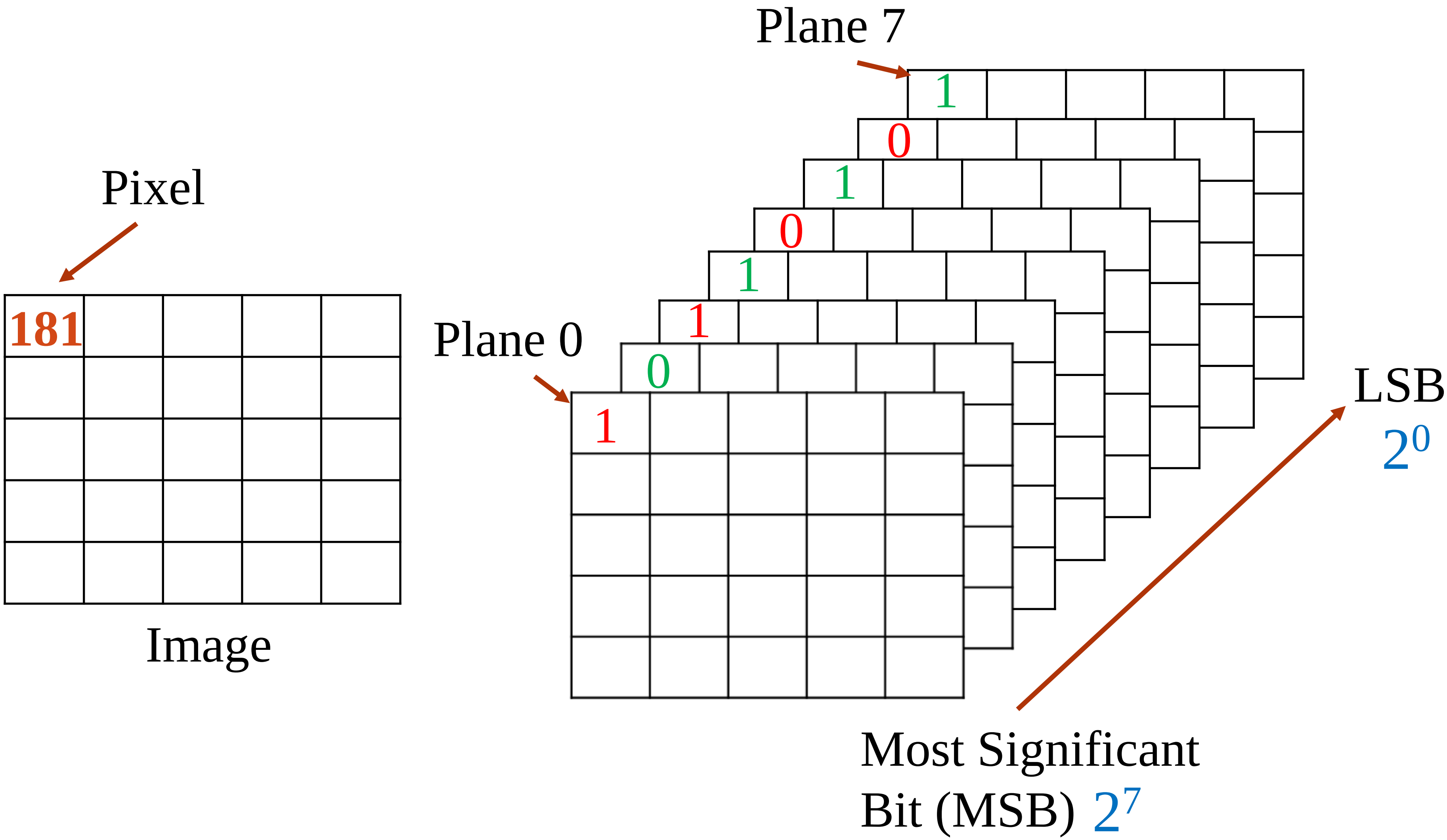


- **Frame buffer**: A portion of RAM (Random Access Memory used to store bitmap.

- **Video card** (actually a graphics card) is used for this purpose.

  - The resolution of the video card does not have to match the desired resolution of the image
  - but if not enough video card memory is available then the data has to be shifted around in RAM for display.

- 8-bit image can be thought of as a set of 1-bit bit-planes, where each plane consists of a 1-bit representation of the image at higher and higher levels of "elevation": a bit is turned on if the image pixel has a nonzero value that is at or above that bit level.



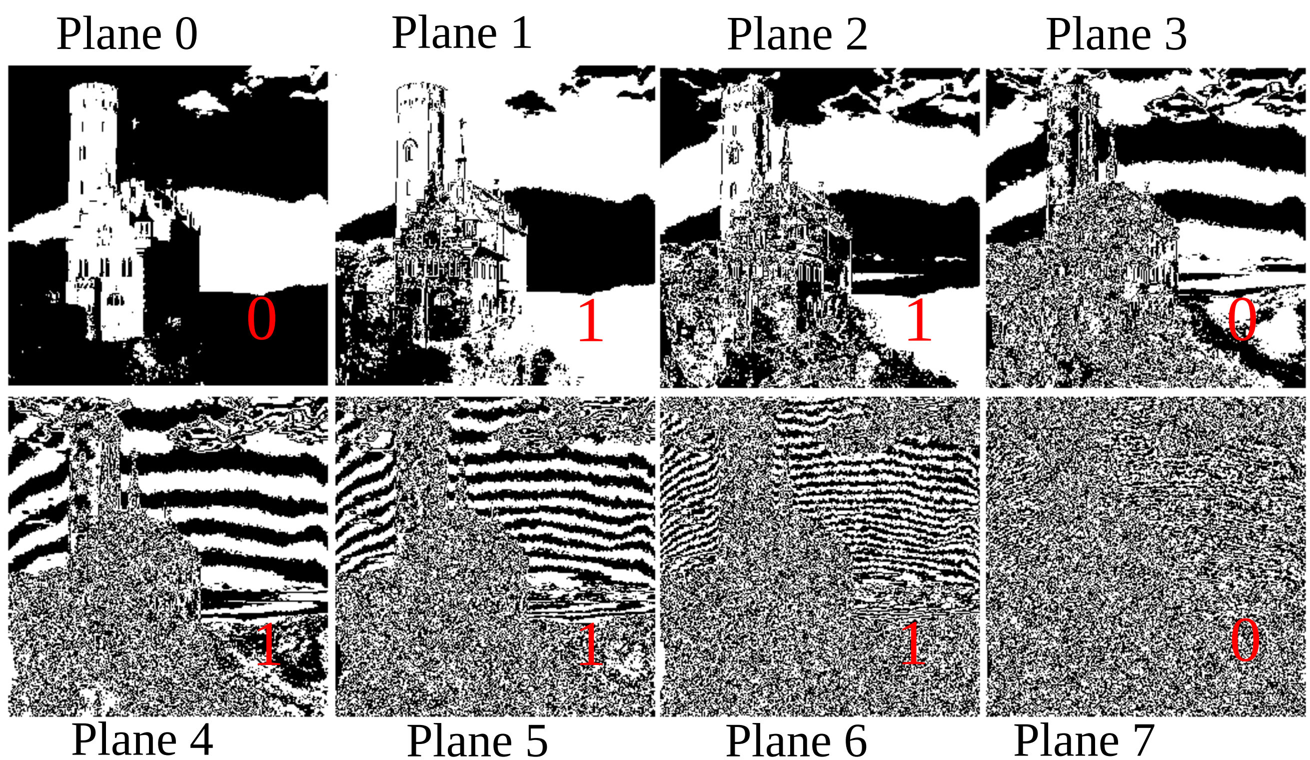
$$181=1*2^7+0*2^6+1*2^5+1*2^4+0*2^3+1*2^2+0*2^1+1*2^0$$

181(in decimal)→10110101(in binary)

Fig. 2.2: Bit-planes for 8-bit grayscale image.

- The first bit plane gives the **roughest** but the most **critical** approximation of values of a medium.
- The higher the number of the bit plane, the less is its contribution to the final stage. Thus, adding a bit plane gives a better approximation.

110 (in decimal)=01101110(in binary)



- Each bit-plane can have a value of 0 or 1 at each pixel.
- All the bit-planes make up a single byte that stores values between 0 and 255 (in this 8-bit situation).
- For the least significant bit, the bit value translates to 0 or 1 in the final numeric sum of the binary number.

For a pixel with intensity value 110 in a 8-bit image

110 (in decimal)=01101110(in binary)

Bit plane	0	1	2	3	4	5	6	7
Weight	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Binary code	0	1	1	0	1	1	1	0

Multimedia Presentation

- Each pixel is usually stored as a byte (a value between 0 to 255), so a 640 x 480 grayscale image requires 300 kB of storage (640 x 480 = 307, 200).
- Fig. 2.3 shows the Lena image again, but this time in grayscale.
- When an image is printed, the basic strategy of **dithering** is used, which trades intensity resolution for spatial resolution to provide ability to print multi-level images on 2-level (1-bit) printers.

Fig. 2.3: Grayscale image of Lena.

Dithering

- **Dithering** is used to calculate patterns of dots such that values from 0 to 255 correspond to pleasing patterns that are more and more filled at darker pixel values, for printing on a 1-bit printer.

- The main strategy is to replace a pixel value by a larger pattern, say 2 x 2 or 4 x 4, such that the number of printed dots approximates the varying-sized disks of ink used in analog, in **halftone printing** (e.g., for newspaper photos).

  - Half-tone printing is an analog process that uses smaller or larger filled circles of black ink to represent shading, for newspaper printing.
  - **NxN** dither matrix provides $\mathbf{N^2+1}$ levels of intensity.

- The rule is:

  If the intensity is > the dither matrix entry then print an on dot at that entry location: replace each pixel by an n x n matrix of dots.

- Note that the image size may be much larger, for a dithered image, since replacing each pixel by a 4 x 4 array of dots, makes an image 16 times as large.

For example, if we use a 2 x 2 **dither matrix**

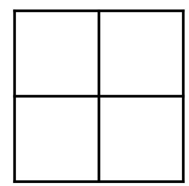
$$\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

it can support $2^2+1=5$ levels of intensity representation.

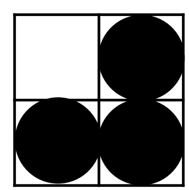
we can first re-map image values in 0..255 into the new range 0..4 by (integer) dividing by 256/5.

Then, e.g., if the pixel value is 4 we print nothing, in a 2 x 2 area of printer output. But if the pixel value is 0 we print all four dots.

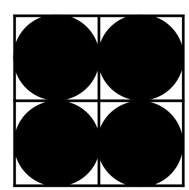
- $181 \div \left(\frac{256}{5}\right) = 3.53 \approx 4$



- $50 \div \left(\frac{256}{5}\right) = 0.98 \approx 1$



- $10 \div \left(\frac{256}{5}\right) = 0.2 \approx 0$



Exercise

- In a 8-bit grey level image, a pixel has the intensity of 80. if dithering is applied to that pixel using a 4x4 dither matrix as below. Work out the dot pattern for that pixel.

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

Solution

- Step 1: the intensity level supported by 4x4 dither matrix:

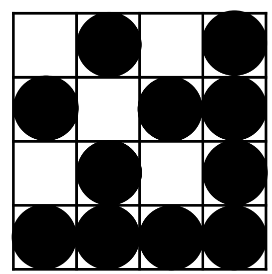
  42+1=17 levels

- Step 2: converting the intensity of the pixel from 256 levels to 17 levels.

$$80 \div \left(\frac{256}{17}\right) = 5.3125 \approx 5$$

- Step 3: comparing 5 with every entry in the matrix, if 5 is larger, print nothing (on dot) in that location, if 5 is equal to or smaller, print a black dot.

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$



A clever trick can get around this problem. Suppose we wish to use a larger, 4 x 4 dither matrix, such as

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

Procedure:

1. Image values converts from 0…255 to 0…15.
2. Slide the dither matrix over the image four pixels in the horizontal and vertical directions at a time.
3. turning on the printer out-put bit for a pixel if the intensity level is greater than the particular matrix element just at that pixel position. (called **ordered dithering**)

- Fig. 2.4 (a) shows a grayscale image of "Lena". The ordered-dither version is shown as Fig. 2.4 (b), with a detail of Lena's right eye in Fig. 2.4 (c).

An algorithm for ordered dither, with n x n dither matrix, is as follows:

BEGIN

for $x = 0$ to x_{max} // columns

for $y = 0$ to y_{max} // rows

$i = x \bmod n$

$j = y \bmod n$

// $I(x, y)$ is the input, $O(x, y)$ is the output,

// D is the dither matrix.

if $I(x, y) > D(i, j)$

$O(x, y) = 1;$

else

$O(x, y) = 0;$

END

(a) (b) (c)

Fig. 2.4: Dithering of grayscale images.

(a): 8-bit grey image "lenagray.bmp". (b): Dithered version of the image (1-bit). (c): Detail of dithered version.

Colour Image Data Types

- The most common data types for graphics and image file formats — 24-bit color and 8-bit color.
- Some formats are restricted to particular hardware / operating system platforms, while others are "cross-platform" formats.
- Even if some formats are not cross-platform, there are conversion applications that will recognize and translate formats from one system to another.
- Most image formats incorporate some variation of a **compression** technique due to the large storage size of image files. Compression techniques can be classified into either **lossless** or **lossy**.

24-bit Color Images

- In a 24-bit color image, each pixel is represented by <u>three bytes</u>, usually representing RGB.
  - This format supports 256 x 256 x 256 possible combined colors, or a total of 16,777,216 possible colors.
  - However such flexibility does result in a storage penalty: A 640 x 480 24-bit color image would require 921.6 kB of storage without any compression.

- **An important point**: many 24-bit color images are actually stored as 32-bit images, with the extra byte of data for each pixel used to store an *alpha* value representing special effect information (e.g., transparency).

- Fig. 2.5 shows the image **forestfire.bmp**, a 24-bit image in Microsoft Windows BMP format. Also shown are the grayscale images for just the Red, Green, and Blue channels, for this image.


(a)


(b)


(c)


(d)

Fig. 2.5: High-resolution color and separate R, G, B color channel images. (a): Example of 24-bit color image “forestfire.bmp”. (b, c, d): R, G, and B color channels for this image

8-bit Color Images

- Many systems can make use of 8 bits of color information (the so-called "256 colors") in producing a screen image.

- Such image files use the concept of a **lookup table** to store color information.

  - Basically, the image stores not color, but instead just a set of bytes, each of which is actually an index into a table with 3-byte values that specify the color for a pixel with that lookup table index.
  - The color is stored in lookup table

Fig. 2.7 shows the resulting 8-bit image, in GIF format.

Fig. 2.7 Example of 8-bit color image.

- Note the great savings in space for 8-bit images, over 24-bit ones:

a 640 x 480 8-bit color image only requires 300 kB of storage, compared to 921.6 kB for a color image (again, without any compression applied).

Color Look-up Tables (LUTs)

LUT for Image

- The idea used in 8-bit color images is to store only the index, or code value, for each pixel.
  - e.g., if a pixel stores the value 25, the meaning is to go to row 25 in a color look-up table (LUT).

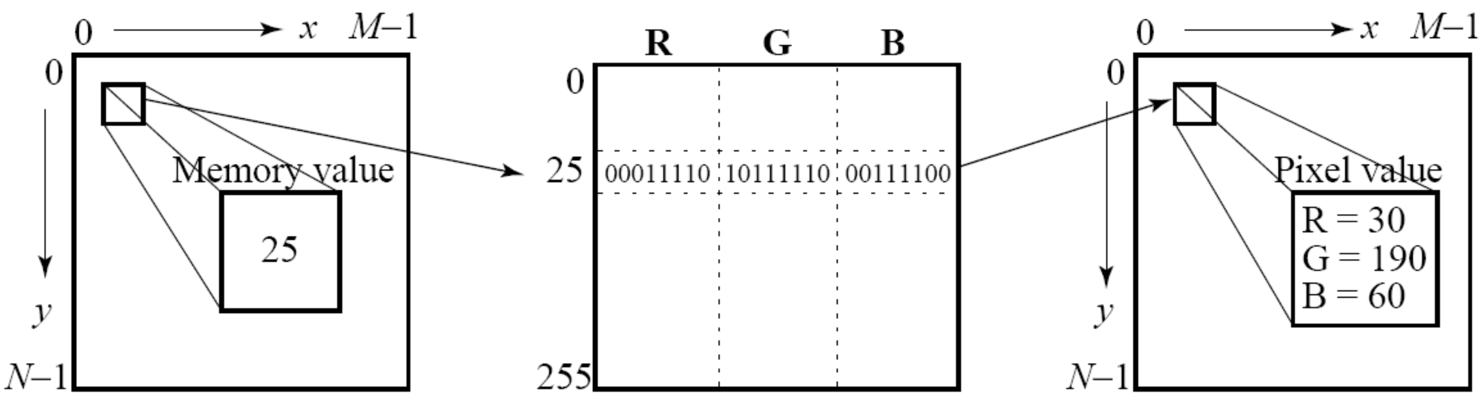
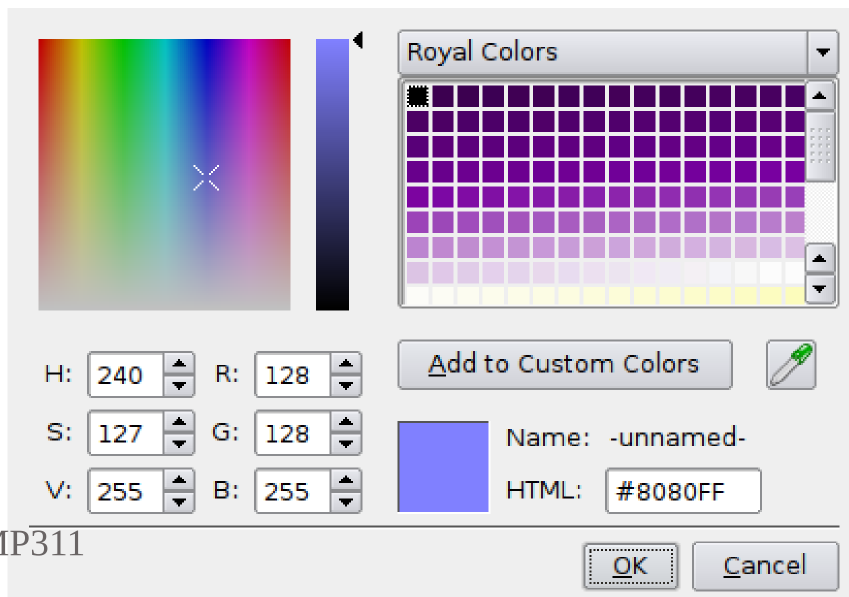


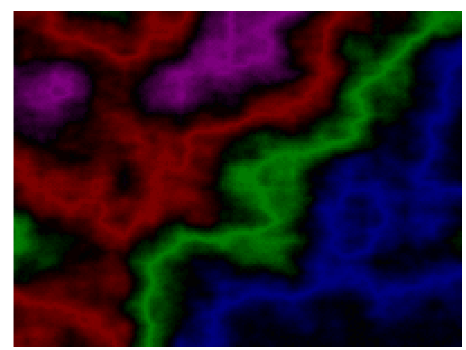
Fig. 2.8: Color LUT for 8-bit color images.

LUT for Color-picker

- A **Color-picker** consists of an array of fairly large blocks of color (or a semi-continuous range of colors) such that a mouse-click will select the color indicated.
  - In reality, a color-picker displays the palette colors associated with index values from 0 to 255.
  - Fig. 2.9 displays the concept of a color-picker: if the user selects the color block with index value 2, then the color meant is cyan, with RGB values (0, 255, 255).
- A very simple animation process is possible via simply changing the color table: this is called **color cycling** or **palette animation**.

Color picker





color cycling

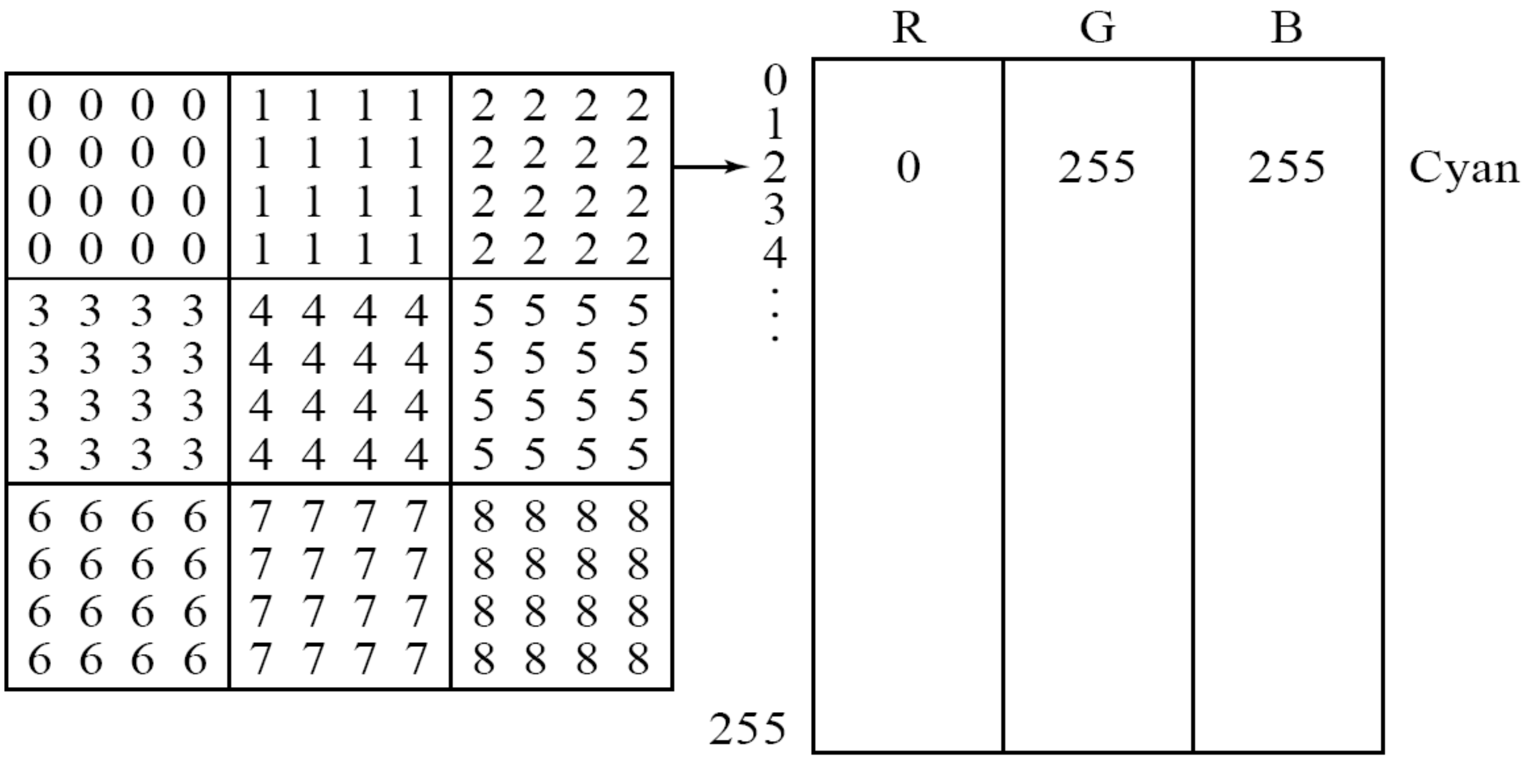


Fig. 2.9: Color-picker for 8-bit color: each block of the color-picker corresponds to one row of the color LUT

- Fig. 2.10 (a) shows a 24-bit color image of "Lena", and Fig. 2.10 (b) shows the same image reduced to only **5 bits** via dithering. A detail of the left eye is shown in Fig. 2.10 (c).

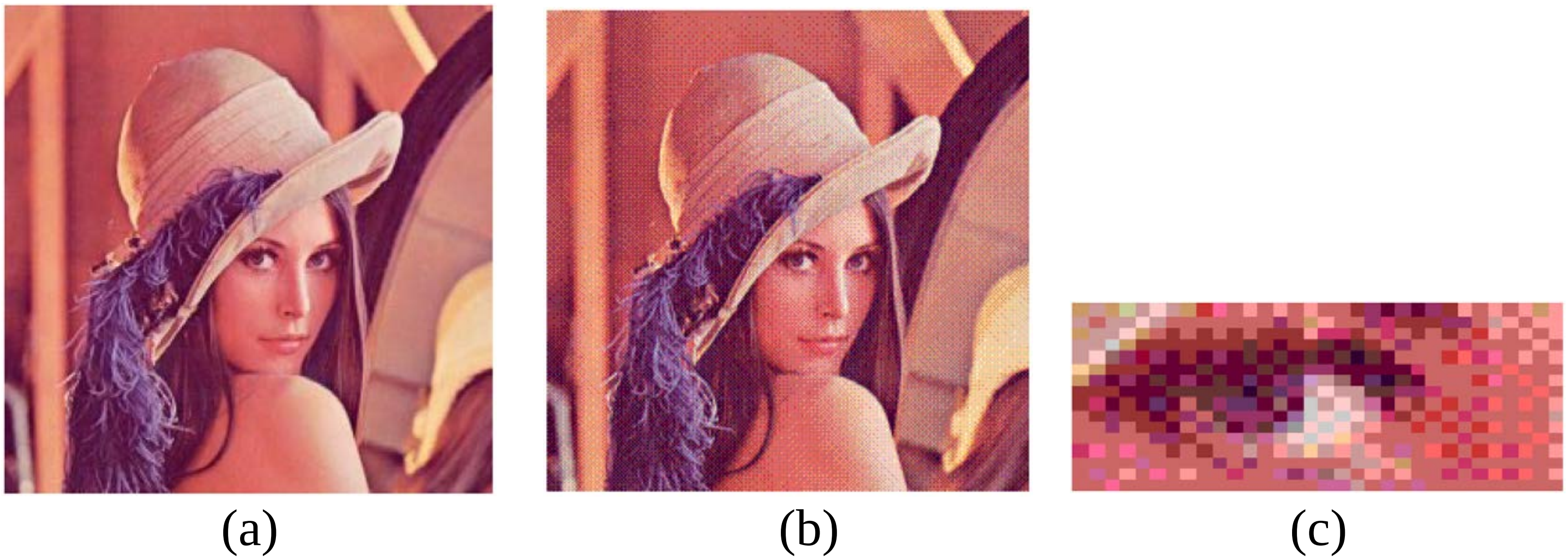
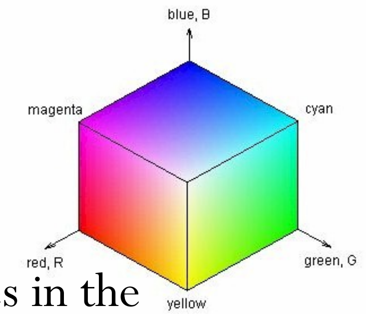


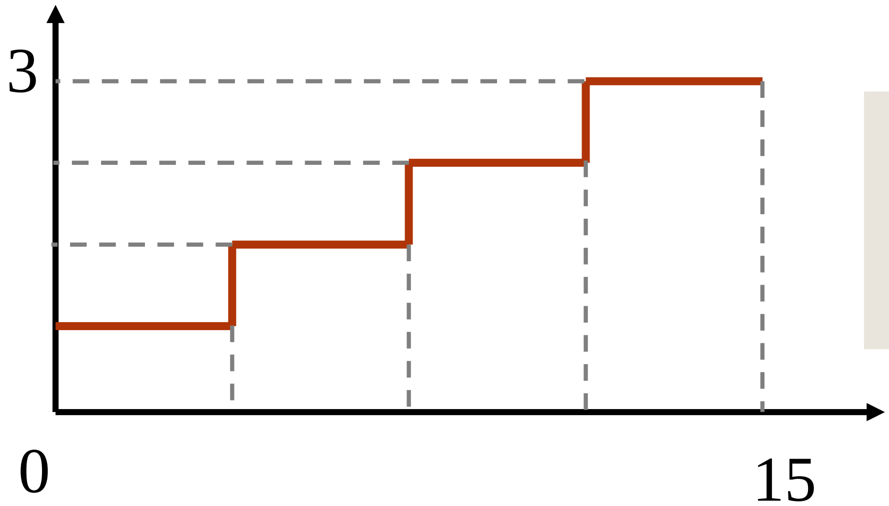
Fig. 2.10: (a): 24-bit color image "lena.bmp". (b): Version with color dithering. (c): Detail of dithered version.

How to devise a color look-up table



To make 8-bit look-up color out of 24-bit color, the most straightforward approach:

1. Divide the RGB cube into equal slices in each dimension.
2. The centers of each of the resulting cubes would serve as the entries in the color LUT, while simply scaling the RGB ranges 0..255 into the appropriate ranges would generate the 8-bit codes.



Example 0…15→0…3 (4-bit→2-bit)

Shades of R=Shades of G=Shades of B=A

$$A \times A \times A = 2^8$$
$$A = \sqrt[3]{256} \approx 6$$

How to devise a color look-up table

- To take the fact that humans are more sensitive to R and G than to B, the second approach is:

1. To shrink the R range and G range 0..255 into the 3-bit range 0..7 and shrink the B range down to the 2-bit range 0..3, thus making up a total of 8 bits. To shrink R and G, we could simply divide the R or G byte value by (256/8)=32 and then truncate.
2. Then each pixel in the image gets replaced by its 8-bit index and the color LUT serves to generate 24-bit color.

Shades of R=8
Shades of G=8
Shades of B=4

Third Approach with better performance: **Median-cut** algorithm

- This type of scheme will indeed concentrate bits where they most need to differentiate between high populations of close colors.
- Steps:
  1. (1-bit labelling) Sort the R values and find median R; label pixels smaller than the median R with a "0" bit and pixel larger than the median with a "1" bit. (The median is the point where half the pixels are smaller and half are larger.)
  2. (2-bit labelling) For pixels with 0 from step **1**, sort their G values and label the pixels with another "0" bit and "1" bit following the same scheme. For pixels with 1 from step 1, sort their G values and label the pixels with another "0" bit and "1" bit following the same scheme.
  3. (3-bit labelling) For pixels with 0 from step **2**, sort their B values and label the pixels with another "0" bit and "1" bit following the same scheme. For pixels with 1 from step **2**, sort their B values and label the pixels with another "0" bit and "1" bit following the same scheme.
  4. Repeating all steps, R, G and B results in a 6-bit scheme, and cycling through R and G once more results in 8 bits.

One can most easily visualize finding the median by using a histogram showing counts at position 0..255.

Fig. 2.11 shows a histogram of the R byte values for the `forestfire.bmp` image along with the median of these values, shown as a vertical line.

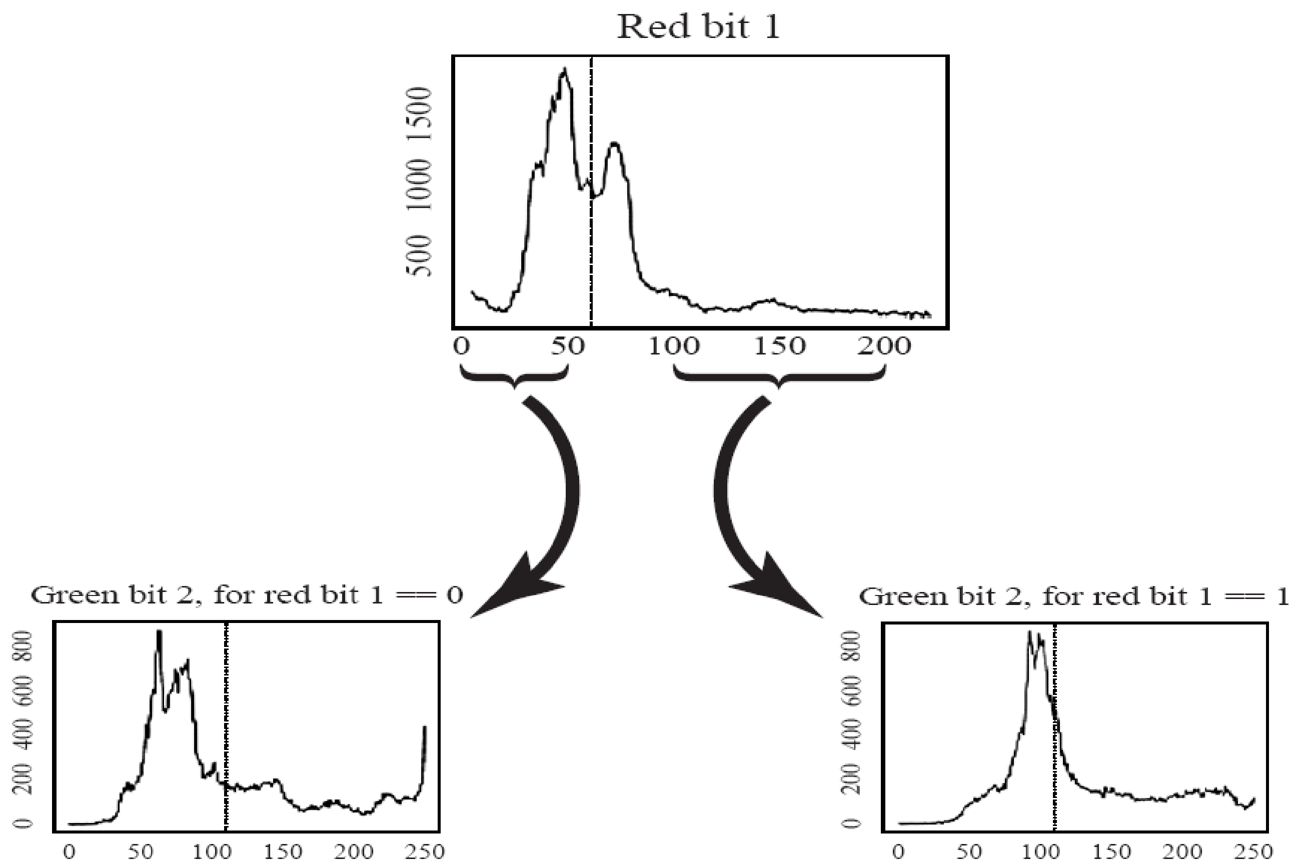


Fig. 2.11 Histogram of R bytes for the 24-bit color image "forestfire.bmp" results in a "0" bit or "1" bit label for every pixel. For the second bit of the color table index being built, we take R values less than the R median and label just those pixels as "0" or "1" according as their G value is less than or greater than the median of the G value, just for the "0" Red bit pixels. Continuing over R, G, B for 8 bits gives a color LUT 8-bit index.

Popular File Formats

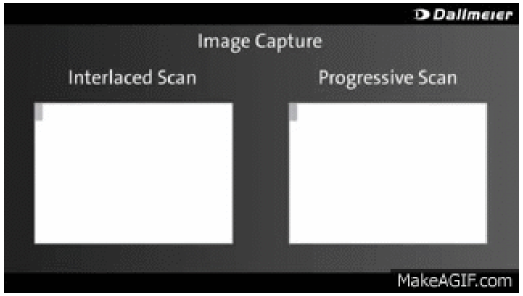
- **8-bit GIF** : one of the most important formats because of its historical connection to the WWW and HTML markup language as the first image type recognized by net browsers.

- **JPEG**: currently the most important common file format.

GIF

- **GIF standard**: (We examine GIF standard because it is so simple! yet contains many common elements.)
- Limited to *8-bit (256)* color images only, which, while producing acceptable color images, is best suited for images with few distinctive colors (e.g., graphics or drawing).

- GIF standard supports *interlacing* — successive display of pixels in widely-spaced rows by a 4-pass display process.



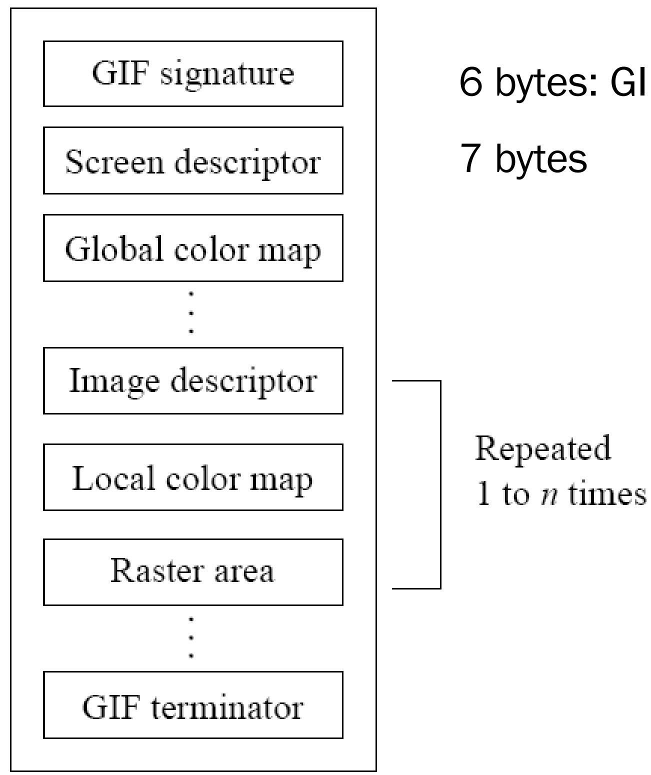
- GIF actually comes in two flavors:

1. **GIF87a**: The original specification.
2. **GIF89a**: The later version. Supports *simple animation* via a Graphics Control Extension block in the data, provides simple control over *delay time*, a *transparency index*, etc.

GIF87

- For the standard specification, the general file format of a GIF87 file is as in Fig. 2.12.

A file can contain >=1 image(s), each can have their only local color map. If not, a global color map is defined.



6 bytes: GIF87a

7 bytes

Fig. 2.12: GIF file format.

- **Screen Descriptor** comprises a set of attributes that belong to every image in the file. According to the GIF87 standard, it is defined as in Fig. 2.13.

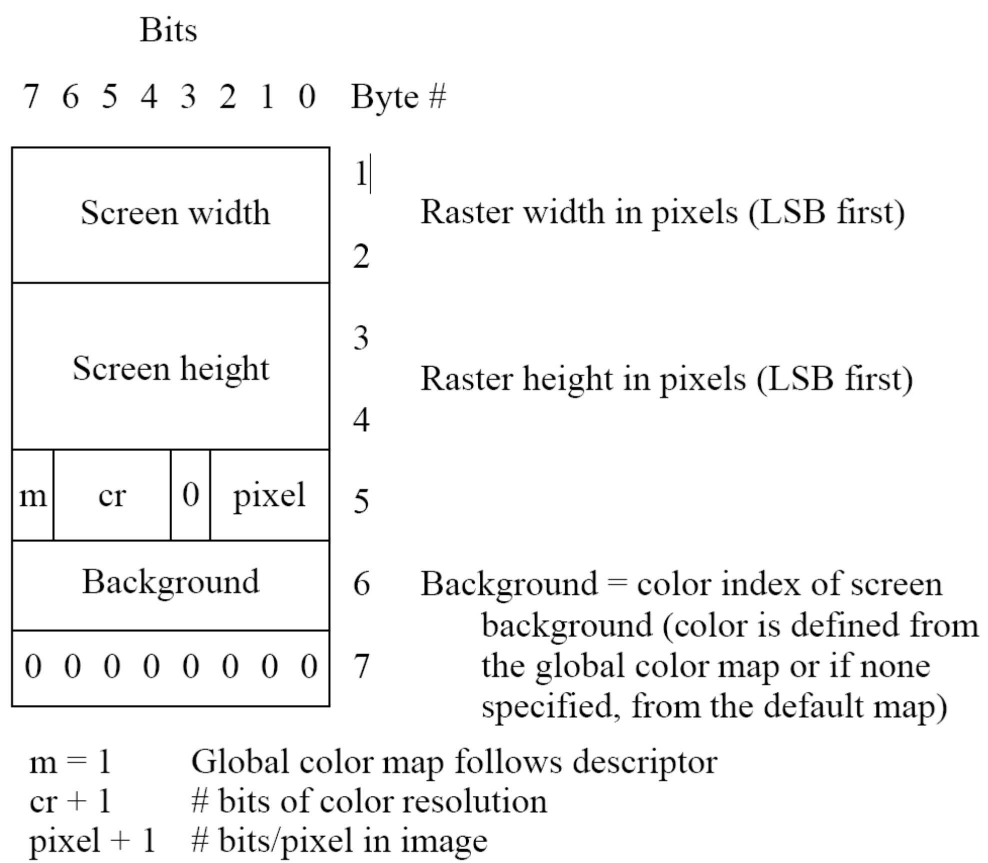


Fig. 2.13: GIF screen descriptor.

- **Color Map** is set up in a very simple fashion as in Fig. 2.14. However, the actual length of the table equals $2^{(pixel+1)}$ as given in the Screen Descriptor.

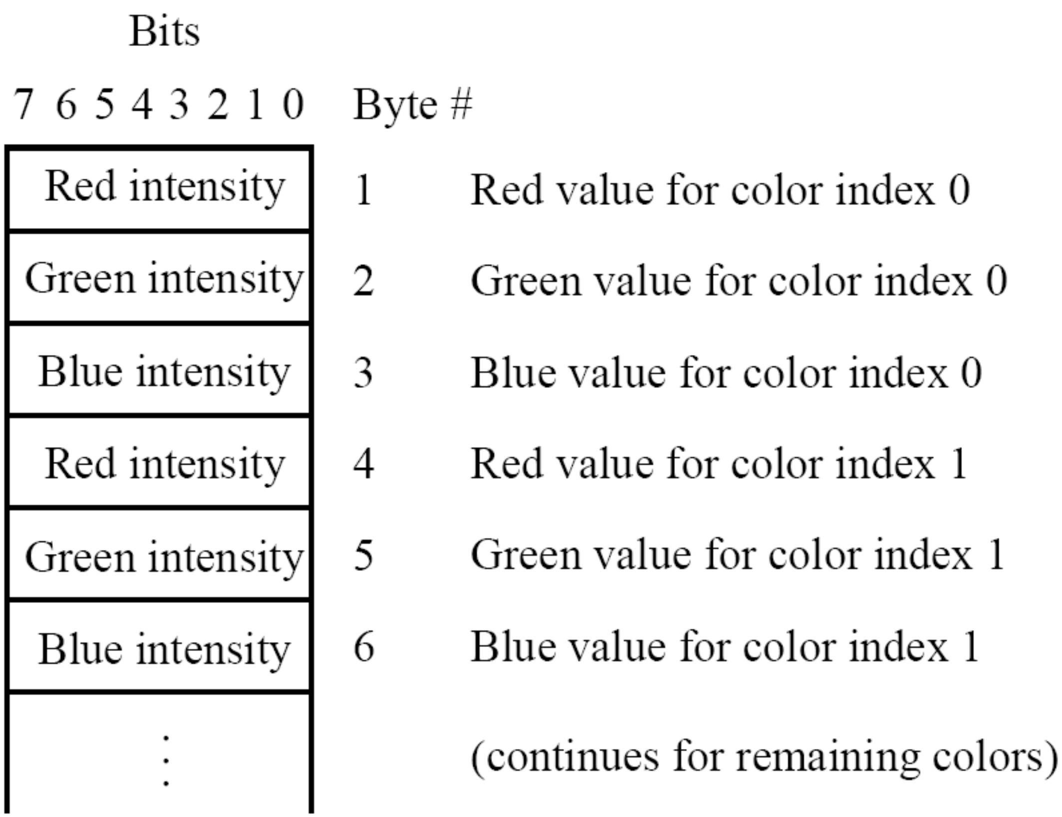


Fig. 2.14: GIF color map.

- Each image in the file has its own **Image Descriptor**, defined as in Fig. 2.15.

Bits (7 6 5 4 3 2 1 0)	Byte #	
0 0 1 0 1 1 0 0	1	Image separator character (comma)
Image left	2 3	Start of image in pixels from the left side of the screen (LSB first)
Image top	4 5	Start of image in pixels from the top of the screen (LSB first)
Image width	6 7	Width of the image in pixels (LSB first)
Image height	8 9	Height of the image in pixels (LSB first)
m | i | 0 | 0 | 0 | pixel	10	m = 0 Use global color map, ignore 'pixel' m = 1 Local color map follows, use 'pixel' i = 0 Image formatted in Sequential order i = 1 Image formatted in Interlaced order pixel + 1 # bits per pixel for this image

Fig. 2.15: GIF image descriptor.

- If the "interlace" bit is set in the local Image Descriptor, then the rows of the image are displayed in a four-pass sequence (Fig.2.16).

Image row	Pass 1	Pass 2	Pass 3	Pass 4	Result
0	*1a*				*1a*
1				*4a*	*4a*
2			*3a*		*3a*
3				*4b*	*4b*
4		*2a*			*2a*
5				*4c*	*4c*
6			*3b*		*3b*
7				*4d*	*4d*
8	*1b*				*1b*
9				*4e*	*4e*
10			*3c*		*3c*
11				*4f*	*4f*
12		*2b*			*2b*
⋮					

Fig. 2.16: GIF 4-pass interlace display row order.

- We can investigate how the file header works in practice by having a look at a particular GIF image. Fig. 2.7 on page is an 8-bit color GIF image, in UNIX, issue the command:

```
od -c forestfire.gif | head -2
```

and we see the first 32 bytes interpreted as characters:

```
G  I  F  8  7  a  \208  \2  \188  \1  \247  \0  \0  \6  \3  \5
J  \132  \24  |  )  \7  \198  \195  \  \128  U  \27  \196  \166  &  T
```

- To decipher the remainder of the file header (after "GIF87a"), we use hexadecimal:

```
od -x forestfire.gif | head -2
```

with the result

```
4749 4638 3761 d002 bc01 f700 0006 0305 ae84 187c 2907 c6c3 5c80
551b c4a6 2654
```

JPEG

- **JPEG**: The most important current standard for image compression.

- The human vision system has some specific limitations and JPEG takes advantage of these to achieve high rates of compression.

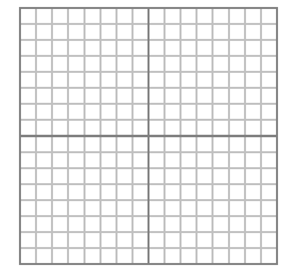
- JPEG allows the user to set a desired level of quality, or compression ratio (input divided by output).

- As an example, Fig. 2.17 shows our **forestfire** image, with a quality factor Q=10%.
  - - This image is a mere 1.5% of the original size. In comparison, a JPEG image with Q=75% yields an image size 5.6% of the original, whereas a GIF version of this image compresses down to 23.0% of uncompressed image size.

Fig. 2.17: JPEG image with low quality specified by user.

PNG

- **PNG format**: standing for **Portable Network Graphics** — meant to supersede the GIF standard, and extends it in important ways.



- Special features of PNG files include:

1. Support for up to 48 bits of color information — a large increase.

2. Files may contain *gamma-correction* information for correct display of color images, as well as *alpha-channel* information for such uses as control of transparency.

3. The display progressively displays pixels in a 2-dimensional fashion by showing a few pixels at a time over seven passes through each 8 x 8 block of an image.

TIFF

- **TIFF**: stands for **Tagged Image File Format**.

- The support for attachment of additional information (referred to as "*tags*") provides a great deal of flexibility.

1. The most important tag is a *format signifier*: what type of compression etc. is in use in the stored image.

2. TIFF can store many different types of image: 1-bit, grayscale, 8-bit color, 24-bit RGB, etc.

3. TIFF was originally a lossless format but now a new *JPEG tag* allows one to opt for JPEG compression.

4. The TIFF format was developed by the Aldus Corporation in the 1980's and was later supported by Microsoft.

EXIF

- **EXIF** (**Exchange Image File**) is an image format for digital cameras:
  - 1. Compressed EXIF files use the *baseline JPEG format*.
  - 2. A variety of tags (many more than in TIFF) are available to facilitate higher quality printing, since information about the camera and picture-taking conditions (flash, exposure, light source, white balance, type of scene, etc.) can be stored and used by printers for possible color correction algorithms.
  - 3. The EXIF standard also includes specification of file format for *audio* that accompanies digital images. As well, it also supports tags for information needed for conversion to FlashPix (initially developed by Kodak).

Graphics Animation Files

- A few dominant formats aimed at storing graphics animations (i.e., series of drawings or graphic illustrations) as opposed to video (i.e., series of images).

- **Difference**: animations are considerably less demanding of resources than video files.

  - 1. FLC is an animation or moving picture file format; it was originally created by Animation Pro. Another format, FLI, is similar to FLC.

  - 2. GL produces somewhat better quality moving pictures. GL animations can also usually handle larger file sizes.

  - 3. Many older formats: such as DL or Amiga IFF files, Apple Quicktime files, as well as animated GIF89 files.

PS and PDF

- **Postscript** is an important language for typesetting, and many high-end printers have a Postscript interpreter built into them.

- Postscript is a *vector-based* picture language, rather than pixel-based: page element definitions are essentially in terms of vectors.

  1. Postscript includes text as well as vector/structured graphics.

  2. GL bit-mapped images can be included in output files.

  3. Encapsulated Postscript files add some additional information for inclusion of Postscript files in another document.

4. Postscript page description language itself does not provide compression; in fact, Postscript files are just stored as ASCII.

- Another text + figures language : Adobe Systems Inc. includes LZW compression in its Portable Document Format (**PDF**) file format.
  - PDF files that do not include images have about the same compression ratio, 2:1 or 3:1, as do files compressed with other LZW-based compression tools.

Some Other Formats

- **Microsoft Windows: WMF**: the native vector file format for the Microsoft Windows operating environment:

  1. Consist of a collection of GDI (Graphics Device Interface) function calls, also native to the Windows environment.

  2. When a WMF file is "played" (typically using the Windows PlayMetaFile() function) the described graphics is rendered.

  3. WMF files are ostensibly device-independent and are unlimited in size.

- **Microsoft Windows: BMP**: the major system standard graphics file format for Microsoft Windows, used in Microsoft Paint and other programs. Many sub-variants within the BMP standard.

- **Macintosh: PAINT and PICT**:

  1. PAINT was originally used in the MacPaint program, initially only for 1-bit monochrome images.

  2. PICT format is used in MacDraw (a vector-based drawing program) for storing structured graphics.

Further Exploration

- More information including a complete up-to-date list of current file formats can be viewed on http://www.cs.sfu.ca/mmbook/
- Other links include:
  - GIF87 and GIF89 details. Again, these file formats are not so interesting in themselves, but they have the virtue of being simple and a useful introduction to how such bitstreams are set out.
  - A shareware program (that is consequently very popular) for developing GIF animations.
  - JPEG considered in detail.
  - PNG details.
  - The PDF file format.
  - The ubiquitous BMP file format.

Color

- Color Science
- Color Models
- Further Exploration

Color Science

- **Light and Spectra**
  - Light is an electromagnetic wave. Its color is characterized by the wavelength content of the light.
  - Laser light consists of a single wavelength: e.g., a ruby laser produces a bright, scarlet-red beam.
  - Most light sources produce contributions over many wavelengths.
  - Human cannot detect all light, just contributions that fall in the "visible wavelengths".
  - Short wavelengths produce a blue sensation, long wavelengths produce a red one.

Color Science

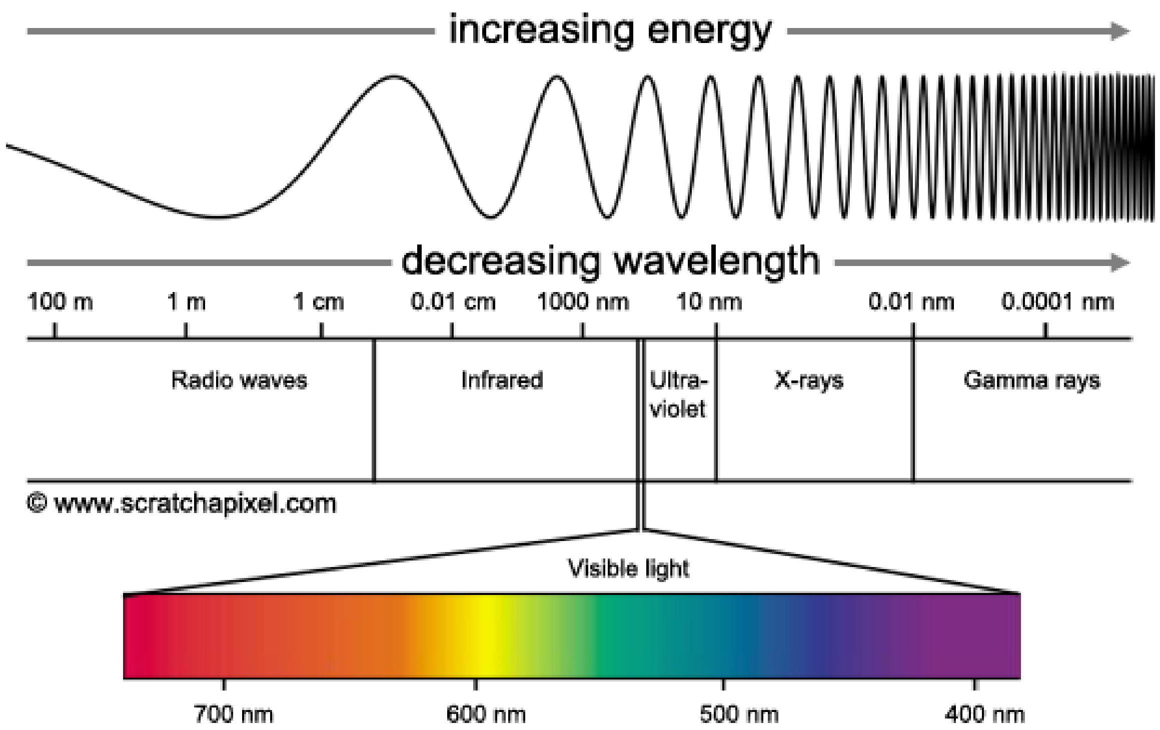


Fig. 3.1: the spectra

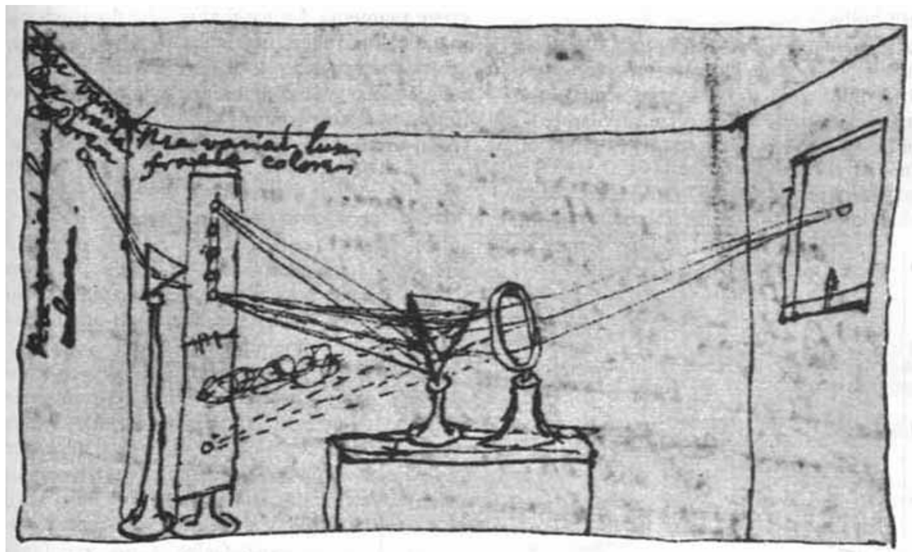


Fig. 3.2: Sir Isaac Newton's experiments.

- Visible light is an electromagnetic wave in the range *400 nm* to *700 nm* (where nm stands for nanometer, 10^{-9} meters).

- Spectrophotometer: a device used to measure visible light, by reflecting light from a diffraction grating (a ruled surface) that spreads out the different wavelengths.
- Figure 3.3 shows the phenomenon that white light contains all the colors of a rainbow and how spectrophotometer works.

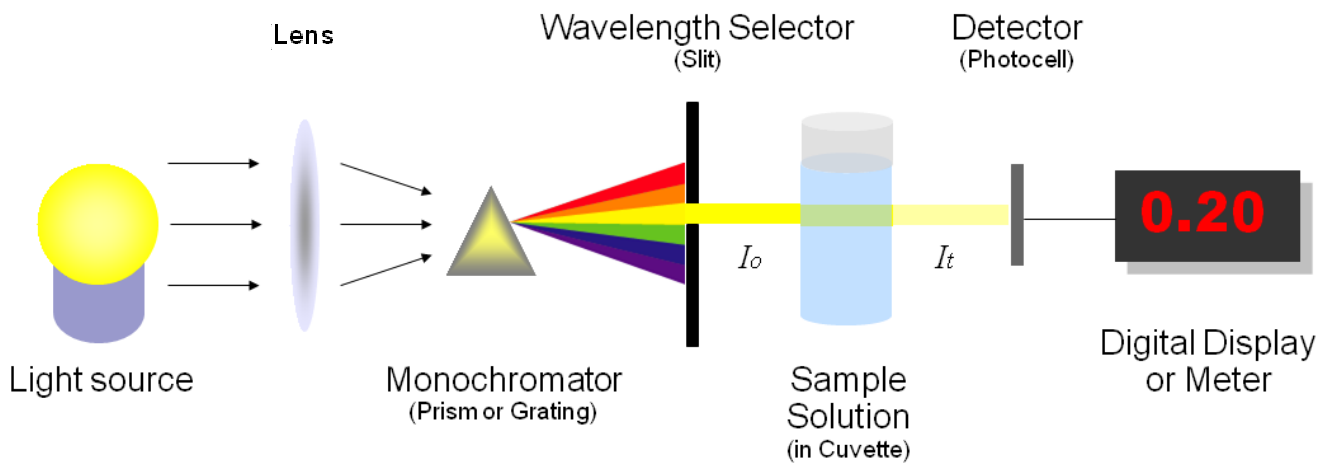


Fig. 3.3: how spectrophotometer works

- Fig. 3.4 shows the relative power in each wavelength interval for typical outdoor light on a sunny day. This type of curve is called a **Spectral Power Distribution** (**SPD**) or a **spectrum**.
- The symbol for wavelength is λ. This curve is called E(λ).

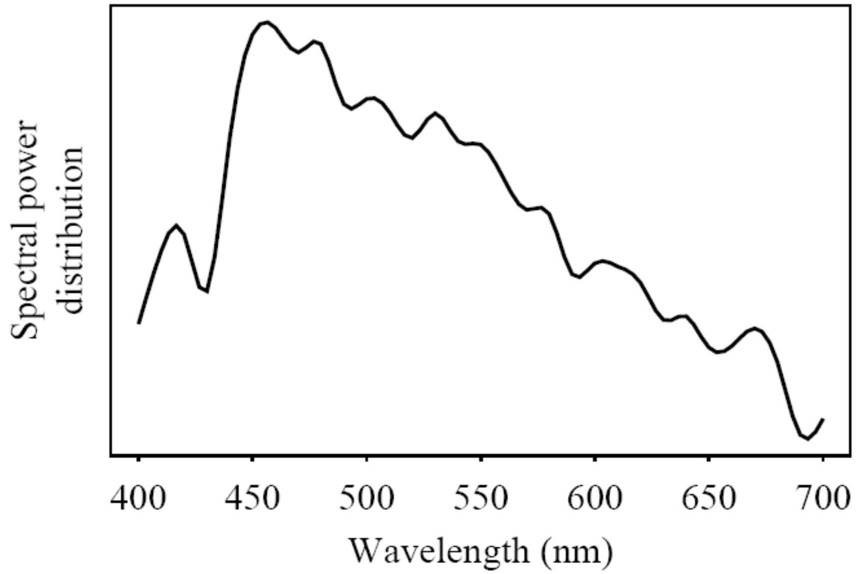


Fig. 3.4: Spectral power distribution of daylight.

Human Vision

- The eye works like a camera, with the lens focusing an image onto the retina (upside-down and left-right reversed).
- The retina consists of an *array* of *rods* and three kinds of *cones*.

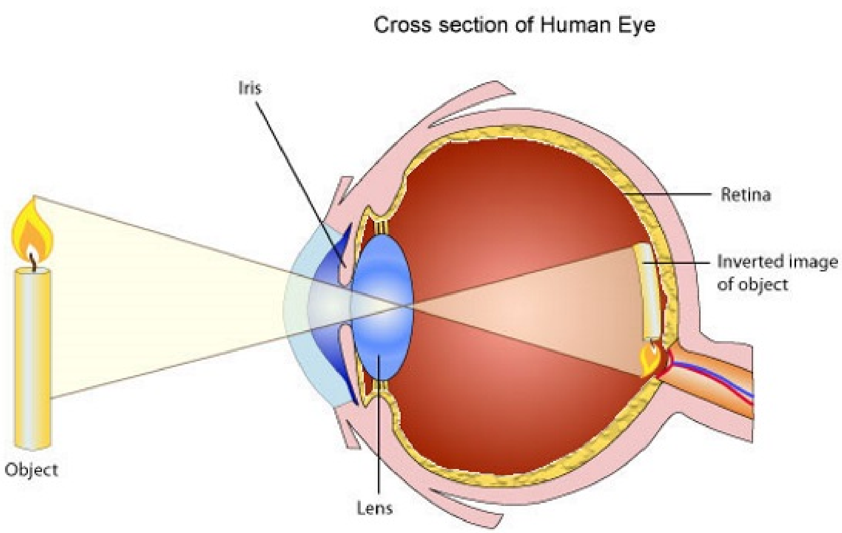


Fig. 3.4: Cross section of human eye

Human Vision

- The rods come into play when light levels are low and produce a image in shades of gray.

- For higher light levels, the cones each produce a signal. Because of their differing pigments, the three kinds of cones are most sensitive to red (R), green (G), and blue (B) light.

- The brain makes use of differences R-G, G-B, and B-R, as well as combining all of R, G, and B into a high-light-level image.

Spectral Sensitivity of the Eye

- The eye is most sensitive to light *in the middle* of the visible spectrum.

- The sensitivity of our *receptors* is also a function of wavelength (Fig. 3.5 below).

- The Blue receptor sensitivity is not shown to scale because it is much smaller than the curves for Red or Green — Blue is a late addition, in evolution.
  - Statistically, Blue is the favorite color of humans, regardless of nationality — perhaps for this reason: Blue is a latecomer.

- Fig. 3.5 shows the overall sensitivity as a dashed line — this important curve is called the luminous-efficiency function.
  - It is usually denoted $V(\lambda)$ and is formed as the sum of the response curves for Red, Green, and Blue.

- The rod sensitivity curve looks like the luminous-efficiency function V (λ).

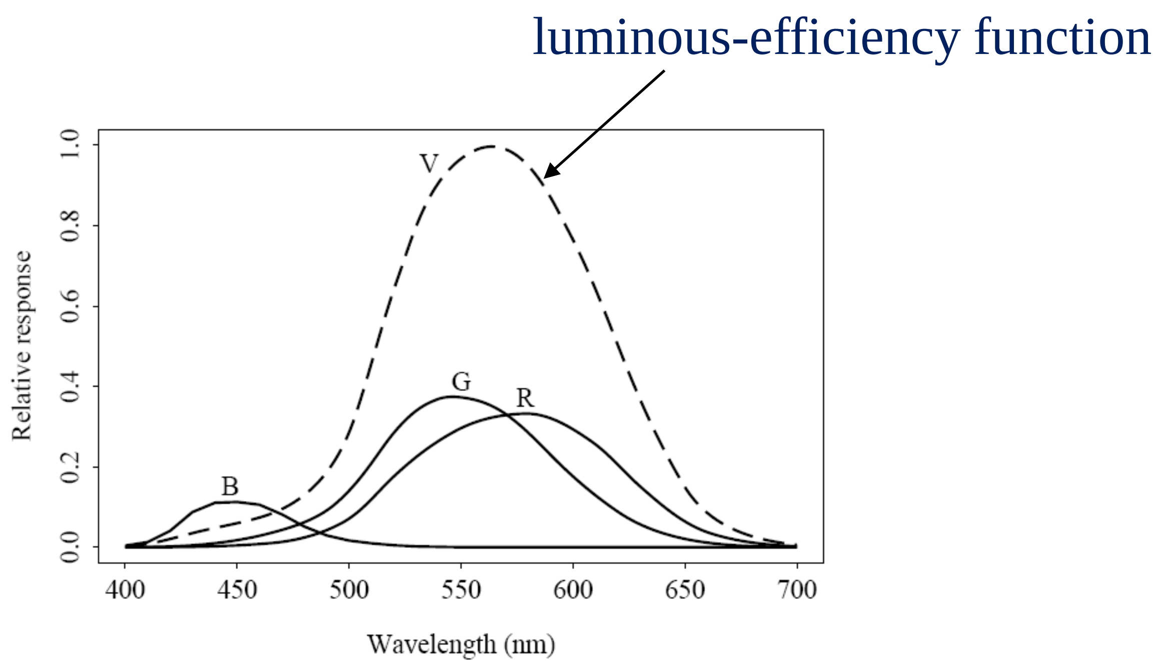


Fig. 3.5: Cone sensitivities: R, G and B cones, and Luminous Efficiency curve V(λ).

- These **spectral sensitivity functions** are usually denoted by letters other than "R,G,B"; here let's use a vector function $q(\lambda)$, with components

$$q\,(\lambda) = (q_R(\lambda), q_G(\lambda), q_B(\lambda))^T \tag{3.1}$$

- The response in each color channel in the eye is proportional to the number of neurons firing.

- A laser light at wavelength λ would result in a certain number of neurons firing. An SPD is a combination of single-frequency lights (like "lasers"), so we add up the light power for all wavelengths, weighted by the eye's relative response at that wavelength.

- We can succinctly write down this idea in the form of an integral
- Applied only when we view *self-luminous* objects

$$R = \int E(\lambda)\, q_R(\lambda)\, d\lambda$$

$$G = \int E(\lambda)\, q_G(\lambda)\, d\lambda$$

$$B = \int E(\lambda)\, q_B(\lambda)\, d\lambda \qquad (3.2)$$

SPD: Energy of light with wavelength Lamda

B cone sensitivity of light with wavelength Lamda

Image Formation

- Surfaces reflect different amounts of light at different wavelengths, and dark surfaces reflect less energy than light surfaces.

- Fig. 3.4 shows the surface spectral reflectance from (1) orange sneakers and (2) faded blue jeans. The reflectance function is denoted $S(\lambda)$.

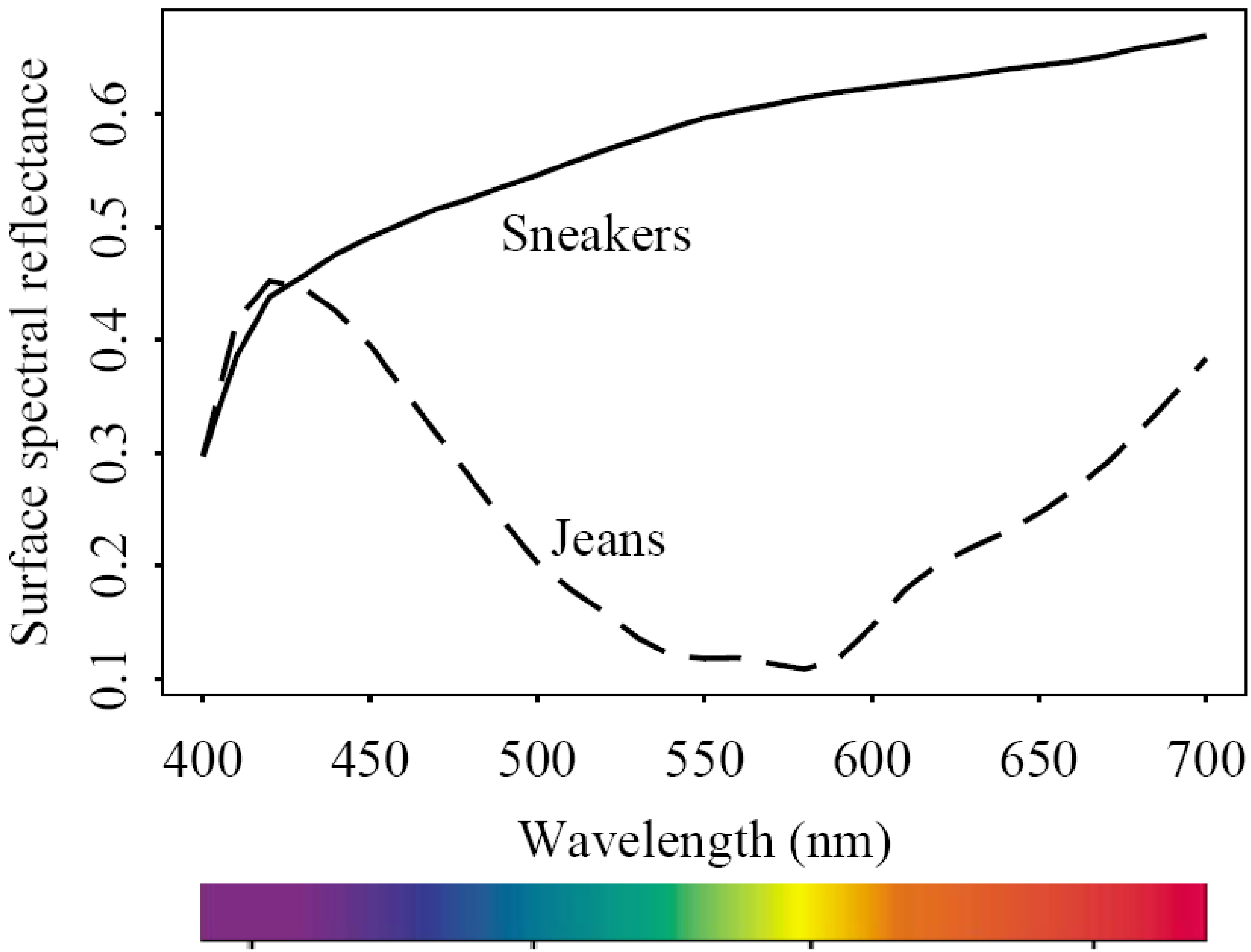


Fig. 3.6: Surface spectral reflectance functions $S(\lambda)$ for objects.

- Image formation is thus:
  - Light from the illuminant with SPD $E(\lambda)$ impinges on a surface, with surface spectral reflectance function $S(\lambda)$, is reflected, and then is filtered by the eye's cone functions $q\ (\lambda)$.
  - Reflection is shown in Fig. 3.7 below.
  - The function $C(\lambda)$ is called the color signal and consists of the product of $E(\lambda)$, the illuminant, times $S(\lambda)$, the reflectance:

$$C(\lambda) = E(\lambda)\ S(\lambda).$$

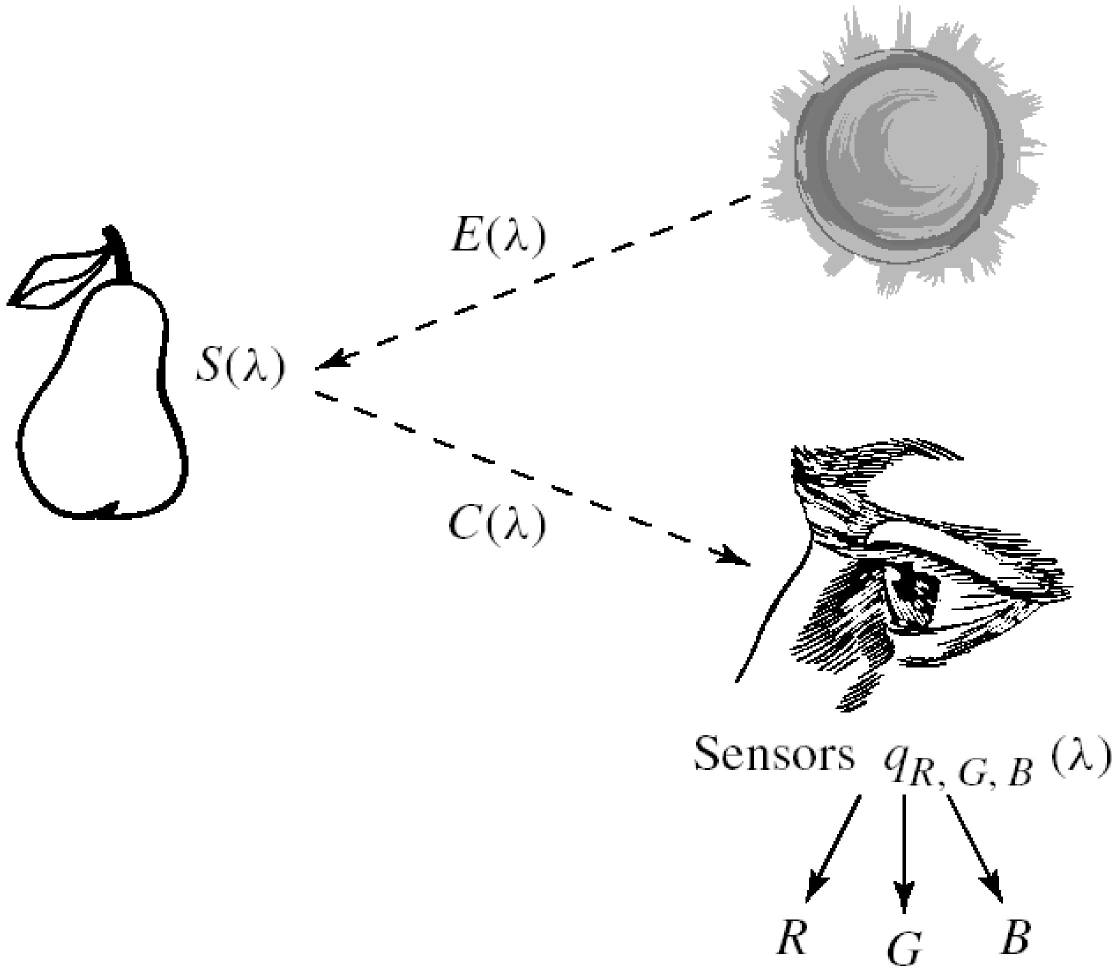


Fig. 3.7: Image formation model.

- The equations that take into account the image formation model are:

$$R = \int E(\lambda)\, S(\lambda)\, q_R(\lambda)\, d\lambda$$

$$G = \int E(\lambda)\, S(\lambda)\, q_G(\lambda)\, d\lambda$$

$$B = \int E(\lambda)\, S(\lambda)\, q_B(\lambda)\, d\lambda \qquad (3.3)$$

Color Models

- Color Space
  - A way of representing colors, usually three dimensional.
  - Examples: RGB, CMYK, HSB, HSL
- Color Models
  - A way of defining colors mathematically.
  - Examples: RGB, CMYK
- Color Gamut
  - A certain complete subset of colors.
  - Entire range of colors available for a particular devices

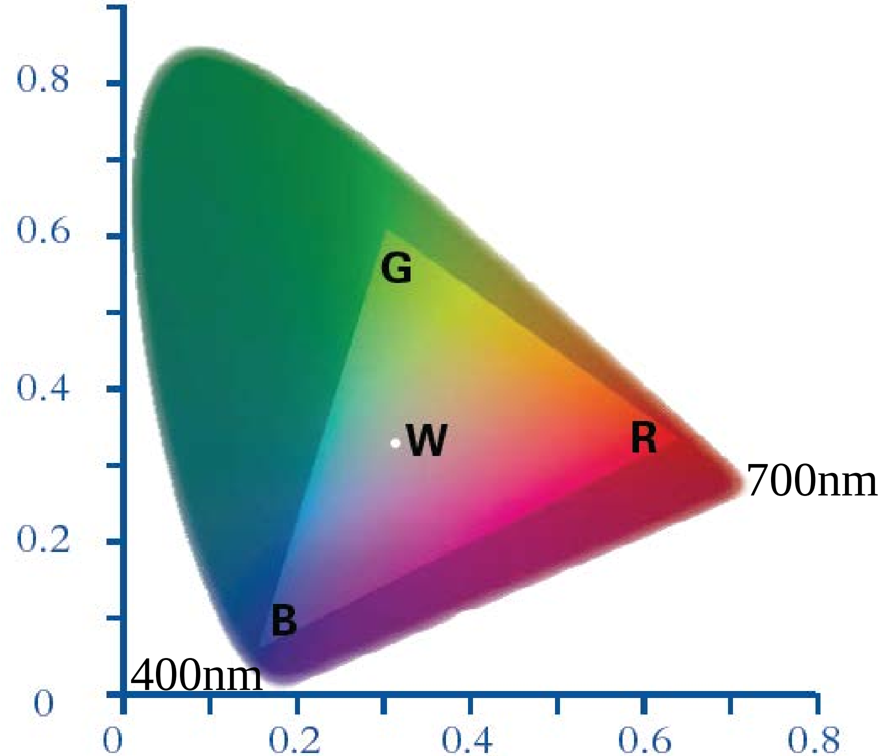
Two Major Types of Color Models

- Additive Color (adding light)
  - Describes the situation where color is created by mixing the visible light emitted from differently colored light sources.
  - Computer monitors and televisions are the most common form of additive light.

- Subtractive Color (subtracting light)
  - Light is removed from various part of the visible spectrum to create colors.
  - Used in paints and pigments and color filters.
    - Example: Subtracts blue from white illumination will reflects red and green, which is yellow.

RGB Color Model

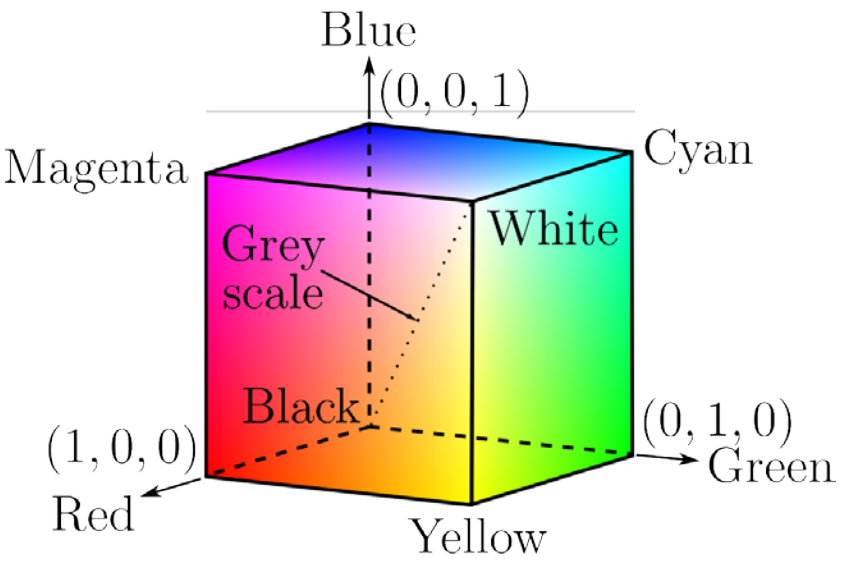
- The RGB color model is an additive primaries in which red, green, and blue light is added together in various ways to reproduce a broad array of colors.
- In RGB color, the three primaries are standard shades of red, green and blue.
- Only colors in the RGB gamut can be represented in this way.



The RGB colour gamut

RGB Color Model

- Any color is specified as three values (*R, G, B*) *giving the relative proportions of the three* primaries.
- This is often written as a 6-digit hexadecimal number, with *R, G and B each* being between 0 and 255, so a color value occupies 24 bits.



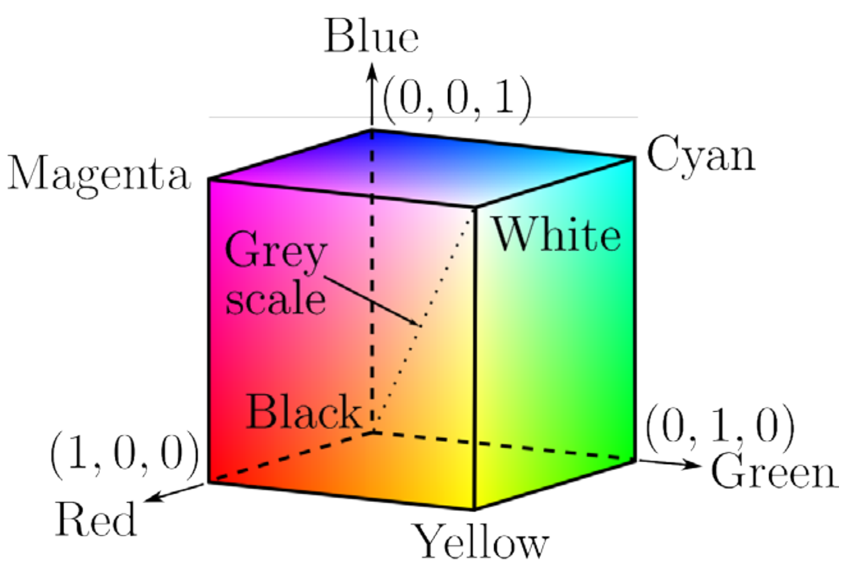
The RGB color space

- Each color is a triplet corresponding to R, G and B component.
- The color representation: (R,G,B)

e.g. Cyan (0,1,1)

- If R=G=B for all pixels in an image, it is a greyscale image.
- The RGB color space can be represented as a color cube.

Vertices are the primary (RGB) and secondary colours (CMY) plus black and white.



The RGB color space

RGB Color Model

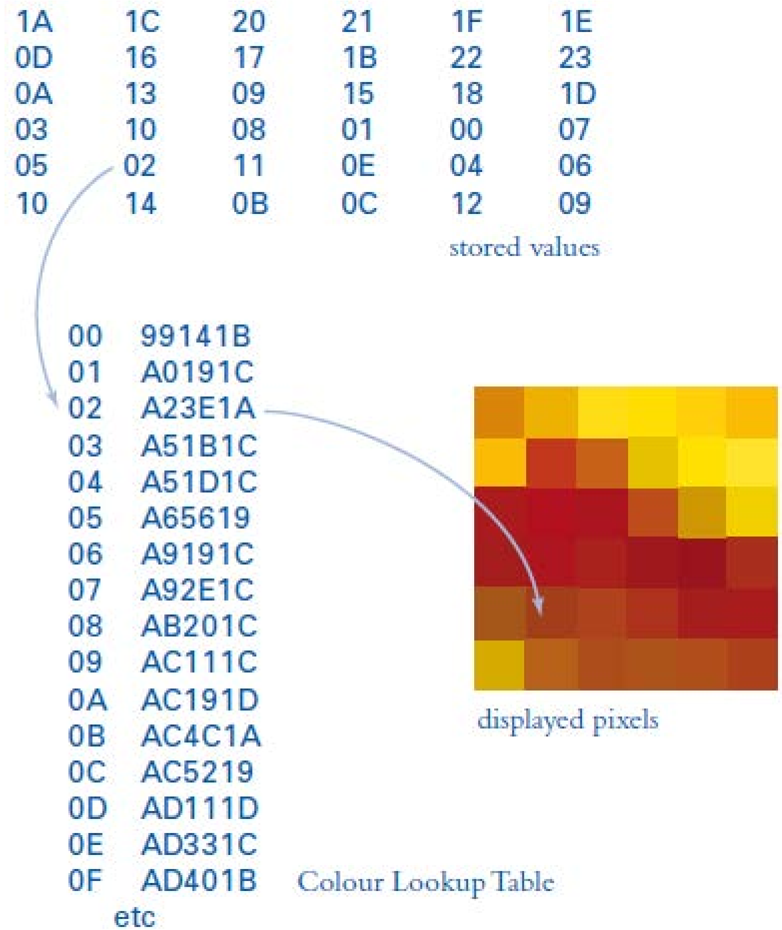
- The number of bits used to store a color value – **the color depth** – determines how many different colors can be represented.
- The use of lower color depths leads to posterization and loss of image detail, but reduces file size.

Detail of photograph in 24-bit (left) and 8-bit (right) colour

A photograph in 24, 8 (top), 4 and 1 (bottom) bit colour

RGB Color Model

- In indexed color, instead of storing a 24-bit color value for each pixel, we use an 8-bit value which serves as an index into a color table.
- The color table contains the palette of colors used in the image.



Using a colour table

Indexed color in RGB model

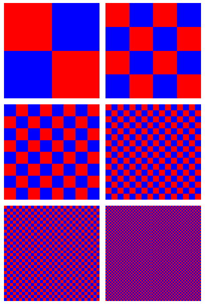
- In indexed color, instead of storing a 24-bit color value for each pixel, we use an 8-bit value which serves as an index into a color table. The color table contains the palette of colors used in the image.

Images and their palettes

Indexed color in RGB model

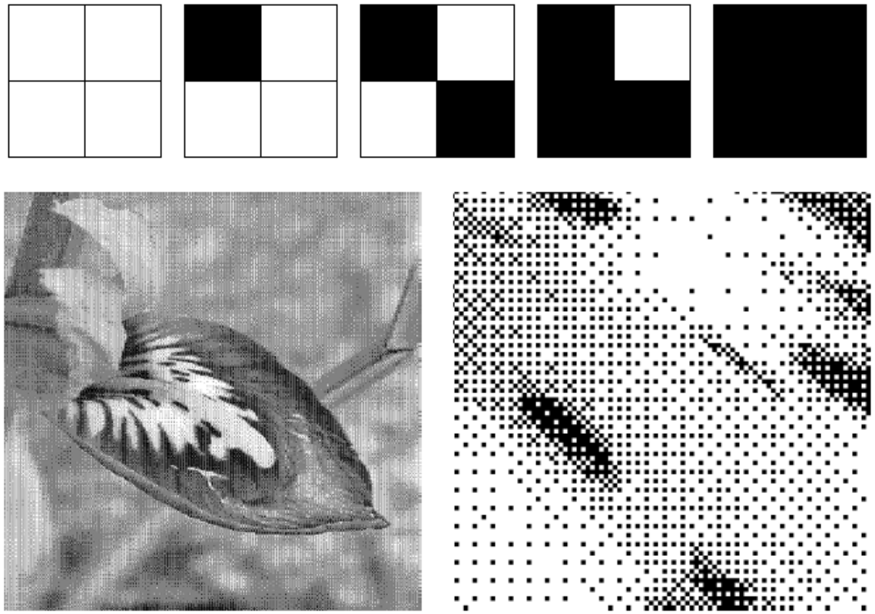
Two approaches of displaying image with a reduced palette

- Replace the colour value of each individual pixel with the color from the palette which is closest to it.
  - Can cause posterization
- Colour dithering
  - Areas of a single colour are replaced by a pattern of dots of several different colours, with the intention that optical mixing in the eye will produce the effect of a colour which is not really present.



RGB Color Model

- Some colors from the original image may be missing from the palette.
- Dithering can be used to reduce the resulting posterization.



Dithering in black and white

RGB Color Model

- Some colors from the original image may be missing from the palette. Dithering can be used to reduce the resulting posterization.

Dithering and posterization

Grayscale

- A grayscale image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.
- Grayscale images are distinct from one-bit black-and-white images, which are images with only the two colors, black, and white. Grayscale images have many shades of gray in between.

Grayscale - Representation

- Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from a full color image.

- The intensity of a pixel is expressed within a given range between a minimum and a maximum, inclusive. This range is represented in an abstract way as a range from 0 (total absence, black) and 1 (total presence, white), with any fractional values in between.

Grayscale - Representation

- Another convention is to employ percentages, so the scale is then from *0% to 100%*. This is used for a more intuitive approach, but if only integer values are used, the range encompasses a total of only *101* intensities, which are insufficient to represent a broad gradient of grays.

- In computing, although the grayscale can be computed through rational numbers, image pixels are stored in binary form. Today grayscale images intended for visual display are commonly stored with *8 bits per sampled pixel,* which allows 256 different intensities to be recorded, typically on a non-linear scale. The precision provided by this format is barely sufficient to avoid visible banding artifacts, but very convenient for programming due to a single pixel occupies a single byte.

Grayscale - Representation

- Technical uses often require more levels, to make full use of the sensor accuracy (typically *10 or 12 bits* per sample) and to guard against round-off errors in computations. Sixteen bits per sample (65,536 levels) is a convenient choice for such uses, as computers manage 16-bit words efficiently.

- The TIFF and the PNG (among other) image file formats supports 16-bit grayscale natively, although browsers and many imaging programs tend to ignore the low order 8 bits of each pixel.

- No matter what pixel depth is used, the binary representations assume that 0 is black and the maximum value (255 at 8 bit, 65,535 at 16 bit, etc.) is white, if not otherwise noted.

Grayscale - Conversion

- Conversion of a color image to grayscale is not unique.
- Different weighting of the color channels effectively represent the effect of shooting black-and-white film with different-colored photographic filters on the cameras.
- A common strategy is **to match the luminance of the grayscale image to the luminance of the color image**.
- To convert color image to a grayscale representation of its luminance
  - Step 1: Obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding.
  - Step 2:

$$30\%*R+59\%*G+11\%*B$$

Note: these weights depend on the exact choice of the RGB primaries, but are typical.

Or

$$(11*R + 16*G + 5*B) /32$$

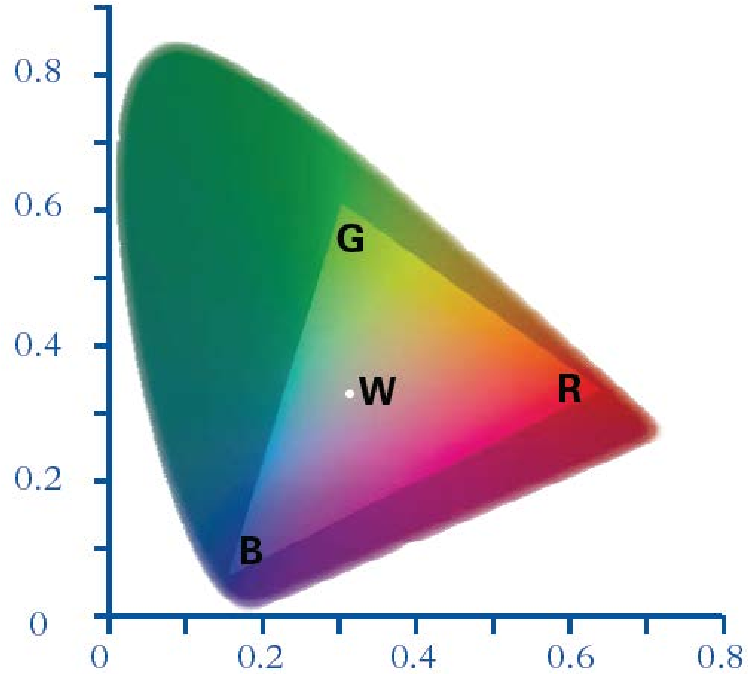
Note: The formula is also popular since it can be efficiently implemented using only integer operations.

Questions

1. Is it true that any colour can be produced by mixing red, green and blue light in variable proportions?
2. What are the colours correspond to the eight corners of the RGB colour space cube?
3. Explain the process of how the image of a non-luminous object is formed in the human eyes and what factors will influence the image formation process?

Solutions

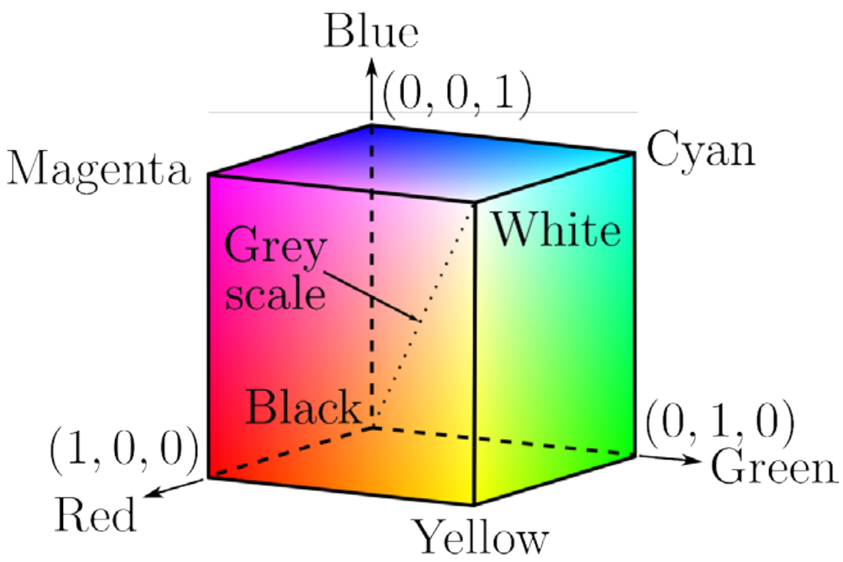
1. No, not all the colors can be represented by mixing Red, Green and Blue. There are colours outside the RGB gamut.



The RGB colour gamut

Solutions

2. Red, green and blue, where the cube intersects the R, G and B axes, respectively. Cyan, magenta and yellow at the corners opposite their respective complements. Black at the origin and white at the remaining corner.



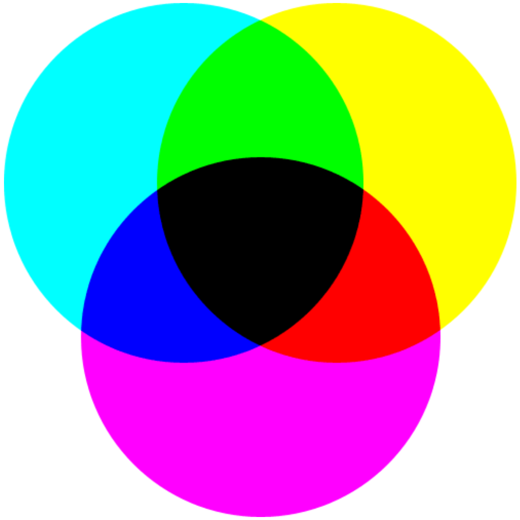
Solutions

3. The process:
   1. Light from a light source with SPD $\mathrm{E}(\lambda)$impinges on a surface of a object. SPD describes the power carried by each wavelength.
   2. The surface reflects different amounts of light at different wavelengths which is described by the surface spectral reflectance $\mathrm{S}(\lambda)$
   3. The reflected light goes into the human eyes and is filtered by the eyes' cone function $\boldsymbol{q}(\lambda) = [q_R(\lambda), q_G(\lambda), q_B(\lambda)]$.
   - The influence factors are: $\mathrm{E}(\lambda)$, $\mathrm{S}(\lambda)$, $\boldsymbol{q}(\lambda)$.

CMYK Color Model

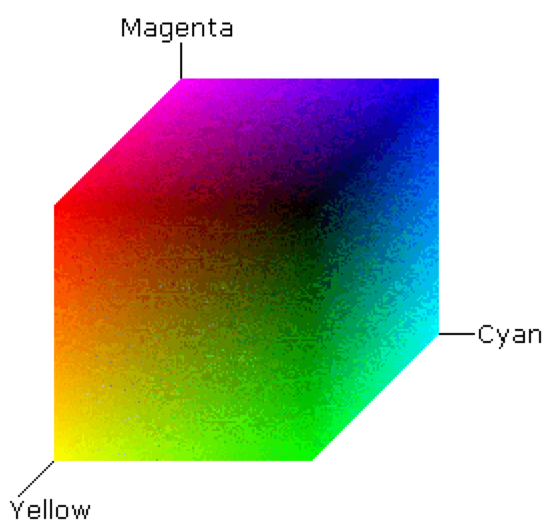
- A **subtractive** color model, used in color printing.
- Cyan, Magenta and Yellow are the subtractive primaries.
- CMYK refers to the four inks used in some color printing:

  cyan, magenta, yellow, and key (black).



CMYK Color Model

- The "K" in CMYK stands for *key* since in four-color printing cyan, magenta, and yellow printing plates are carefully keyed or aligned with the black key plate.
- The CMYK color space is related to the RGB color space by being the inverse of it.
  - White(0,0,0) and Black (1,1,1) and the primary axes of the coordinate system are cyan, yellow, and magenta.



The CMYK color space

CMYK Color Model

Since in this color space:

C =G+B=W-R

M=R+B=W-G

Y =R+G=W-B

+ means additive mixing of light

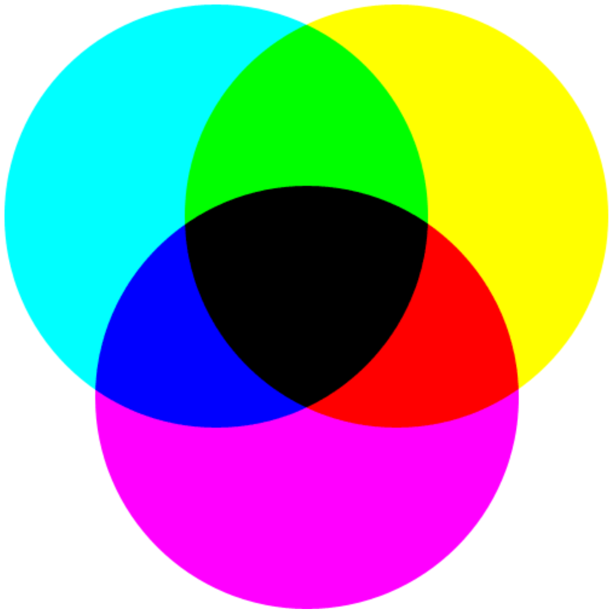
- subtraction of light

The complementary colors are:

C <-> R

M <-> G

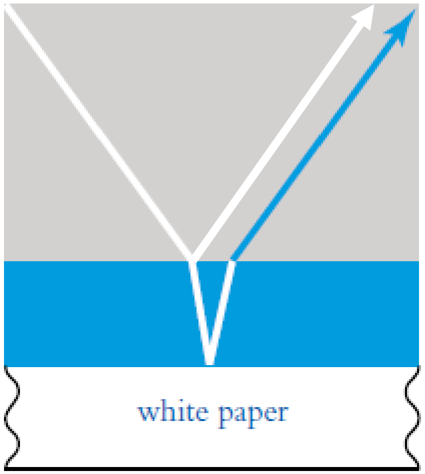
Y <-> B



CMYK Color Model

Two situations when light shine out on a colored surface:

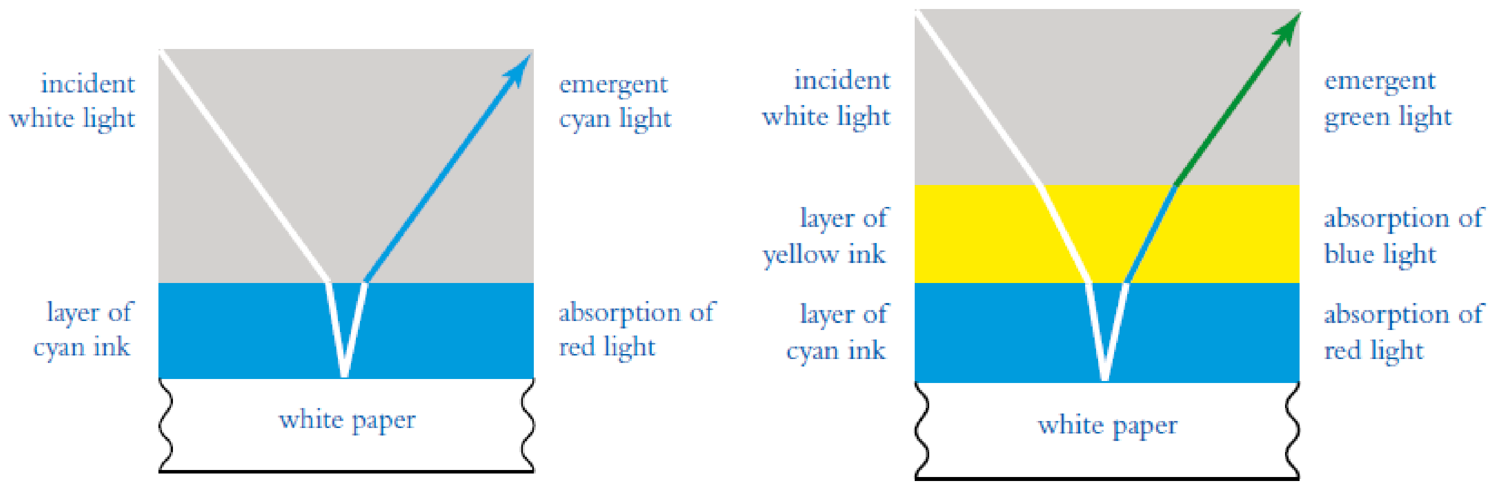
1. The light that is reflected from a colored surface is not changed in color.
2. The light that penetrates through a colored surface will be reflected or scattered back from beneath it. During the light's journey through the particles of dye, ink or paint, the pigments absorb light at some frequencies. The light that emerges thus appears to be colored.



CMYK Color Model

- Thin layers of ink absorb some components of the incident light, so overlaying ink, as in printing processes, mixes colors subtractively.
- Cyan ink absorbs red from incident white light but reflects blue and green which becomes Cyan.

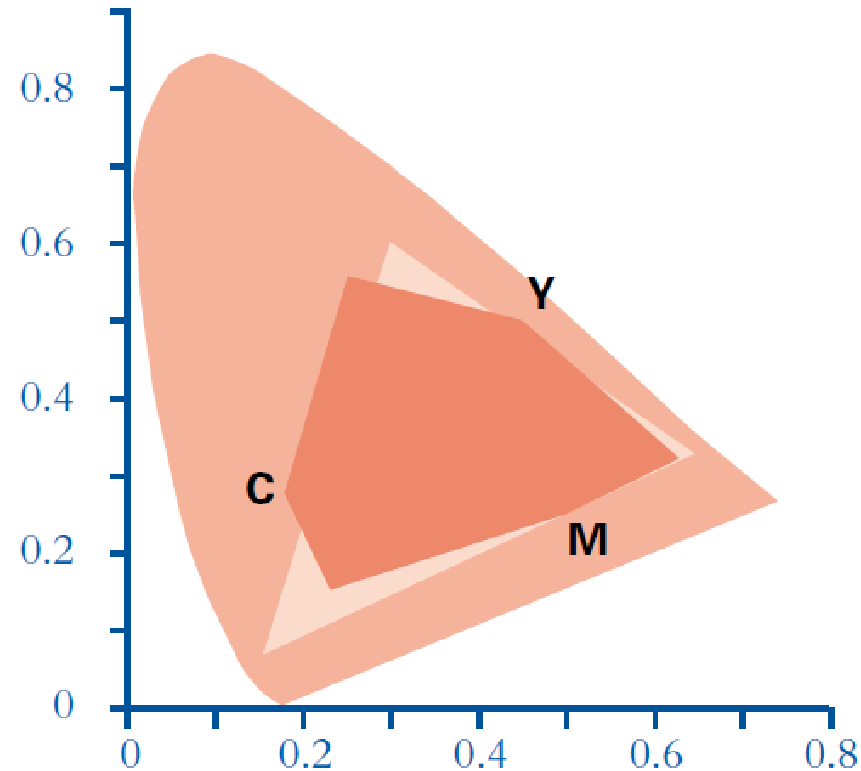
Cyan + Yellow → Green   Magenta + Cyan → Blue   Yellow + Magenta → Red



Coloured inks

CMYK Color Model

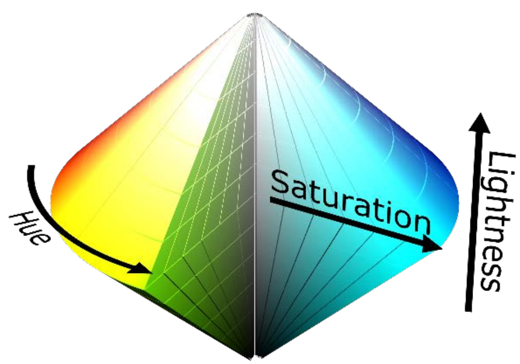
- The CMYK color gamut, corresponding to easily printable colors, is smaller than the RGB gamut, but some CMYK colors lie outside the RGB gamut.

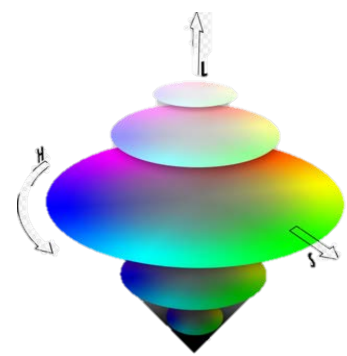


The CMYK gamut

HSL Color Space

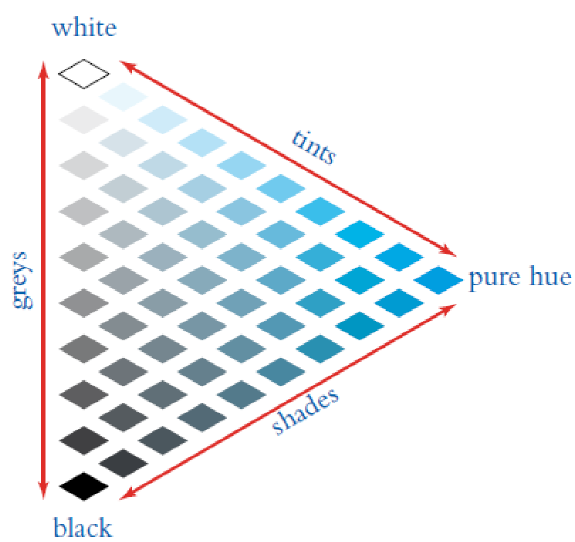
- A color can be identified by its hue, saturation and lightness (H,S and L).
- Hue is the particular wavelength at which the energy of the light is concentrated. (Hue is the pure color of light).
- Saturation describes the purity of the hue.
- Lightness describes how much light is put into the Hue.



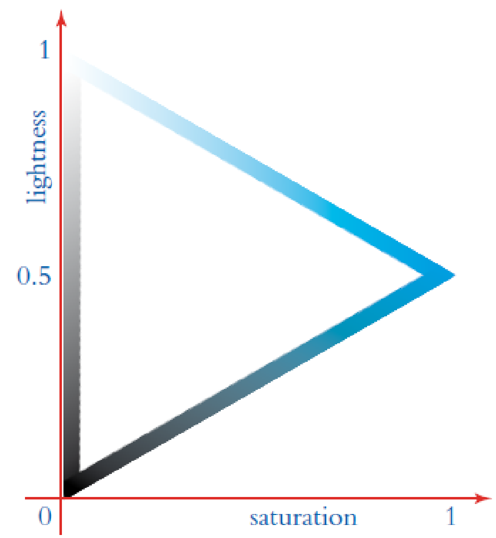


HSL Color Space

- Tones of a single hue can be arranged two-dimensionally, with lightness increasing upwards, and saturation increasing from left to right.



Tints, shades and tones



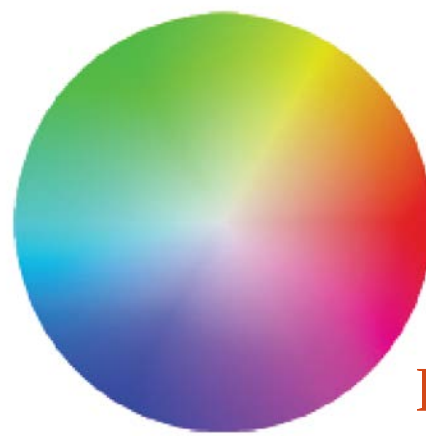
Saturation and lightness

HSL Color Space

- Hues can be arranged around the rim of a color wheel, with complementary colors opposite each other. A hue's value is the angle between its position on the wheel and the position of red.
- Saturation can be added to the model by filling in the circle, putting 50% grey at the centre of the circle, and then showing a gradation of tints from the saturated hues on the perimeter to the neutral grey at the centre

Hue only

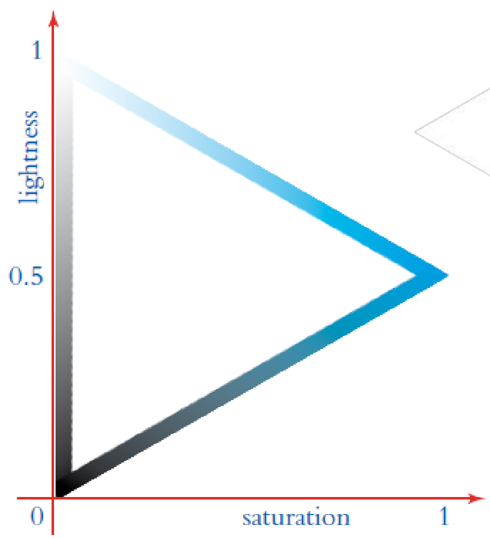
A colour wheel



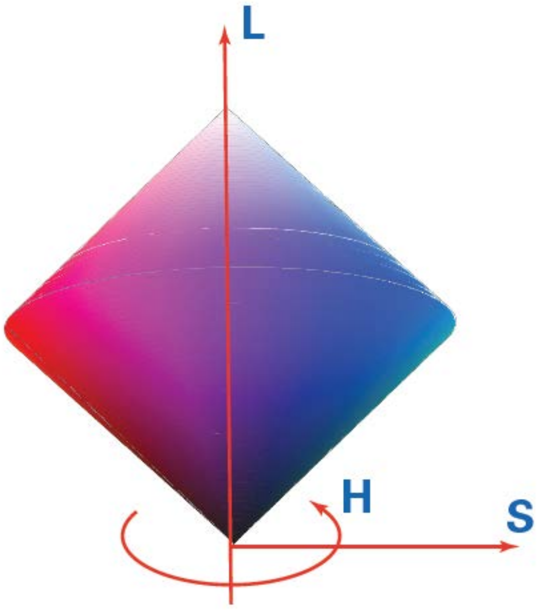
Hue+Saturation

Filling the colour wheel

HSL Color Space



Hue + Saturation



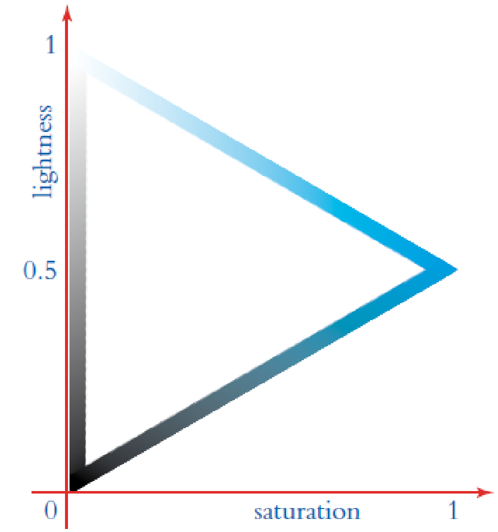
The HSL colour solid

Lightness + Saturation

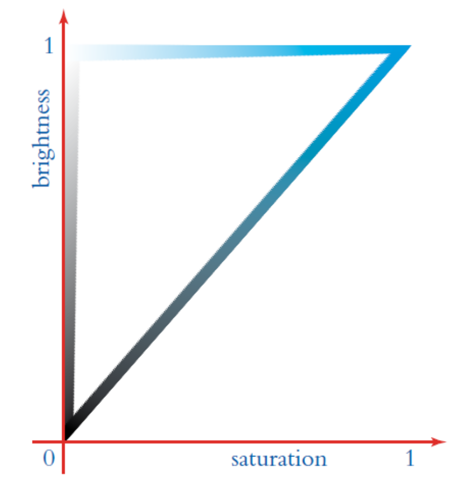
Hue + Saturation + Lightness

HSL Color Space

- Hue, saturation and lightness can be combined into a three-dimensional double cone. Any components color can be specified by its H, S and L.
- HSB is a variant of HSL, where the tones are arranged differently.
  - HSB is equivalent to HSV (brightness is called value sometimes)
  - Different from HSL assigning a lightness of 0.5 to pure hues, HSB gives pure hues a brightness value of 1.
  - Its shape is an inverted cone.



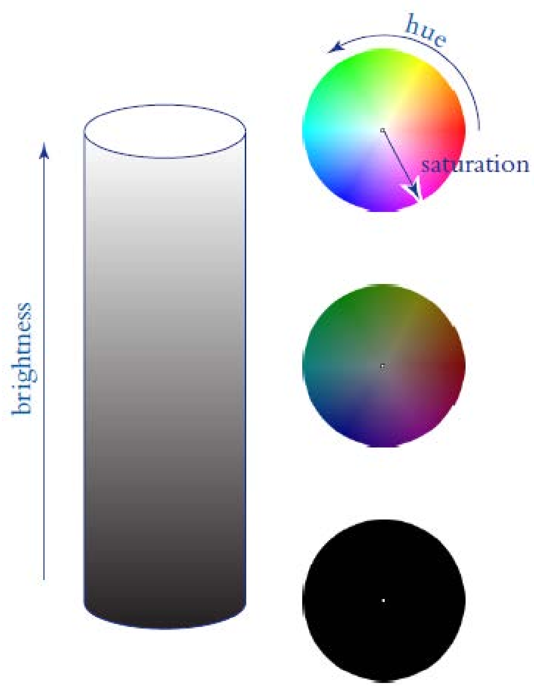
Saturation and lightness



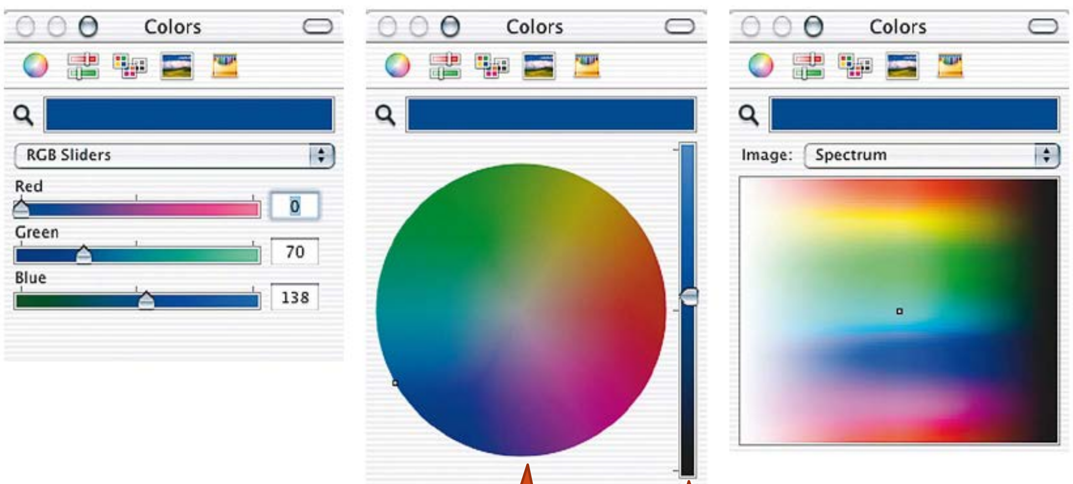
Saturation and brightness

HSL Color Space

- Both HSL and HSB are normally distorted into a cylindrical shape, so that they can be presented as color pickers.



The HSB cylinder



Colour pickers

Corresponding slice through the cylinder

Brightness

Models – Device Independence

- Different monitors provide different red, green and blue.
- Most color models suffers from not being *perceptually uniform*.
- Perceptually uniform:
  - The same change in one of the RGB values produce the same change in appearance, no matter what the original value is.
  - Difference between two colors (as perceived by the human eye) is proportional to the Euclidian distance within the given color space.
- CIE L*a*b* and L*u*v* color spaces are perceptually uniform and serve as device-independent reference models.

- Far away from mature!!!

Channels and Color Corrections

- The R, G and B components of each pixel can be stored as separate values.
- The three arrays of values can be treated as grayscale images, called channels.

An RGB colour image and its red, green and blue channels

The lightness represents the intensity of the corresponding color on a channel.

Red cherry appears white in Red channel and appears black in Green and Blue channels.

Channels and Color Corrections

- Each channel can be manipulated separately.

  In particular, levels and curves can be used to alter the brightness and contrast of each channel independently.

Correcting and over-correcting a colour cast

Channels and Color Corrections

Hue and saturation adjustments

- Adjustments to hue and saturation alters the colors of the image.

Channels and Color Corrections

- The color balance, hue and saturation and color replacement adjustments change the color of the image as a whole.
- Figure shows a repainted house door using the color replacement setting.

Colour replacement

Further Exploration

"Further Exploration" directory on http://www.cs.sfu.ca/mmbook/ include:

– More details on gamma correction

– The full specification of the new sRGB standard color space for WWW applications.

– A link to an excellent review of color transforms.

– A Matlab script to exercise (and expand upon) the color transform functions that are part of the Image Toolbox in Matlab: the standard 'Lena' image is transformed to YIQ and to YCbCr.

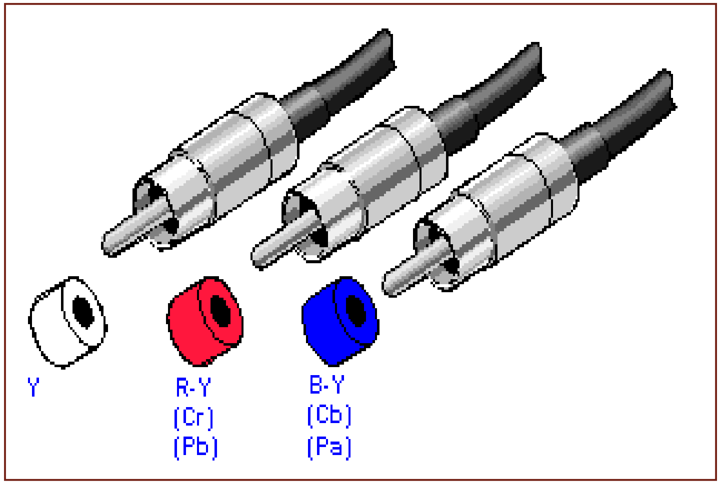
– The new MPEG standard, MPEG-7, includes six color spaces.

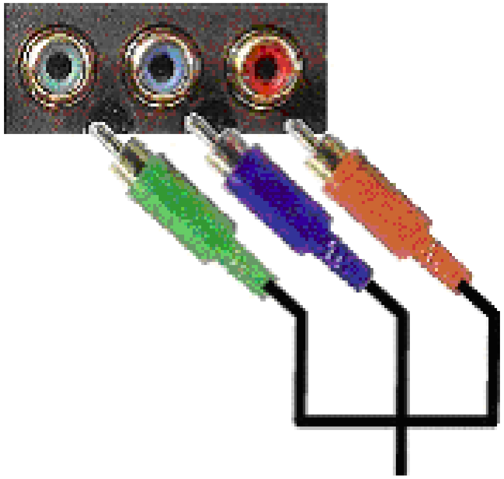
Fundamental Concepts in Video

- Types of Video Signals
- Analog Video
- Digital Video
- Video Processing Techniques
- Further Exploration

Component video — 3 signals

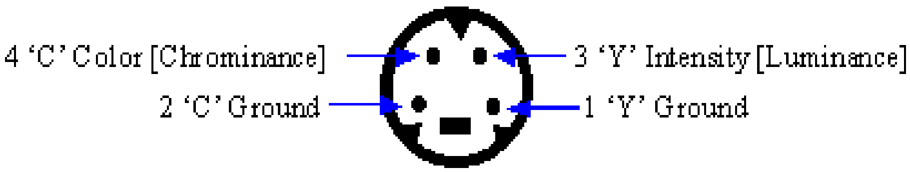
- Higher-end video systems make use of three separate video signals for the red, green, and blue image planes.
- Most computer systems use Component Video, with separate signals for R, G, and B signals.
- For any color separation scheme, Component Video gives the best color reproduction since there is no "crosstalk" between the three channels, unlike S-Video or Composite Video.
- Component video requires more bandwidth and good synchronization of the three components.





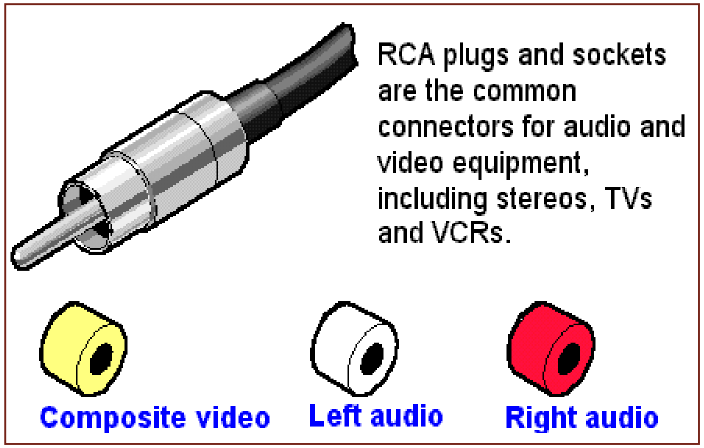
S-Video — 2 Signals

- Also known as Separated Video, or Super-video
- S-Video uses two wires, one for luminance and another for a composite chrominance signal.
  - The grayscale information is most crucial for visual perception.
  - Humans are able to differentiate spatial resolution in grayscale images much better than for the color part of color images.
  - Less crosstalk between the color information and the grayscale information
- S-video provides a sharper image than composite video, but is not as good as component video.



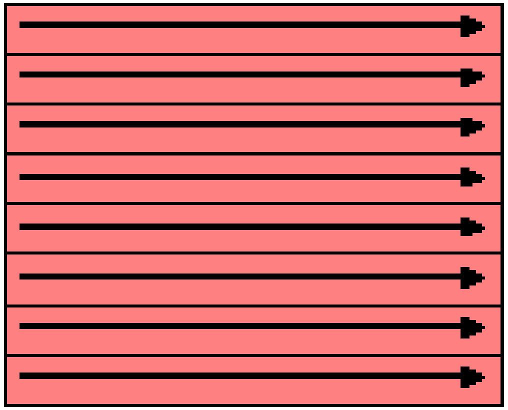
Composite Video — 1 Signal

- Color (“chrominance”) and intensity (“luminance”) signals are mixed into ***a single carrier*** wave.
- The chrominance and luminance components can be separated at the receiver end and then the color components can be further recovered.
- When connecting to TVs or VCRs
  - Composite Video uses only one wire.
  - Video color signals are mixed, not sent separately.
  - The audio and *sync* signals are additions to this one signal.
- Since color and intensity are wrapped into the same signal, some interference between the luminance and chrominance signals is inevitable.



4.2 Analog Video

- An analog signal *f(t)* samples a time-varying image.
- Two type of scanning
  - *Progressive* scanning
    - traces through a complete picture (a frame) row-wise for each time interval.
    - A HD computer monitor typically uses a time interval of 1/72 second.
  - Interlaced scanning
    - is used in TV and in some monitors and multimedia standards.
    - reduces perceived flicker since it was difficult to transmit the amount of information in a full frame quickly.



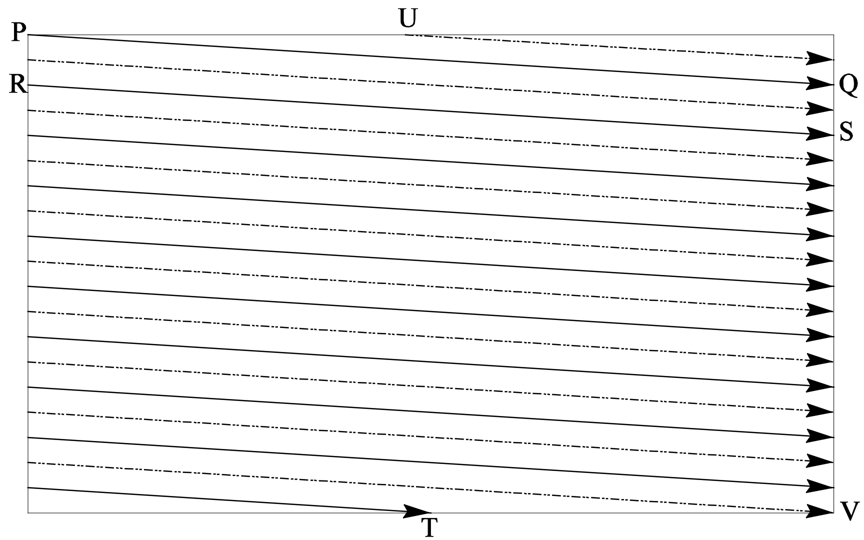
Progressive Scan (Non-interlaced)

Interlaced scanning

- The odd-numbered lines are traced first, and then the even-numbered lines are traced.
- This results in "odd" and "even" fields — two fields make up one frame.
- In fact, the odd lines (starting from 1) end up at the middle of a line at the end of the odd field, and the even scan starts at a half-way point.

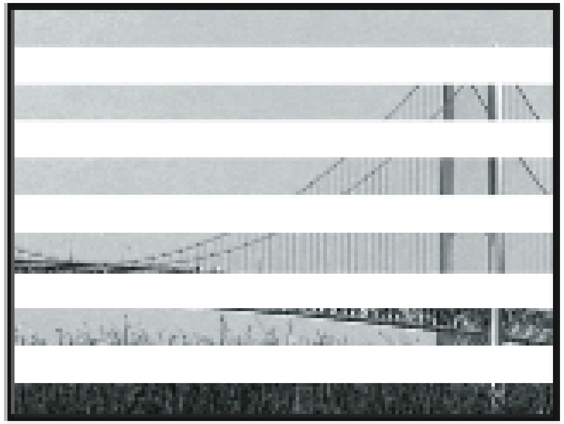
Interlaced

Interlaced scanning

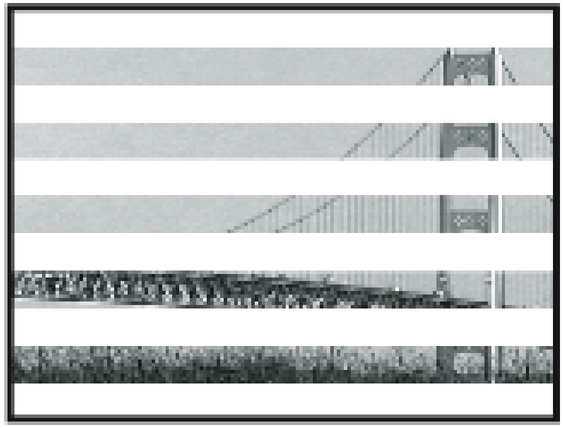


- The solid (odd) lines are traced, P to Q, then R to S,…,ending at T. The even field starts at U and ends at V.
- Horizontal retrace-the jump from Q to R, etc.
- Vertical retrace-the jump from T to U or V to P, etc.

Interlaced scanning



Field 1 is sampled first and contains only the odd lines.



Field 2 sampled 1/60th of a second later contains the even-lines.

A complete frame consists of field 1 and field 2.

Interlaced scanning

- The odd and even lines are displaced in time from each other
  - Generally not noticeable except when very fast action is taking place on screen, when blurring may occur.

- In the video in Fig. 4.2, the moving helicopter is blurred more than the still background.

Interlaced scanning

(a) The video frame

(b) Field 1

(c) Field 2

(d) Different of Fields

- De-interlacing
    - Is used to change the frame rate, resize, or produce still images from an interlaced source video.
    - The simplest de-interlacing method consists of discarding one field and duplicating the scan lines of the other field. The information in one field is lost completely using this simple technique.
- Define the beginning of a new video line
    - Voltage is one dimensional signal that varies with time.
    - Analog video use a small voltage offset from zero to indicate "black", and another value such as zero to indicate the start of a line. Namely, we could use a "blacker-than-black" zero signal to indicate the beginning of a line.

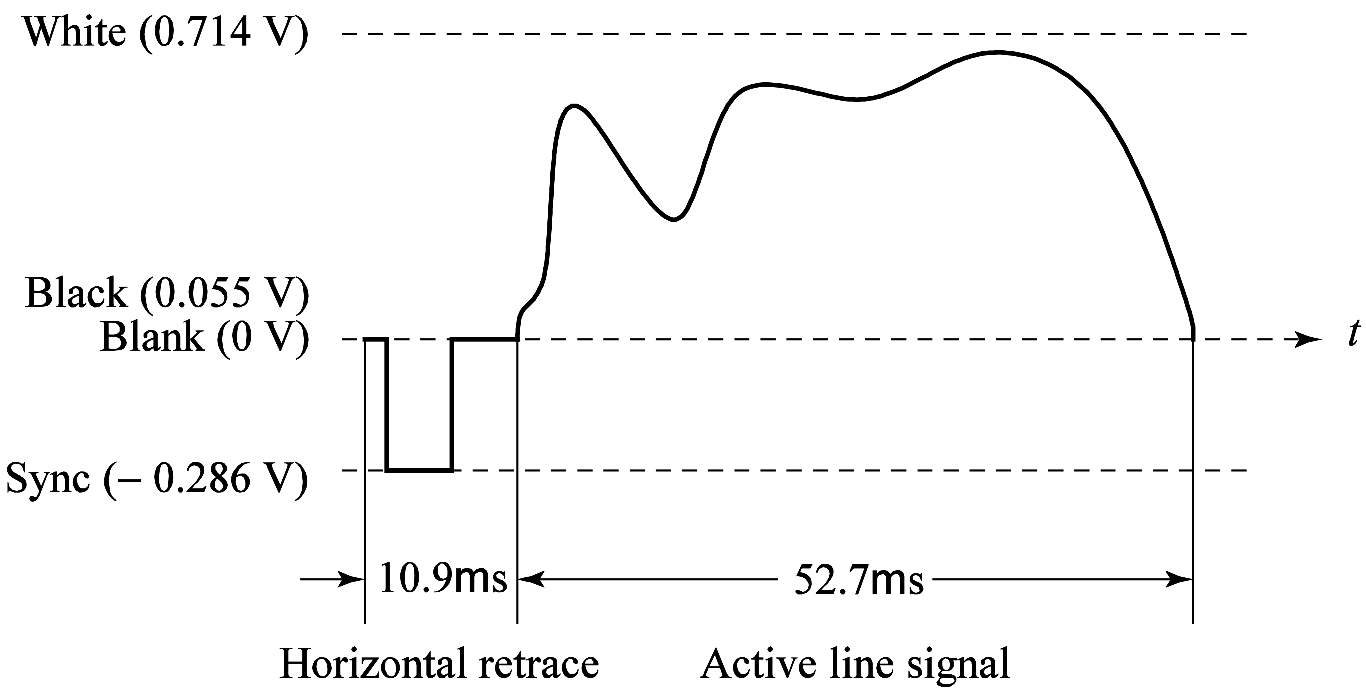


Figure. Electronic signal for one NTSC scan line.

4.3 Digital Video

- Some advantages:
  - Storing video on digital devices or in memory, ready to be processed (noise removal, cut and paste, etc.), and integrated to various multimedia applications.
  - Direct access, which makes nonlinear video editing simple.
  - Repeated recording does not degrade image quality.
  - Ease of encryption and better tolerance to channel noise.

Digital Video connectors

- Digital Video connectors are used to deliver the highest quality video signal. The technology uses TMDS (Transition Minimized Differential Signaling) to transmit large amounts of digital data from the source to the display, resulting in a high-quality image.
- DVI (Digital Visual Interface) was developed by the industry body DDWG (the Data Display Working Group) to send digital information from a computer to a digital display, such as a flat-panel LCD monitor.
- HDMI took a step forward by integrating audio and video into a more compact interface.
- DisplayPort is an interface technology that is designed to connect high-graphics capable PCs and displays as well as home theater equipment and displays. DisplayPort is similar to HDMI in that the DisplayPort signal carries both digital audio and video.

DVI - Digital Visual Interface

- DVI is a video display interface that is used to connect a video source to a display device, such as a computer monitor. It was developed with the intention of creating an industry standard for the transfer of digital video content.
- This interface is designed to transmit uncompressed digital video and can be configured to support multiple modes such as DVI-A (analog only), DVI-D (digital only) or DVI-I (digital and analog). Featuring support for analog connections, the DVI specification is compatible with the VGA interface.
- Although DVI is predominantly associated with computers, it is sometimes used in other consumer electronics such as television sets and DVD players.

HDMI - High Definition Multimedia Interface

- HDMI stands for High Definition Multimedia Interface.
- This technology carries the same video information as DVI but adds the capacity for digital audio and control signals as well.
- Found on many home theater/consumer electronics devices, HDMI uses a 19-pin connector that is held in place by friction.

HDMI™
HIGH-DEFINITION MULTIMEDIA INTERFACE

DisplayPort

- DisplayPort is an interface technology that is designed to connect high-end graphics capable PCs and displays as well as home theater equipment and displays.
- Like HDMI and DVI, DisplayPort utilizes TMDS (Transition Minimized Differential Signaling) link technology to send high bandwidth video and audio signals.
- The 20-pin connector allows the contact point to send maximum data transfer rates of 8.64 Gbps plus 1 Mbps for its AUX channel which can carry additional data.

HDTV (High Definition TV)

- High Definition Television (HDTV) is video that has resolution substantially higher than that of traditional television systems (standard-definition TV, or SDTV, or SD).
- HDTV has one or two million pixels per frame, roughly five times that of SD.
- Early HDTV broadcasting used analog techniques, but today HDTV is digitally broadcast using video compression.

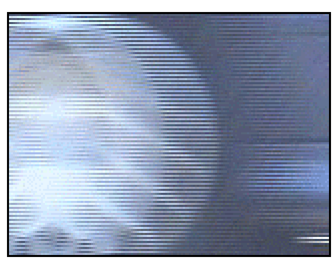
- The standard supports video scanning formats shown in Table below, where "I" mean interlaced scan and "P" means progressive (non-interlaced) scan.

- Table 1. Advanced Digital TV formats

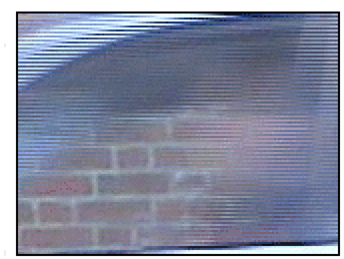
# of Active Pixels per line	# of Active Lines	Aspect Ratio	Picture Rate
1,920	1,080	16:9	60I 30P 24P
1,280	720	16:9	60P 30P 24P
704	480	16:9 & 4:3	60I 60P 30P 24P
640	480	4:3	60I 60P 30P 24P

- For video, *MPEG-2* is the compression standard.
- For audio, *AC-3* is the standard. It supports the so-called 5.1 channel Dolby surround sound, i.e., five surround channels plus a subwoofer channel.
- The salient difference between conventional TV and HDTV:
  - HDTV has a much wider aspect ratio of 16:9 instead of 4:3. (1/3 wider)
  - HDTV moves toward progressive (non-interlaced) scanning. The rationale is that interlacing introduces serrated edges to moving objects and flickers along horizontal edges.


progressive scan


interlace


progressive scan


interlace

- The FCC has planned to replace all analog broadcast services with digital TV broadcasting. The services provided include:
  - **SDTV (Standard Definition TV)**: the current NTSC TV or higher.
  - **EDTV (Enhanced Definition TV)**: 480 active lines or higher, i.e., the third and fourth rows in Table 1.
  - **HDTV (High Definition TV)**: 720 active lines or higher.
  - Popular choices:
  - 720p (720 lines, progressive, 30fps)
  - 1080I(1080 lines,interlaced,30fps or 60fps)

Data Rate and Video Storage Size

- Calculate the data rate in **bps** (bits per second) and storage requirement in **bytes** for a one-hour grayscale video with 800 X 600 frame size and 24 fps (frames per second) frame rate.

- Data rate = Resolution * Bits per pixel * Frame rate
- Storage size= Data rate * Time

4.4 Video Processing Techniques

- This section mainly discusses how animation can be produced with the help of computers.

- Other processing techniques, such as video compression and transmission, will be discussed in another course "Digital Image and Video Processing" in year 4.

Image Sequences

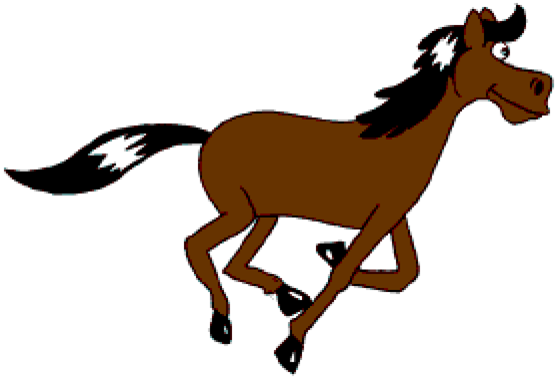
- Animation is the creation of moving pictures, one frame at a time.

- Traditional animators have developed many techniques, including cel animation, stop motion and claymation.

Traditional Animation

- Cel animation (classical animation or hand-drawn animation)
  - The oldest and historically the most popular form of animation.
  - Each frame is drawn by hand.
  - In contrast with the more commonly used computer animation nowadays.

  https://conceptartempire.com/cel-animation/



A horse animated from Eadweard Muybridge's 19th century photos. The animation consists of 8 drawings, which are "looped", i.e. repeated over and over. This example is also "shot on twos", i.e. shown at 12 frames per second. (24fps)

Traditional Animation

- Stop motion (stop action or frame-by-frame)
  - An animation technique to make a physically manipulated object appear to move on its own.
  - The object is moved in small increments between individually photographed frames, creating the illusion of movement when the series of frames is played as a continuous sequence.

A simple stop motion animation of a moving coin.

Traditional Animation

- Claymation (or clay-animation)
  - One of many forms of stop motion animation.
  - Each animated piece, either character or background, is "deformable" — made of a malleable substance, usually Plasticine clay.

Wallace and Gromit

A clay animation scene from “A Fan Ti”.

Image Sequences

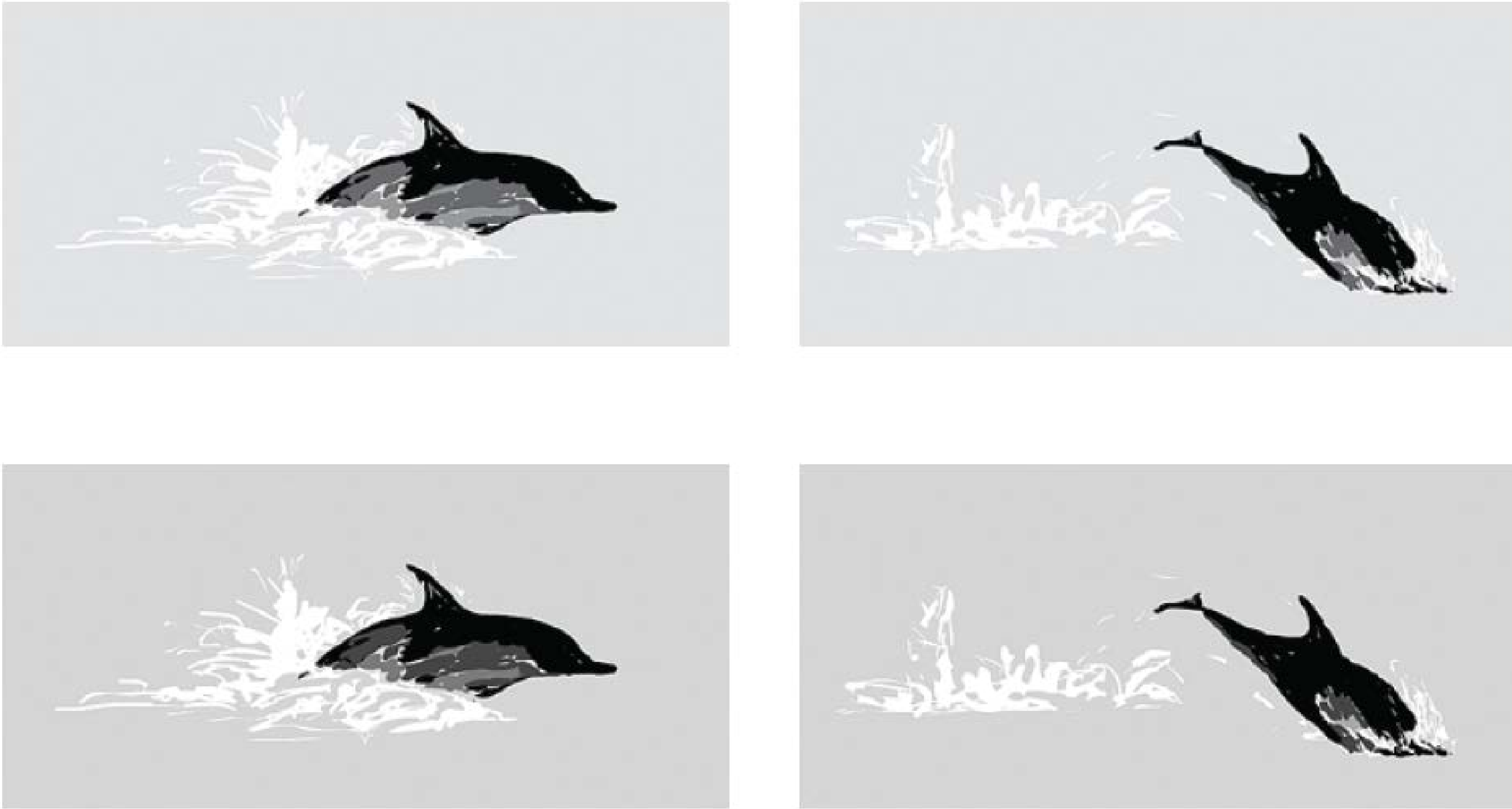
- Traditional animation can be captured one frame at a time using a camera connected to a computer, instead of being recorded on film.
- Animation can be created digitally.
- Individual frames can be created in a graphics program.
- Using layers to represent the contents of a frame can streamline the animation process.
- A sequence of images can be stored in consecutively numbered image files, which can be imported into video editing programs such as Flash.

Image Sequences

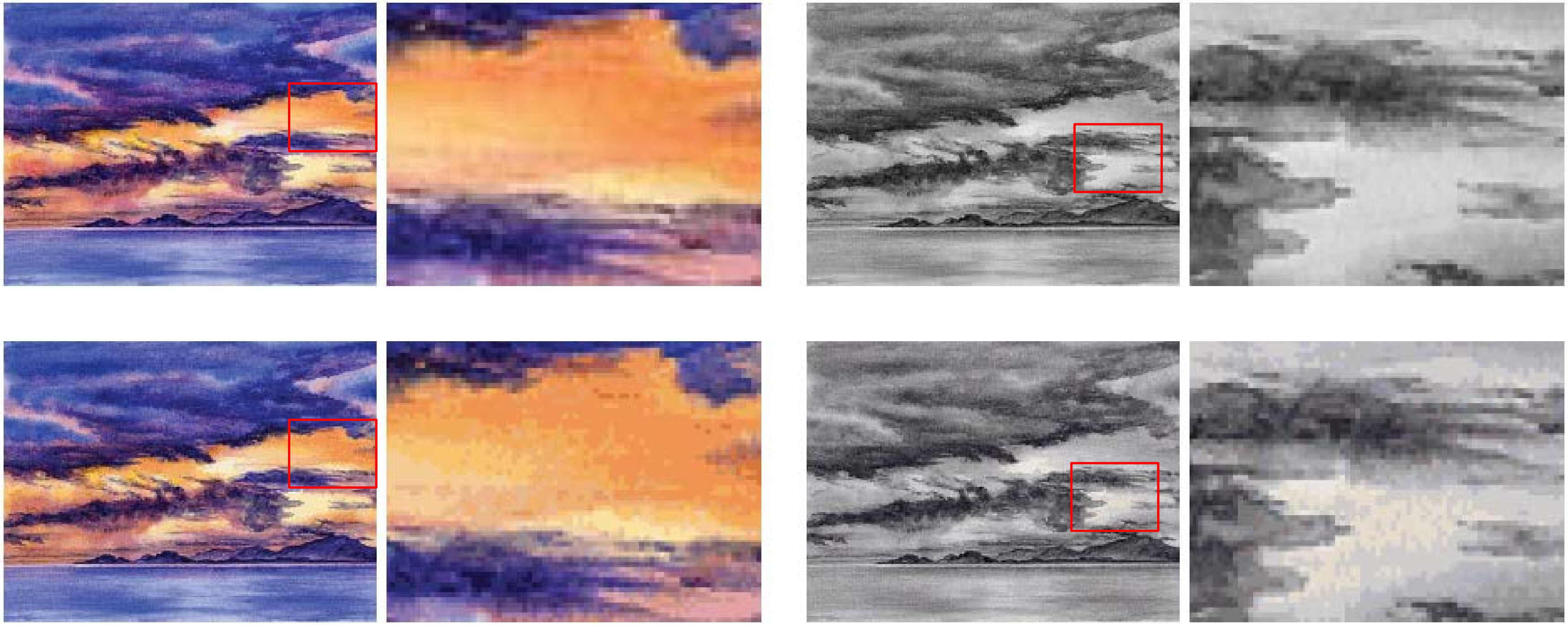
- An animation is a sequence of frames, each one a still image.
- Either bitmapped images or vector graphics can be used for the individual frames.
  - Vector graphics offer more possibilities for creating and manipulating frames using computer programs.
  - Bitmapped images are conceptually simpler, though, and correspond more closely to the traditional animations consisting of a sequence of photographed images on film.

Image Sequences

- An animated GIF contains multiple bitmapped images in a single file.
- The individual images can be displayed in sequence by Web browsers and other programs, without plug-in.
- GIFs are only suitable for short simple animations.
- GIFs use indexed color and lossless intra-frame compression, whose effectiveness depends on the nature of the images in the animation.
- GIFs cannot have a soundtrack or player controls.



Original (top) and animated GIF (bottom) frames, suitable material



Original (top) and animated GIF (bottom) frames, unsuitable material

Interpolation

- In traditional animation, chief animators draw key frames at important points; in-betweeners create intervening frames.
- Interpolation: the calculation of values lying between known points.
- Animation programs perform equivalent in-betweening by interpolating the values of properties such as position between key frames.
- Interpolation can be applied to layers in bitmapped images or to properties of vector objects.

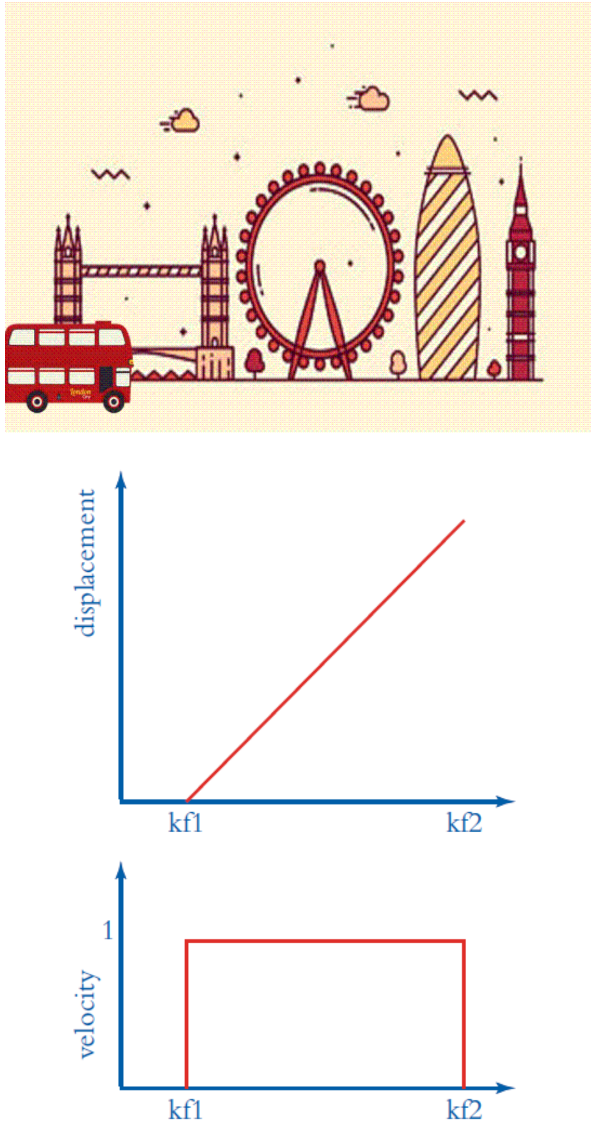
Interpolation

- Bitmaps do not contain identifiable objects
  - Use layers to isolate different elements of an animation if we wish to change them independently.

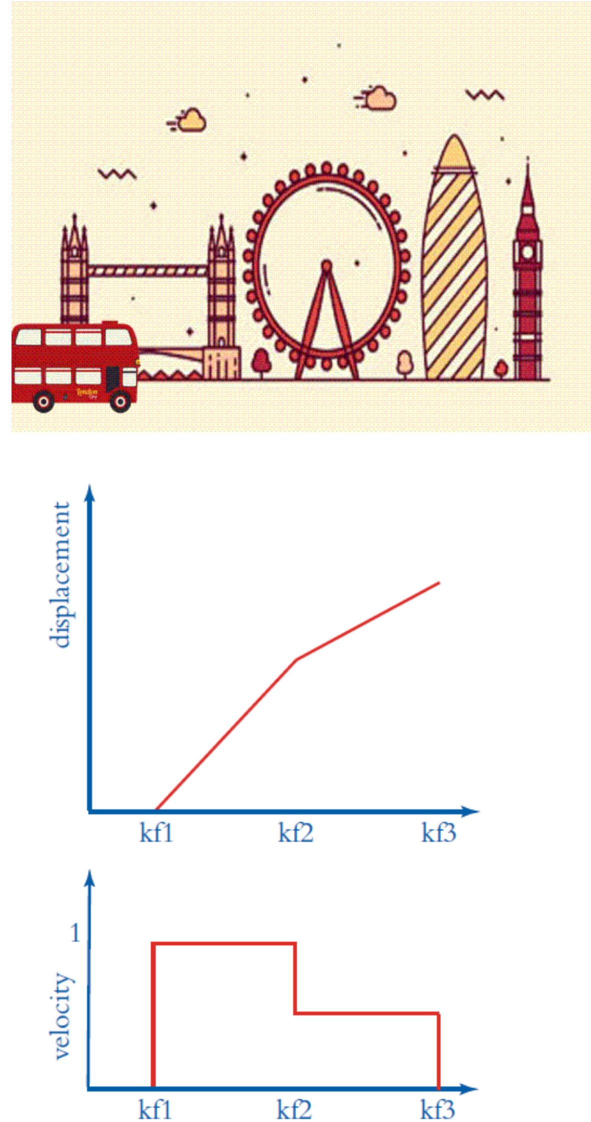
- Vector animations, we do have identifiable objects, and their properties are represented entirely numerically.
  - Makes interpolating the position, size, color and other properties of vector objects conceptually and practically easier.

Interpolation

- Walt Disney developed a mass production approach to animation.
- Disney's approach create animations relied on breaking down the production of a sequence of drawings into sub-tasks.
- If motion is interpolated linearly, movement begins and ends instantaneously. And there may be unnatural discontinuities between interpolated sequences.



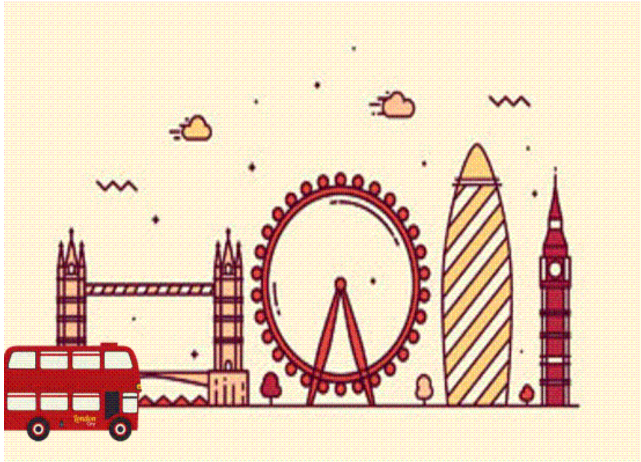
Linearly interpolated motion

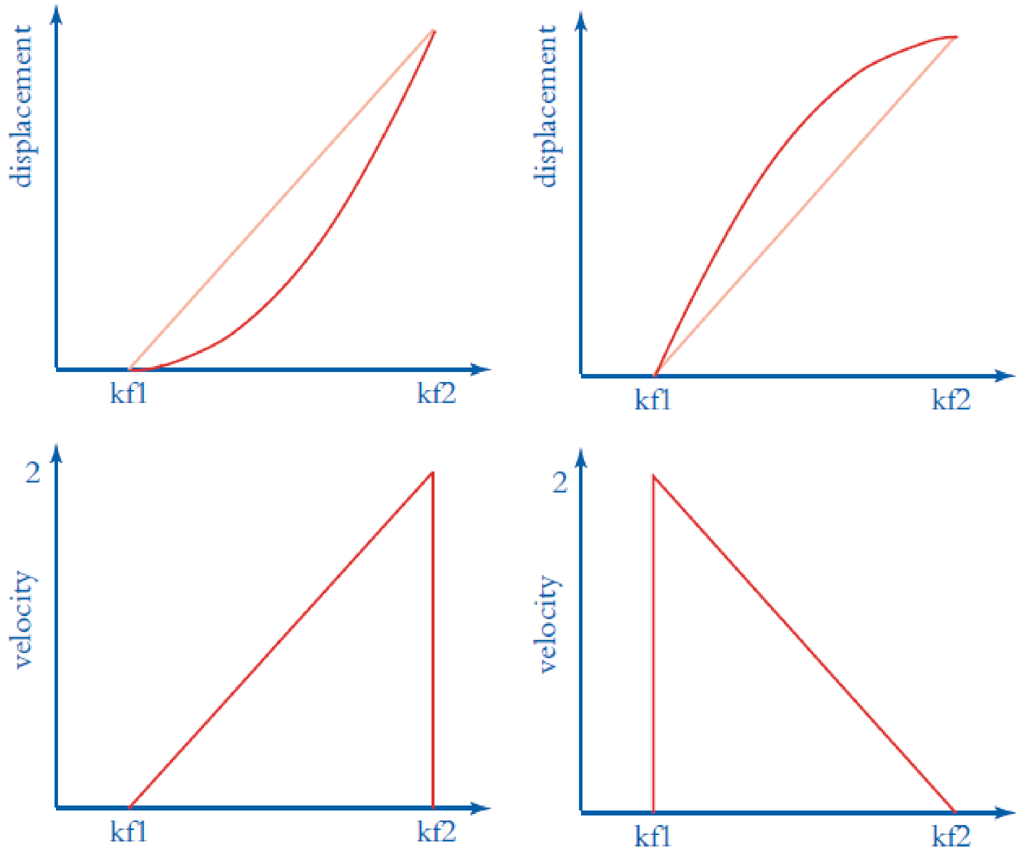


Abrupt change of velocity caused by linear interpolation

Interpolation

- Easing in and out can be used to cause the motion to increase or decrease gradually.

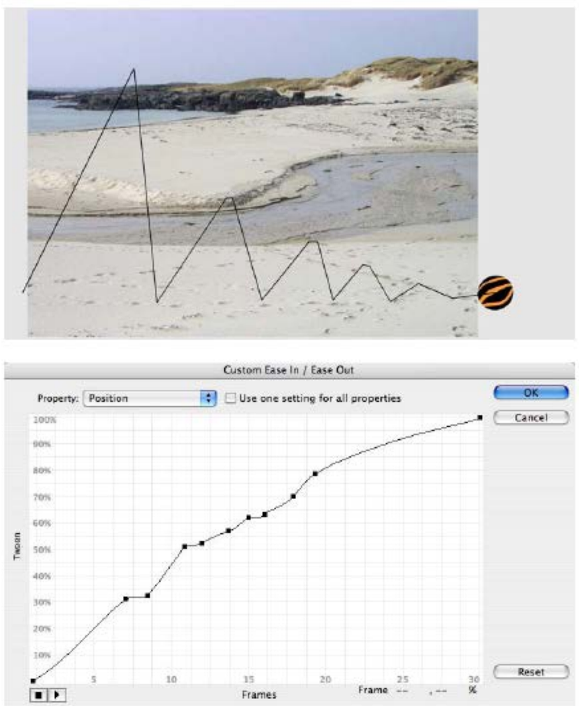




Quadratic easing in (left) and out (right)

Interpolation

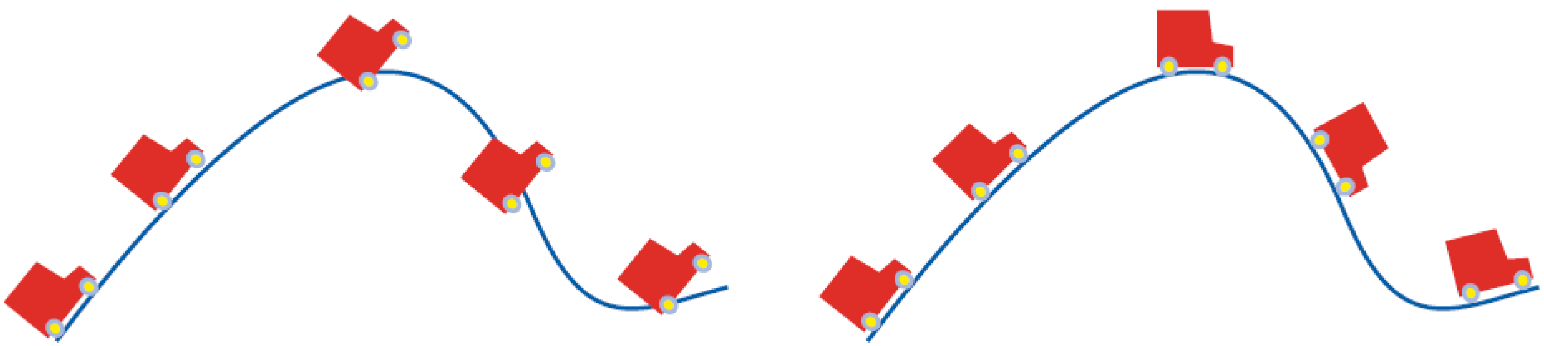
- Custom easing using Bézier curves is used to control the rate of change in arbitrarily complex ways.



Interpolating motion along a path with custom easing

Interpolation

- Objects or layers can be made to move along motion paths.
- When using motion paths, it is usually necessary to orient the moving object to the path to achieve a realistic effect.



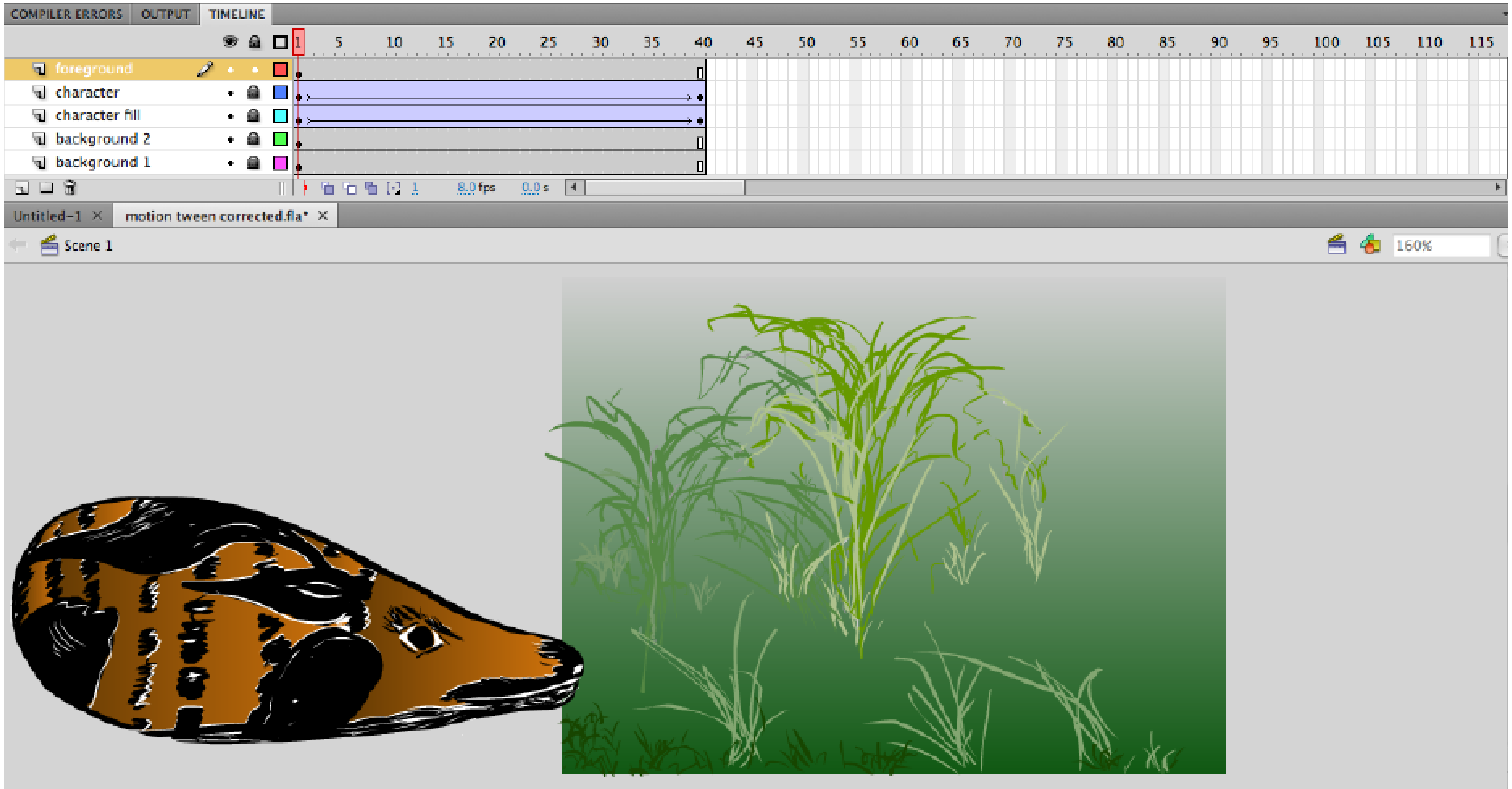
Fixed orientation (left) and orientation to the motion path (right)

Vector Animation

- Flash movies, also known as SWF files, a popular Web animation format. They are usually created in Flash, but SWFs may also be exported from other programs.

- An animation being created in Flash is organized using a timeline.

- The vector objects used in the animation are created on the stage, using conventional vector drawing tools and techniques.



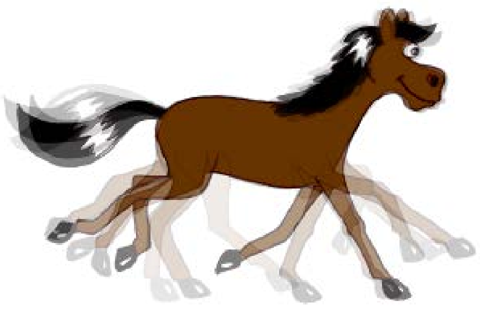
The timeline (top) and stage (below) in a simple Flash movie

Vector Animation

- Onion-skinning can be used to help align and change objects in consecutive frames.

- Key frames are drawn in their entirety on the stage.

- Ordinary frames have no content, they just hold the picture from the preceding key frame.



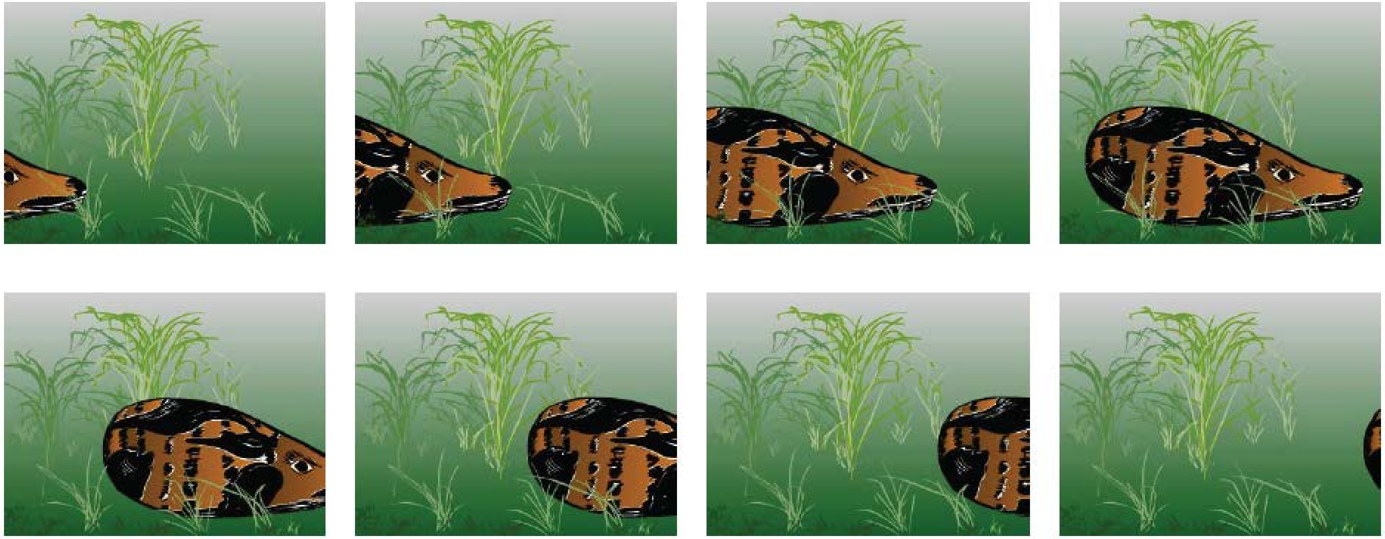
Vector Animation

- Graphical objects can be stored in a library as **symbols**.
- Instances of symbols can be created on the stage, allowing objects to be reused.
- Instances can be transformed independently and have different visual effects applied to them.

Instances of a symbol

Vector Animation

- Interpolation ("tweening") is applied to symbol instances.
- Easing can be applied to tweened motion.
- An object's size, orientation, opacity and color may also be interpolated.



Simple tweened motion of a symbol instance

Vector Animation

- Shape tweening ("morphing") is used to transform one shape into another.

4.5 Further Exploration

http://www.cs.sfu.ca/mmbook/ Further Exploration-Chapter5

- Tutorials on NTSC television
- The latest news on the digital TV front
- Introduction to HDTV
- Adobe Flash Software

Multimedia Compression

- Image Compression
  - JPEG
- Video Compression
  - Spatial Compression
  - Temporal compression
  - MPEG
- Entropy Coding

JPEG

- Image standard developed by the Joint Photographic Experts Group.
- It is a lossy compression - some information is permanently lost.
- JPEG compresses color images as three separate greyscale images.
- The core of JPEG is to work out what data to throw away.

JPEG

Original image
24-bit RGB bitmap image
with 73,242 pixels
219kB in size

Image Compression: JPEG

JPEG quality level

Q = 50 15kB

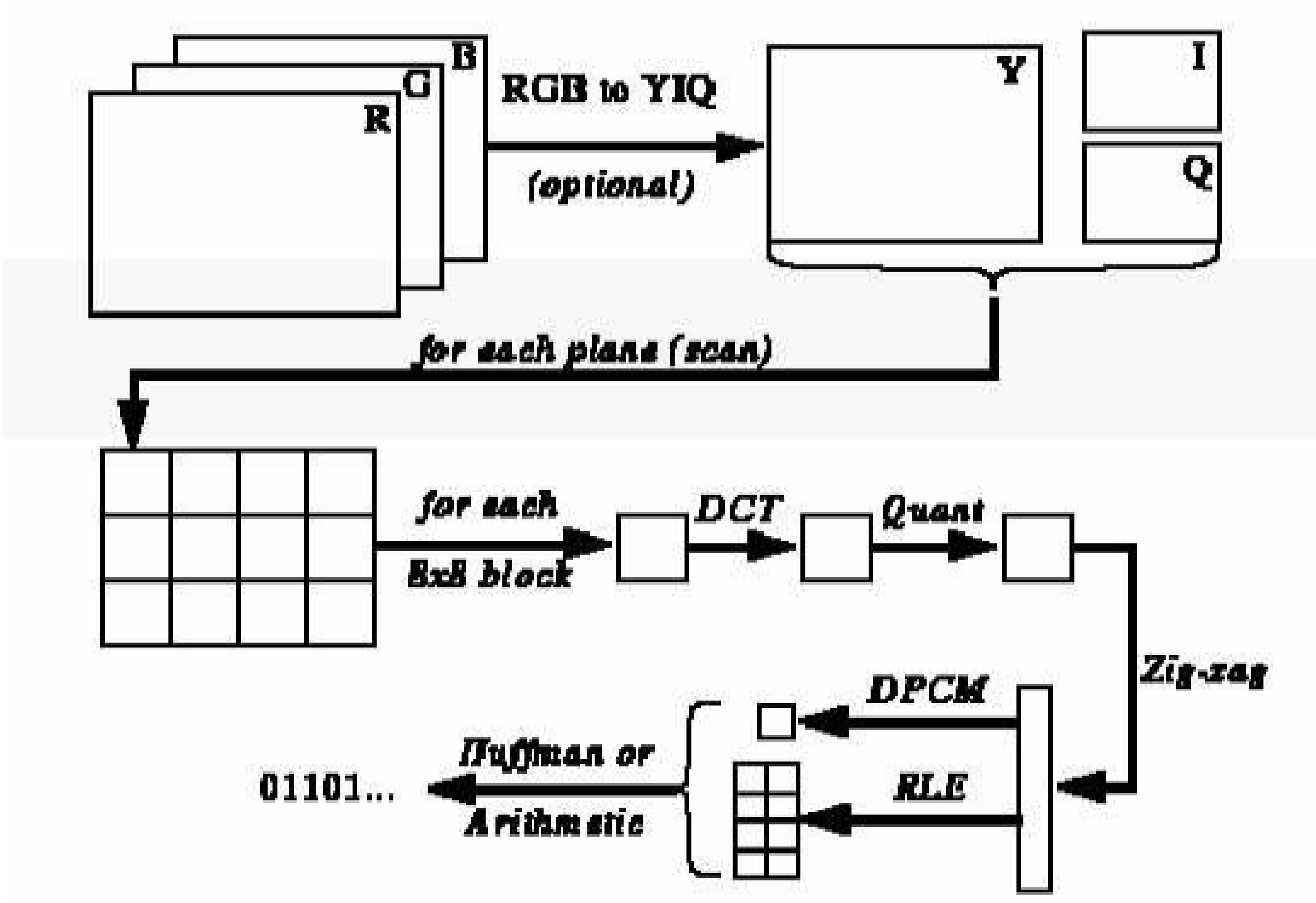
Q = 25 9kB

Q = 10 5kB

Q = 1 2kB

File size in bytes

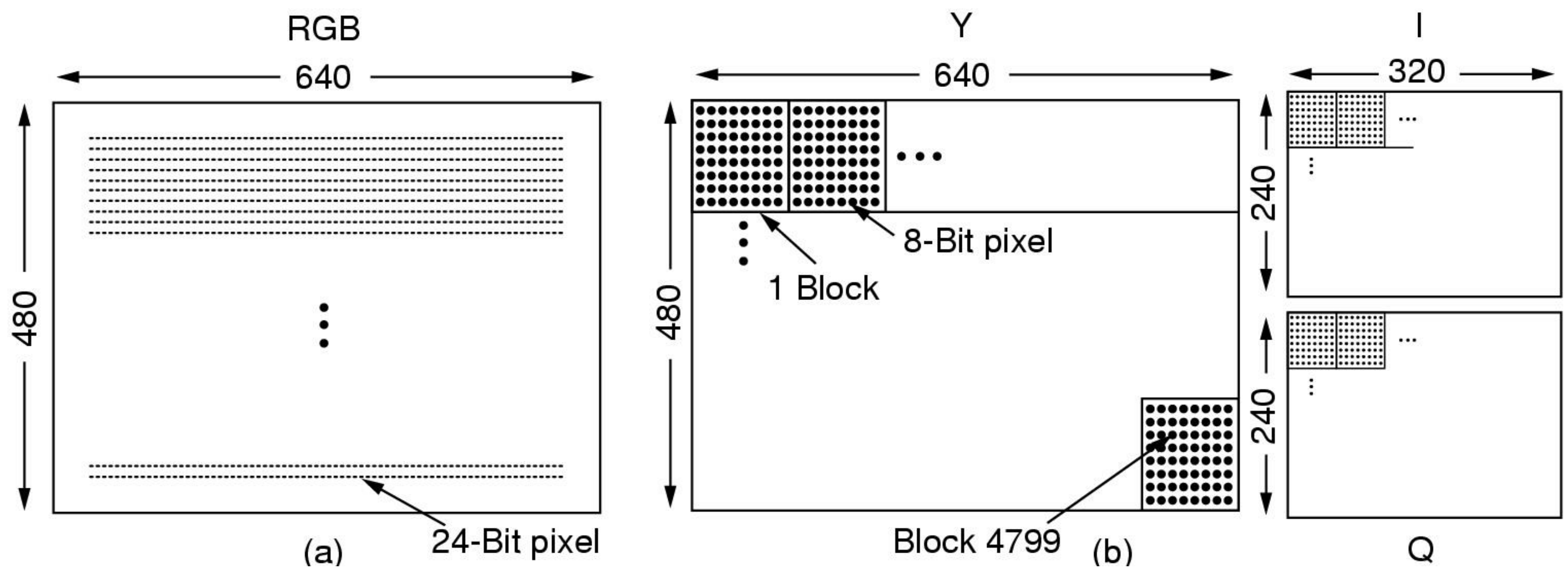
JPEG - Compression Steps



JPEG - Compression Steps

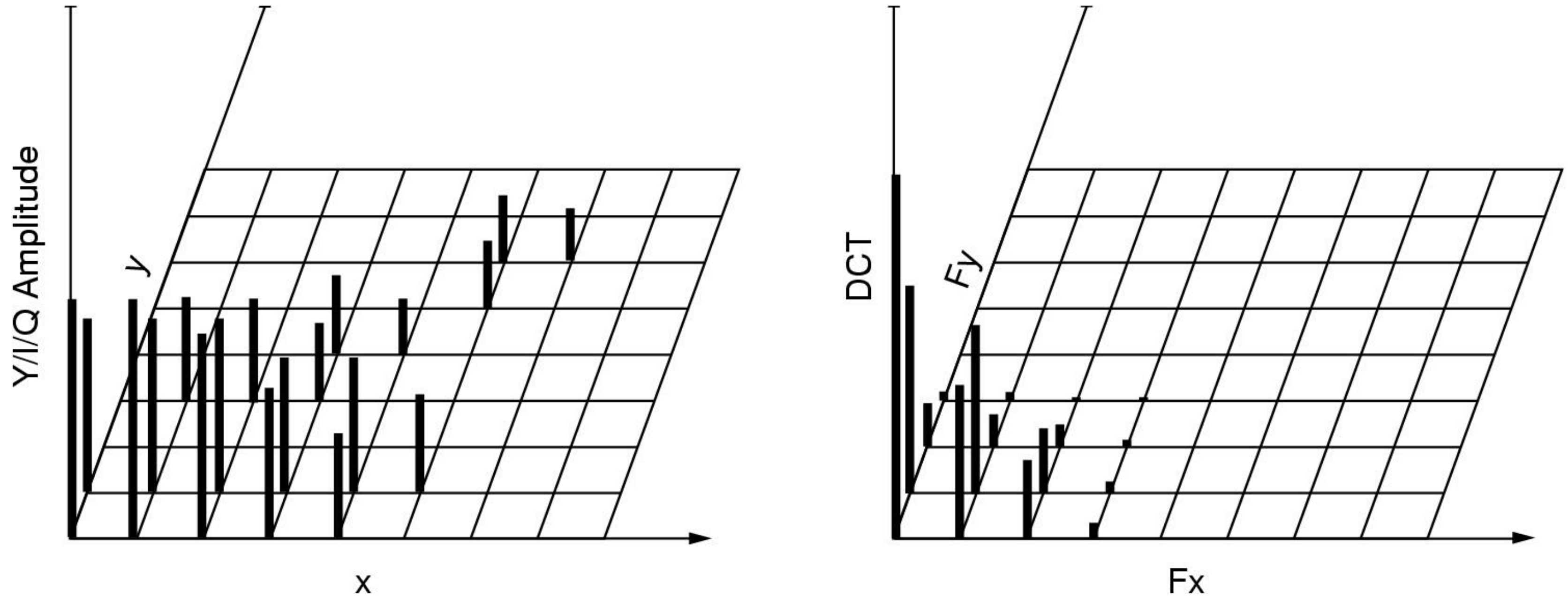
- Step 1: Block preparation
- Step 2: DCT (Discrete Cosine Transform) transformation
- Step 3: Quantization
- Step 4: Zig-zag scanning of coefficients
- Step 5: RLE compression and Huffman coding

Step 1: Block Preparation



Step 2: DCT Transformation

For each block, do DCT transformation



$$F(u,v)=\frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)*\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right]$$

Step 3: Quantization

DCT Coefficients

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized coefficients

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantization table

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

- Computation of the quantized DCT coefficients
- LOSSY COMPRESSION occurs in the step!
- (0,0) Coefficient is DC Coefficient
- Other (i,j) coordinates are AC Coefficients

Step 4: Zig-Zag Organization (from matrix to vector)

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Step 5: RLE and Huffman Coding

- Take the Zig-zag vector, apply RLE (run length encoding) and then apply Huffman coding on the values in the vector

Video Compression

- Input to any video compression algorithm is a sequence of bitmapped images (The digitalized video)
- *Spatial (intra-frame) compression and temporal (inter-frame) compression are used together in most contemporary video codecs.*
  - ***Spatial compression****:* each individual image can be compressed in isolation
  - ***Temporal compression***: sub-sequences of frames can be compressed by only storing the differences between them.

Compression and Decompression

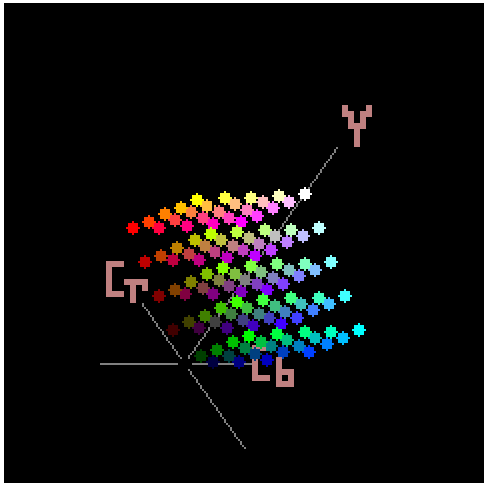
- Compression and decompression need not take the same time
- If they do, the codec is said to be *symmetrical*, otherwise it is *asymmetrical*.
- Codecs which take much longer time to decompress video than to compress it are essentially useless.
  - since playback must take place at a reasonably fast frame rate.

Spatial Compression

- Spatial compression is image compression applied to a sequence of bitmapped images.
- It could be lossy or lossless.
- Lossy compression
  - Can deteriorate the image quality.
  - Can provide high compression ratios to reduce video data to manageable proportions.

Spatial Compression

- Spatial compression of individual video frames is usually based on a Discrete Cosine Transformation, like JPEG.
- JPEG compression is applied to the three components of a colour image separately.
- Video data is usually stored using $Y'C_BC_R$ colour, with chrominance sub-sampling.



Applications

- **Motion JPEG or MJPEG**
  - The technique of compressing video sequences by applying JPEG compression to each frame
  - MJPEG was formerly the most common way of compressing video.
- **DV compression**
  - It is purely spatial.
  - It extends the JPEG technique by using a choice of sizes for transform blocks.

Temporal Compression

- Certain frames in a sequence are designated as key frames.
- Key frame occurs at regular intervals.
    - These key frames are either left uncompressed or only spatially compressed.
- Each frames between the key frames is replaced by a difference frame.
    - which records only the differences between the frame which was originally in that position and either the most recent key frame or the preceding frame.
    - The differences will only affect a small part of the frame.

Temporal Compression

Subtract the corresponding pixel values in each frame.

First Frame

Second Frame

Frame difference

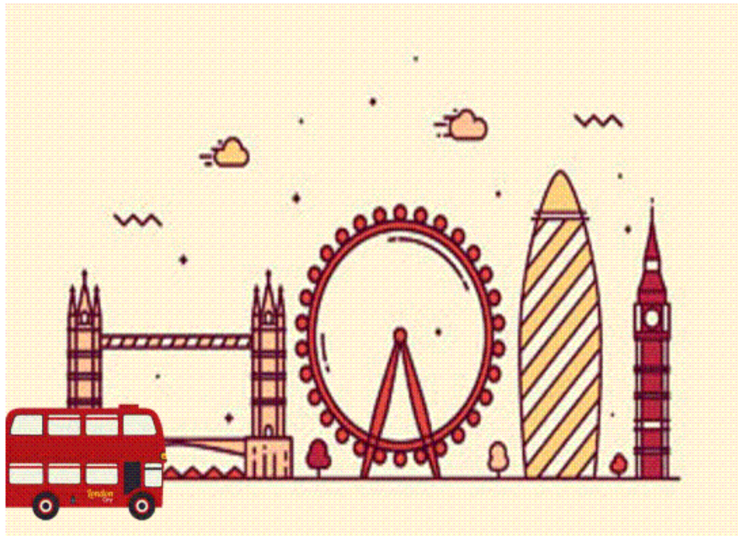
Temporal Compression

- MPEG
  - first expressed systematically broad principles to achieve either much higher compression ratios, or better quality at the same ratio, relative to DV or MJPEG.
  - It combines temporal compression based on *motion compensation* with spatial compression.
  - Temporal compression works by computing the difference between frames instead of storing every one in full.
  - In MPEG terminology, *I-pictures (Intra)* are only spatially compressed. *P-pictures (predictive)* are computed from a preceding I- or P-picture.

Motion Compensation

- **Motion compensation** is the technique of incorporating a record of the relative displacement of objects in the difference frames, as a motion vector.
- In existing codecs, motion compensation is applied to macroblocks (16*16 pixels), since coherent objects cannot usually be identified.
- Although basing difference frames on preceding frames probably seems the obvious thing to do, it can be more effective to base them on following frames.

Motion Compensation

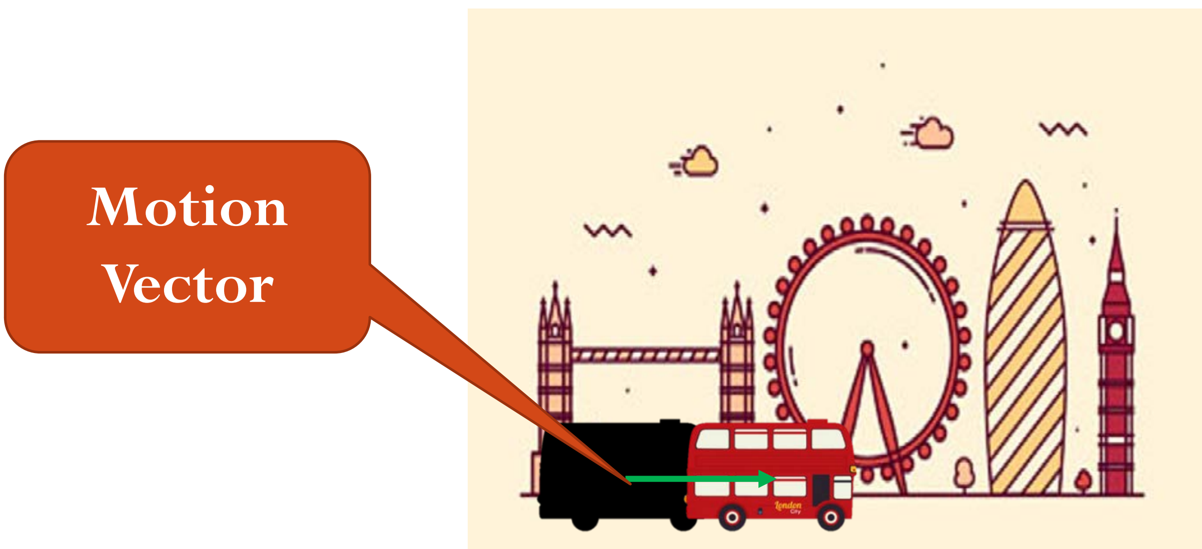


Motion Compensation

- Black areas: the difference of two consecutive frames

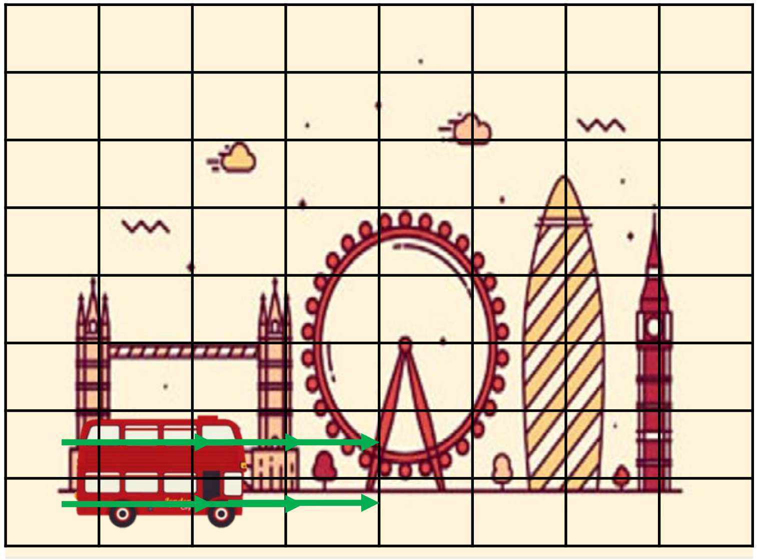
Motion Compensation

- The pixels values for the bus in the second frame are all there in the first frame.
- Record displacement of the bus by motion vector together with the changed pixels in the smaller area.



Motion Compensation

- Motion compensation based on Macroblocks



Motion vectors for Macroblocks

After applying motion compensation on Frame 1

MPEG: Motion Picture Experts Group

- MPEG-1 was designed for video recorder-quality output (320x240 for NTSC) using the bit rate of 1.2 Mbps.
- MPEG-2 is for broadcast quality video into 4-6Mbps
- MPEG takes advantage of *temporal and spatial redundancy*. Temporal redundancy means that two neighboring frames are similar, almost identical.
- MPEG-2 output consists of three different kinds of frames that have to be processed:
  - I (Intracoded) frames - self-contained JPEG-encoded still pictures
  - P (Predictive) frames - Block-by-block difference with the last frame
  - B (Bidirectional) frames - Differences with the last and next frames

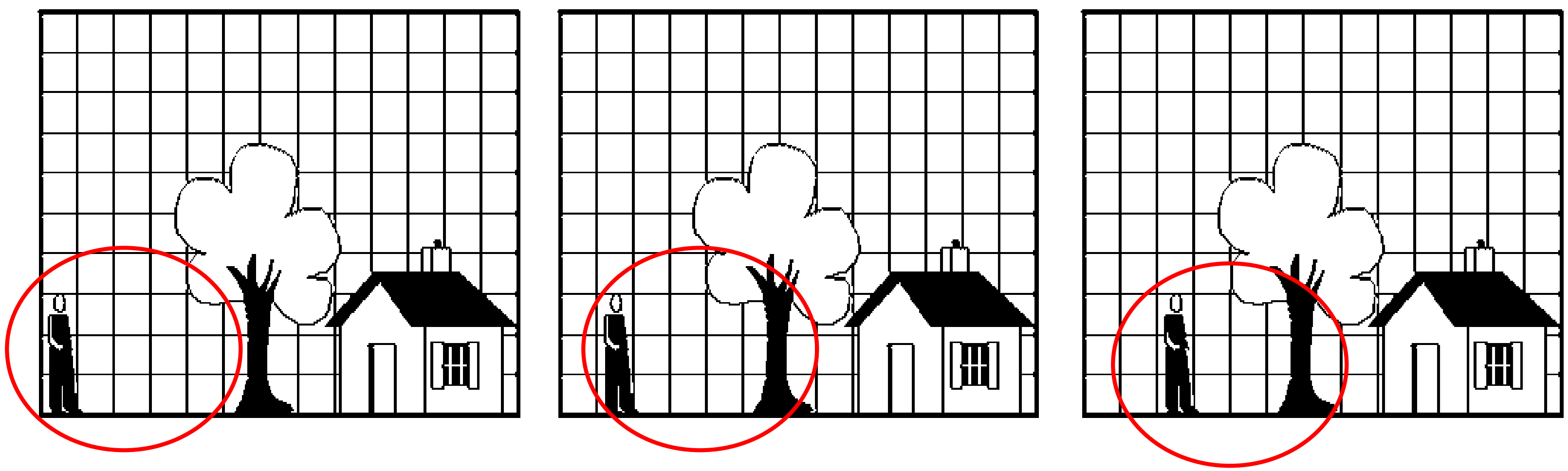
The MPEG Standard

- Suitable for stored video because it is an ***asymmetric lossy*** compression.
- Encoding takes long time, but decoding is very fast.
- MPEG display sequence: IPPBBBPPIPPBBBPPI
- MPEG transmitted dependency sequence: IPPPBBBPIPPPBBBPI
- Frames are delivered at the receiver in the dependency order rather than display order, hence we need buffering to reorder the frames.

MPEG/Video I-Frames

- I frames (intra-coded images)
- MPEG uses JPEG compression algorithm for I-frame encoding
- I-frames
  - use 8x8 blocks defined within a macro-block.
  - use DCT (Discrete Cosine Transform) on blocks.
  - do Quantization by a constant value for all DCT coefficients
    - no quantization tables exist

The MPEG Standard



Consecutive Video Frames

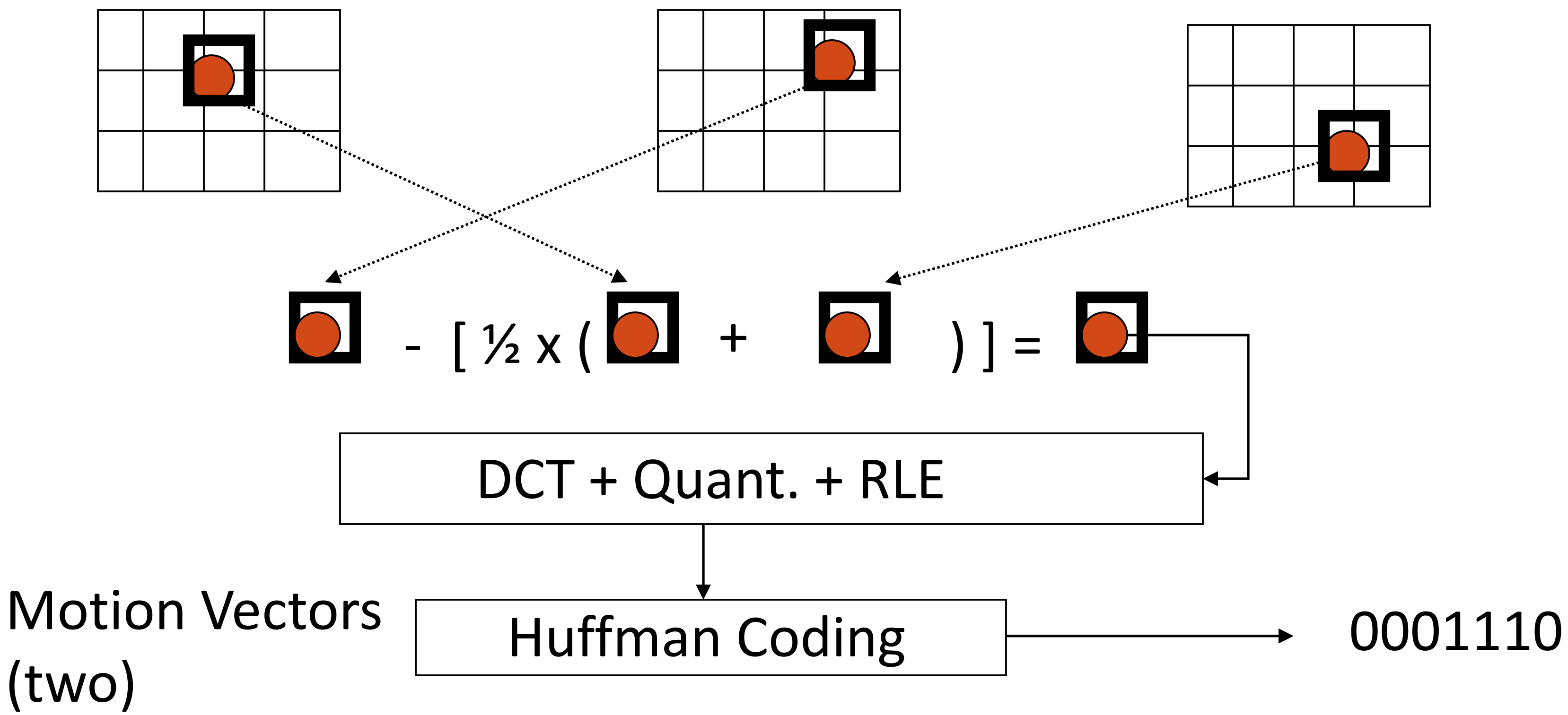
Usefulness of P frames (motion-based compression)

MPEG/Video P-Frames

- P-frames (predictive coded frames) requires previous I-frame and/or previous P-frame for encoding and decoding
- Use motion estimation method at the encoder
  - Define match window within a given search window.
  - Matching methods:
    - SSD correlation uses SSD = $\Sigma_i(x_i-y_i)^2$
    - SDA correlation uses SAD = $\Sigma_i|x_i-y_i|$

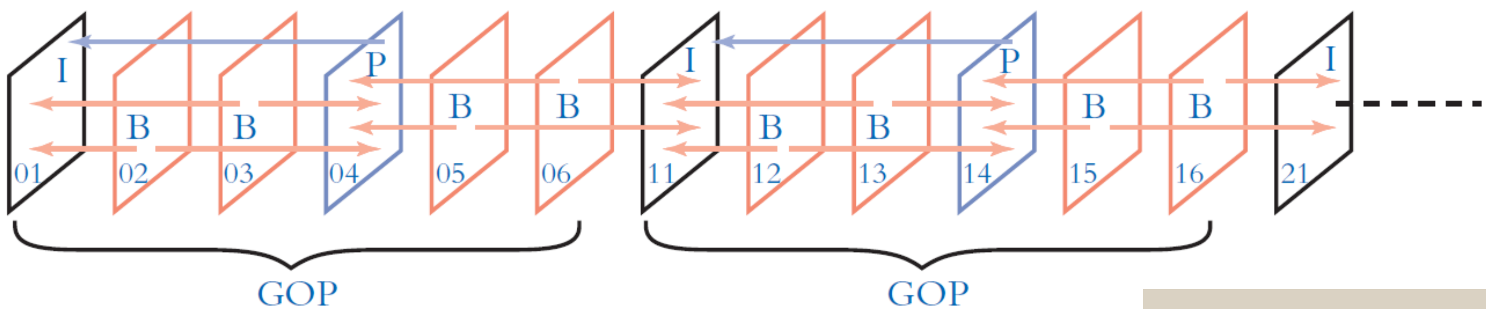
MPEG/Video B-Frame

- B-frames (bi-directionally predictive-coded frames) require information of the previous and following I and/or P-frame



MPEG Video Sequence

- A video sequence is encoded as a Group of Pictures (GOP). If B-pictures are used, a GOP may have to be reordered into display order for decoding.
- B-pictures use following pictures as well as preceding ones as the basis of frame differences and motion compensation.

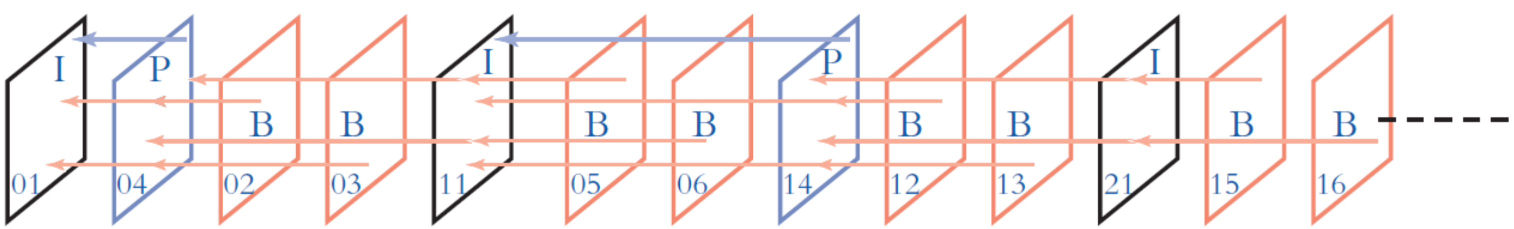


An MPEG sequence in display order

The arrows indicate the forward and bi-directional prediction.

MPEG Video Sequence

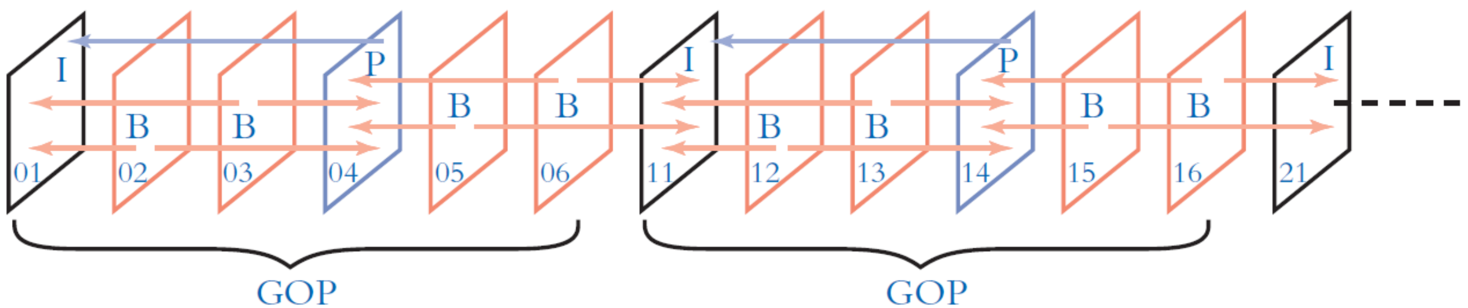
- All three types of picture are compressed using the MPEG-1 DCT-based compression method.
- P-pictures compress three times as much as I-pictures, and B-pictures one and a half times as much as P-pictures.
- Only I-picture allow random access.
- There is a trade-off to be made between compression and computational complexity when choosing the pattern of a GOP.



An MPEG sequence in bit stream order (for transmission)

Exercises

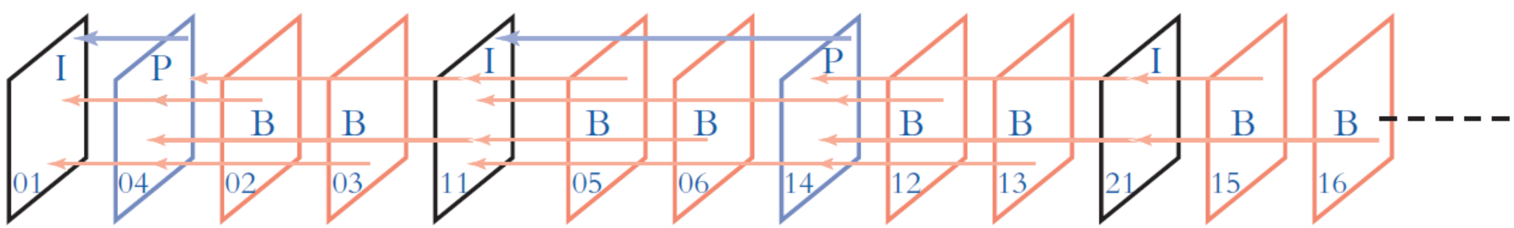
1. For a MPEG video sequence has a display order as IBBPBBIBBPBBI as below, work out the sequence in bit-stream order (for transmission) and explain why their orders are different.



2. Explain the differences of I, P and B-pictures.

Solution to 1

The key to this question is the B pictures has bi-directional referencing for motion compensation. So the following reference picture of B picture should to bring to the front.



Solution to 2

Explain the differences of I, P and B-pictures.

I-picture

is spatial compressed.

Allows random access.

Has lowest compression ratio.

P-picture

Uses motion compensation based on temporal compression

Uses preceding I or P pictures as reference.

Has higher compression ratio than I frame.

B-picture

Uses bi-directional motion compensation.

Takes following and preceding I and P-pictures as references.

Achieves highest compression ratio and highest computational complexity.

H.264/MPEG-4 AVC

- it is jointly developed by ITU-T and ISO/IEC.
- It address the full range of video applications.
  - including standard-definition and high-definition broadcast television, video streaming over the Internet, delivery of high-definition DVD content, and the highest quality video for digital cinema applications.
- Before it, MPEG-2 has gained mass-market acceptance
  - in areas such as DVD and digital television broadcast (over cable and satellite).
- The new H.264/MPEG-4 AVC standard represents the single largest improvement in coding efficiency and quality since the introduction of MPEG-2.

Steps for H.264/MPEG4 AVC

- Motion estimation and intra estimation
- Transform (inverse transform)
- Quantization (and inverse quantization)
- Loop filter
- Entropy coding

Motion Estimation

- Motion estimation is used to identify and eliminate the temporal redundancies that exist between individual pictures.
- H.264 : *smaller macroblocks* to contain and isolate the motion.
- H.264 : *multiple reference frames.*

Intra Estimation

- Intra estimation is used to eliminate spatial redundancies.
- Intra estimation attempts to predict the current block by extrapolating the neighboring pixels from adjacent blocks in a defined set of different directions.
- The difference between the predicted block and the actual block is then coded.

Transform

- H.264/MPEG-4 AVC uses a DCT-like 4x4 integer transform.
- In contrast, MPEG-2 and MPEG-4 ASP employ a true DCT 8x8 transform that operates on floating-point coefficients.
- Smaller block size reduces blocking and ringing artefacts.

Quantization

- The coefficients from the transform stage are quantized, which reduces the overall precision of the integer coefficients and tends to eliminate high frequency coefficients, while maintaining perceptual quality.

Loop Filter

- H.264 has a de-blocking filter for Macroblocks to remove artifacts

Entropy Coding

- Lossless encoding
- Examples: run-length coding, Huffman coding, arithmetic coding

Entropy Encoding

- All compression systems require two algorithms:
  - Encoding at the source
  - Decoding at the destination
- Simple lossless compression algorithm is the **Run-length Encoding (RLE)**, where multiple occurring bytes are grouped together.
- What is the compression ratio of AAAAAAAABBBCCCCCCCDD?

Entropy Encoding

- We wish to construct and transmit a message using N symbols.
- Simplest method: use binary numbers of equal length L bits to represent each symbol ($L \geq log_2(N)$).
  - *Example: consider message of 5 symbols, then L will be approximately 3 bits ($L \geq log_2(5)$)*
  - This type of coding is called **fixed-length coding**
- **Fixed-length coding is not efficient – need statistical coding**

Entropy Encoding

- Statistical encoding:
  - Given sequence of symbols: s_1, s_2, …and probability of occurrence of each symbol $P(s_i)=P_i$
  - Example: P(A) = 0.16, P(B)=0.51, P(C)=0.09, P(D)=0.13, P(E)=0.11
    - Fixed-length coding encode A,B,C,D, E with 3 bits as A=000, B=001, C=010, D=011, E=100
  - Question: What is the minimum average number of bits per symbol in statistical encoding?
- The theoretical minimum average number of bits per codeword is known as **ENTROPY(H)**
- According to Shannon:- $\sum_i P_i log_2 P_i$ bits per codeword
  - Example above: - (0.16 x $\log_2$0.16 + 0.51 x $\log_2$0.51 + 0.09 x $\log_2$0.09 + 0.13 x $\log_2$0.13 + 0.11 x $\log_2$0.11) = 1.3 bits per codeword (approx)
  - This type of coding represents the **variable-length coding**

Huffman Encoding

- Statistical encoding
- To determine Huffman code, it is useful to construct a binary tree
- Leaves are characters to be encoded
- Nodes carry occurrences of the characters belonging to the subtree

- How does a Huffman code look like for symbols with statistical symbol occurrence probabilities:

$P(A) = 8/20$, $P(B) = 3/20$, $P(C) = 7/20$, $P(D) = 2/20$?

Summary

- Compression is very important due to the large amount of multimedia data, especially in video (e.g., HDTV)
- Consider lossless entropy coding schemes if you do not want to lose any data
- Keep in mind symmetric vs asymmetric compression schemes

Further Exploration

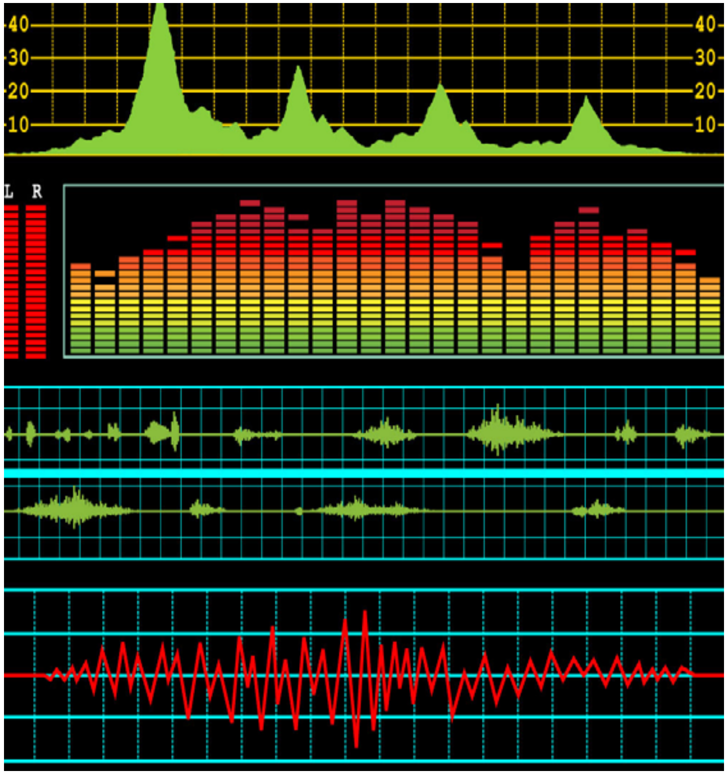
- JPEG Compression
- JPEG tutorial: http://johnloomis.org/ece563/notes/compression/jpeg/tutorial/jpegtut1.html
- JPEG online converter: http://www.jpeg-optimizer.com/

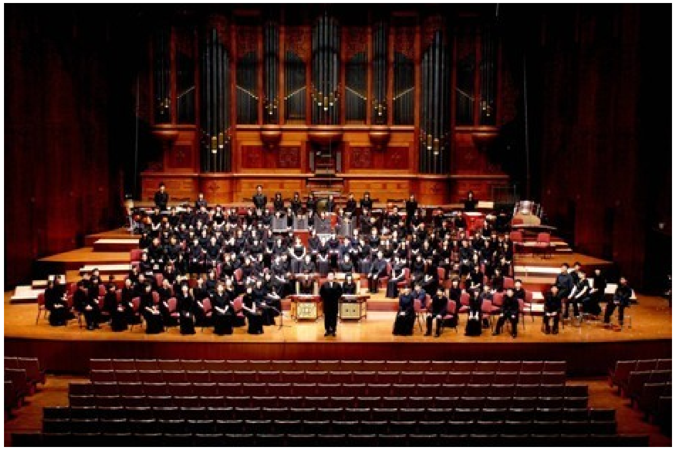
- MPEG-4 and H.264/AVC
- http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf

Basics of Digital Audio

- The nature of sound
- Sound Digitalization
  - Sampling
  - Quantization
  - Compression
- Sound Processing
- Audio conversion
- File formats
- Further exploration

Sound





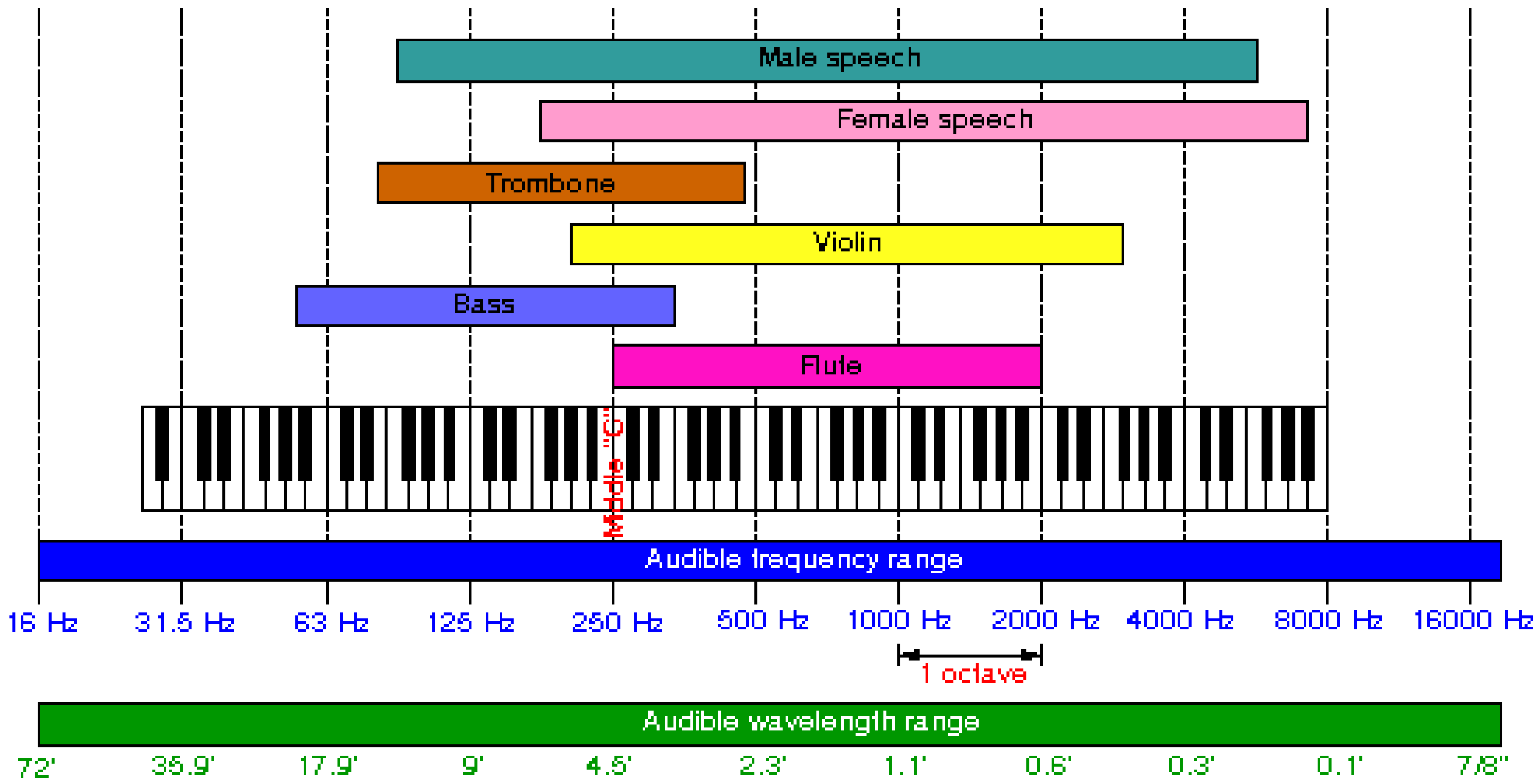
Sound waves

- Sound is a range of wave frequencies to which the human ear is sensitive, it will not travel through a vacuum.
- Sounds are produced by vibrating matter
- The audio spectrum extends from approximately *20 Hz to 20,000 Hz*.

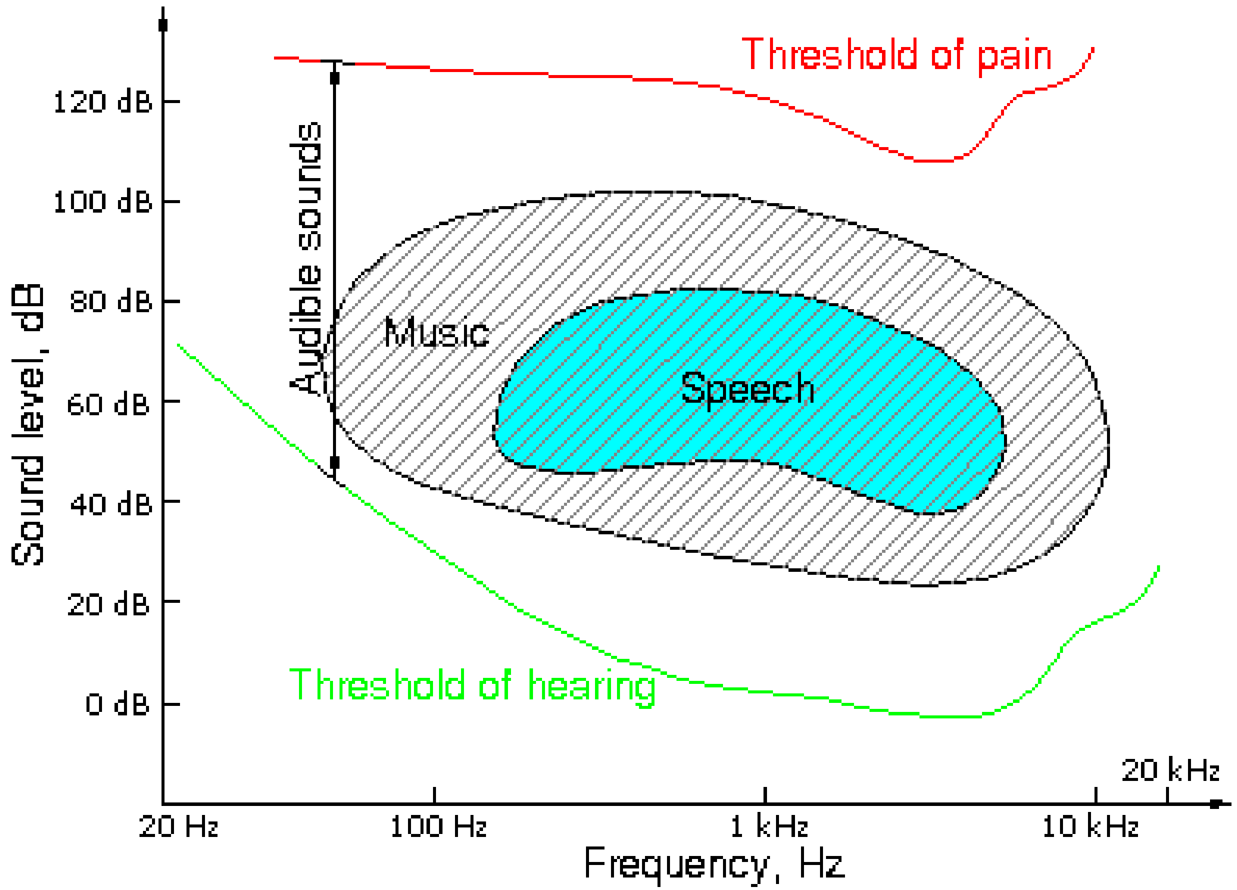
Sound waves

- Sounds are produced by vibrating matter.
  - eg. Speaker vibrates back and forth and produces longitudinal pressure wave.
- Sounds are transmitted via elastic medium like air.
  - No sound in space.
- Sound is a mechanical wave (longitudinal).
  - Sounds possess the characteristics and properties that are common to all waves
  - Sound waves posses a velocity, frequency, wavelength, phase, period and amplitude.
  - Sound waves reflect(bouncing), refract(changing angle) and diffract(bending around).

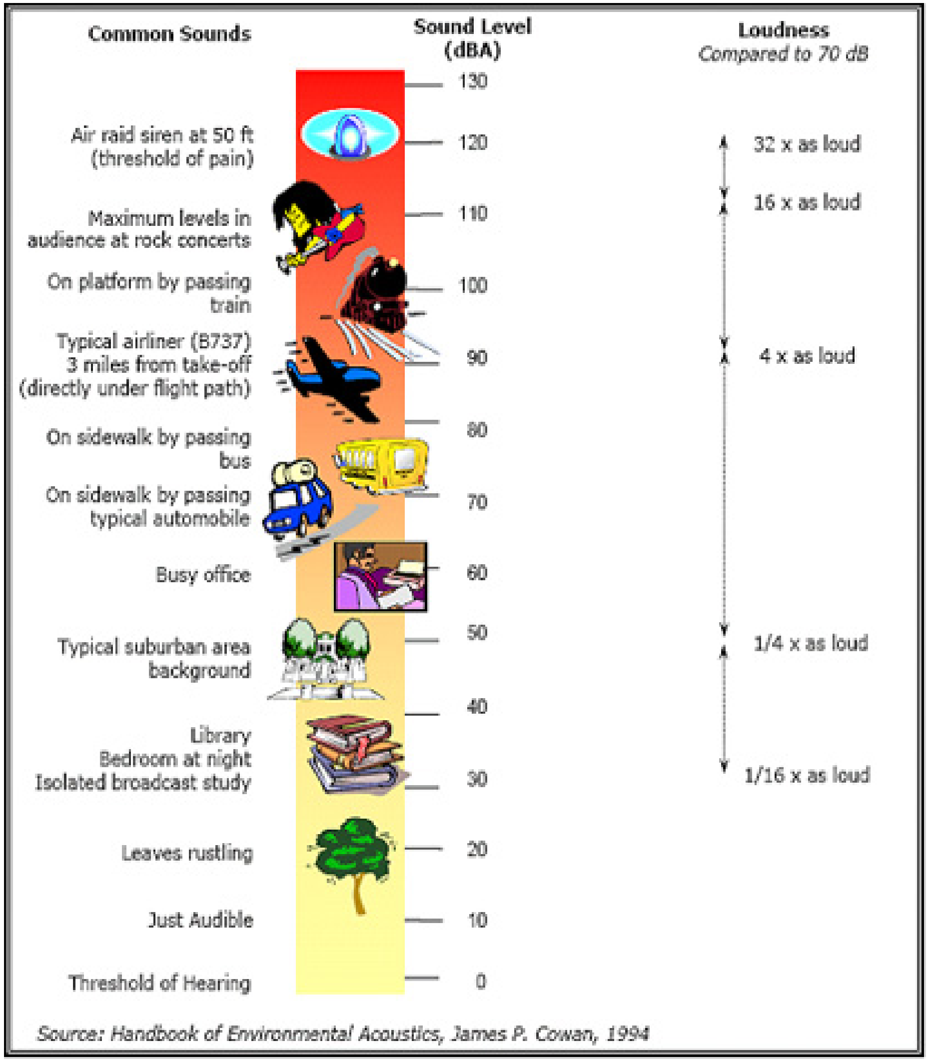
Range of some common sounds



Intensity Range for Some Common Sounds



Intensity Range for Some Common Sounds



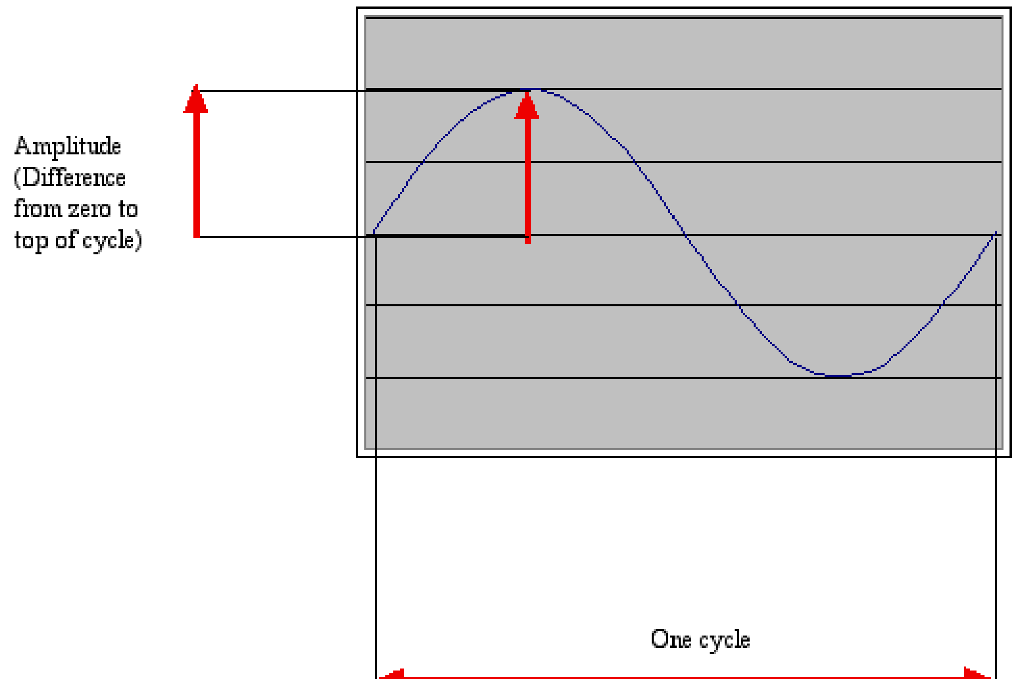
Sound waves

- The velocity of sound in air depends on the air temperature.
  - eg. 311.5 m/s at 0 °C in dry air.
  - Velocity increases 0.6m/s for every 1°C increased in temperature.
- Sound generally travels fastest in solids and slowest in gases but there are exceptions.

Medium	Velocity (m/s)	Medium	Velocity (m/s)
Air	330	Carbon dioxide	260
Helium	930	Hydrogen	1270
Oxygen	320	Water	1460
Sea water	1520	Mercury	1450
Glass	5500	Granite	5950
Lead	1230	Pine wood	3320
Copper	3800	Aluminum	5100

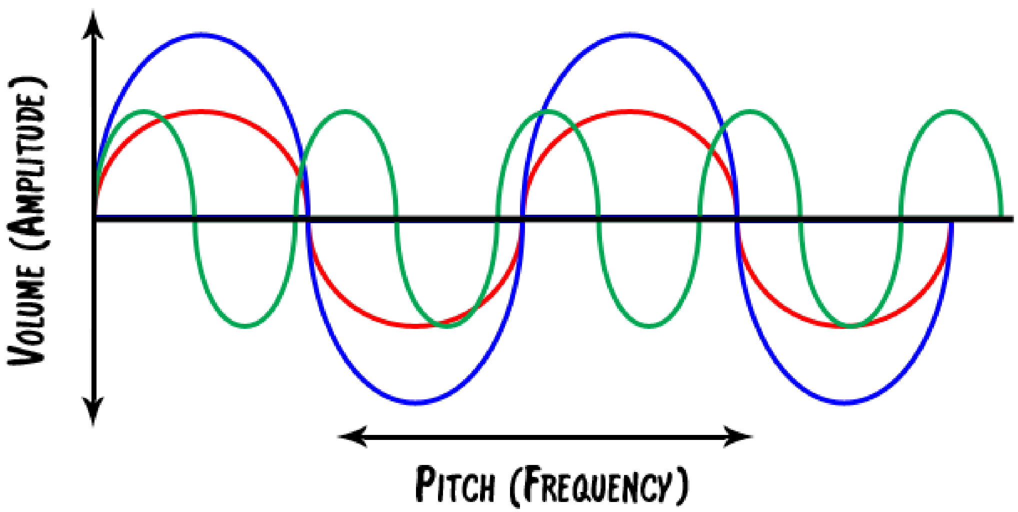
Sound waves

- Sound comes in cycles.
- The *frequency* of a wave is the number of cycles per second (cps), or *Hertz*
  - Complex sounds have more than one frequency in them.
- The amplitude is the maximum height of the wave.



Sound waves

- The human ear relates amplitude to loudness and frequency to pitch.
  - Listen to various sound frequencies here.
  - Listen to sound with frequency from 1 Hz-1000Hz here.



Decibel is a logarithmic measure

- A *decibel* is a ratio between two intensities: $10 * \log_{10}(I_1/I_2)$
  - As an absolute measure, it's in comparison to threshold of audibility
  - 0 dB can't be heard.
  - Normal speech is 60 dB.
  - A shout is about 80 dB.

Sound Recording & Audio Qualities

- We typically use a microphone to collect sounds. In such cases, environmental noises, and many other factors need to be considered. (omitted)

- In our course, we mainly consider the subsequent **sound storage**, **sampling** and **quantization** problems.

- For reproduction quality, we also focus on the effects brought by different sampling and quantization methods.

http://en.wikipedia.org/wiki/Sound_recording_and_reproduction

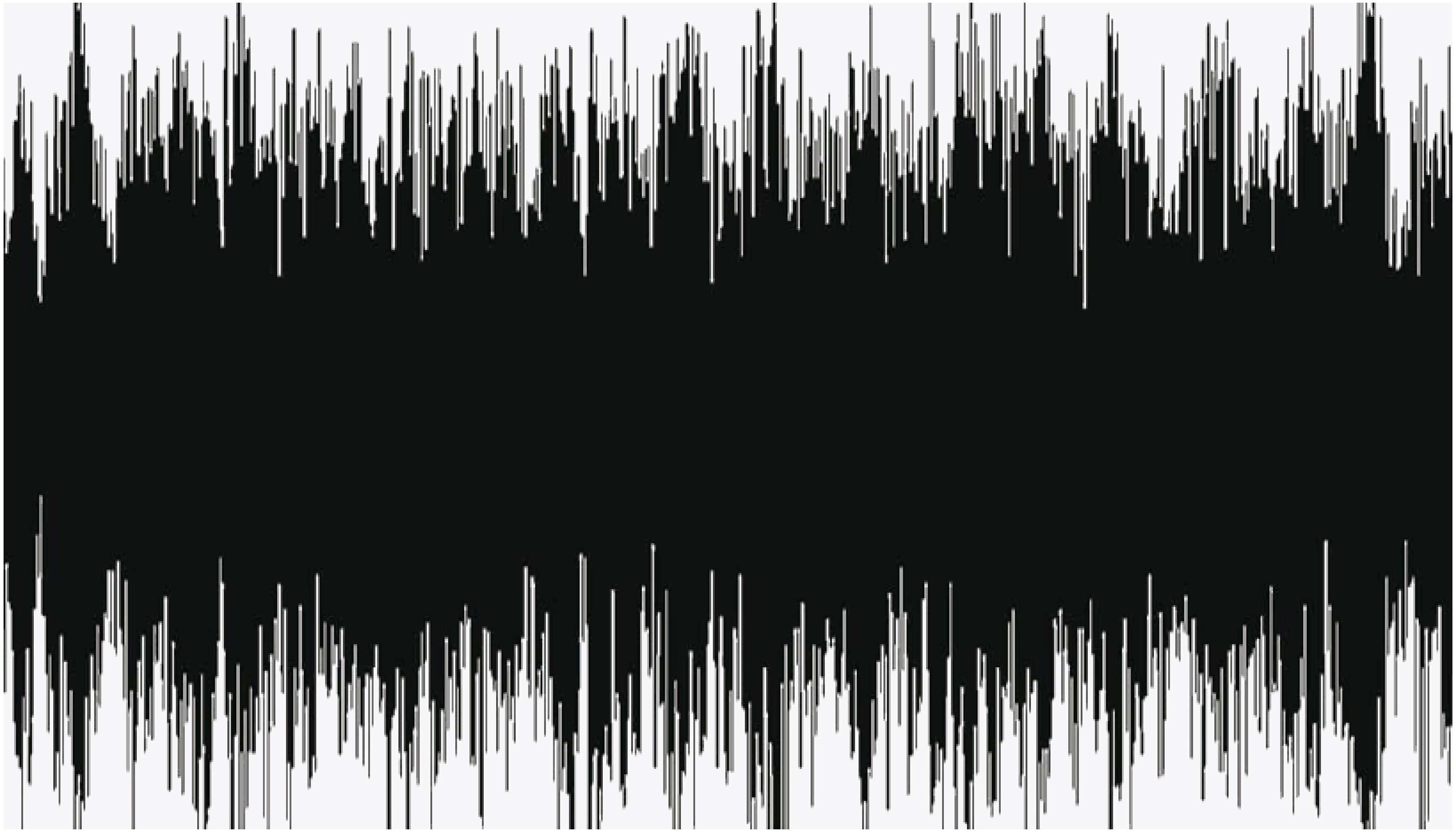
Identifying a waveform?

- A sound's waveform shows how its amplitude varies over time.

Speech: "Feisty teenager"

Men grow cold, as girls grow old
And we all lose our charms in the end

by **Marilyn Monroe**



Music: Didgeridoo

Music: Boogie-woogie

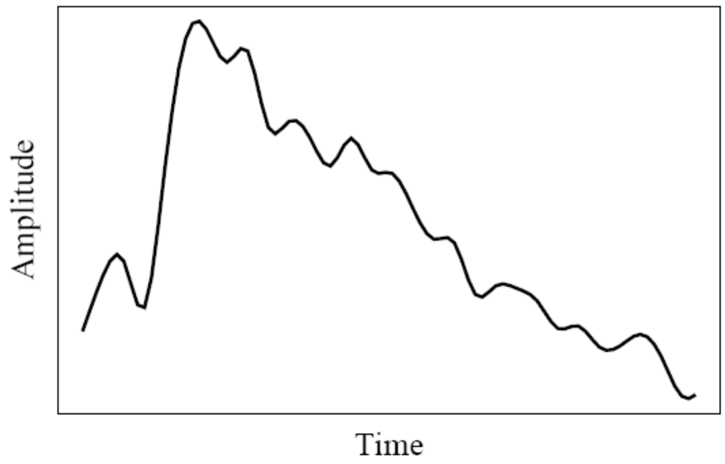
Music: Contemporary classical piece for violin, cello and piano

A trickling stream

The sea

Sound Digitization

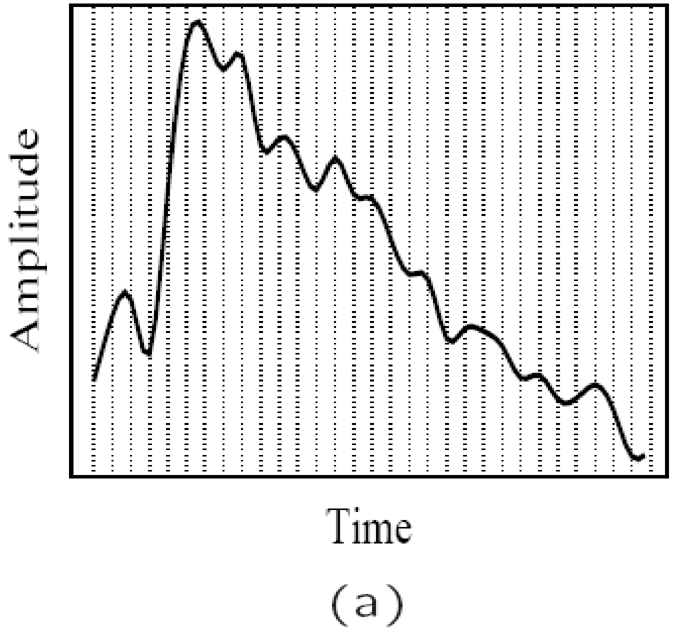
- **Digitization** means conversion to a stream of numbers, and preferably these numbers should be integers for efficiency.
- 1-dimensional nature of sound: **amplitude** values depend on a 1D variable, **time**.

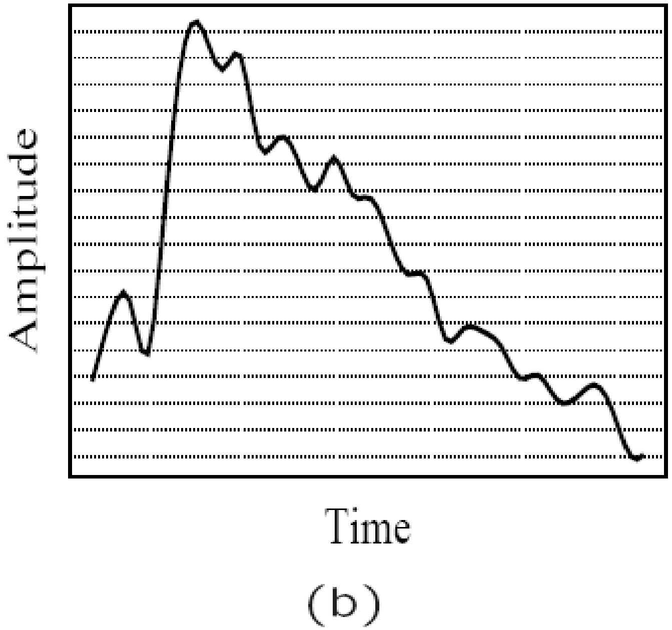


Sound Digitization

- Digitization must be in both time and amplitude.
  - Measuring the quantity we are interested in, usually at evenly-spaced intervals.
- First, using measurements only at evenly spaced time intervals, is simply called ***sampling***. The rate at which it is performed is called the *sampling frequency*
  - For audio, typical sampling rates are from 8 kHz (8,000 samples per second) to 48 kHz. This range is determined by Nyquist theorem discussed later.
- Second, get samples in the amplitude or voltage dimension is called **quantization**

Sampling and Quantization



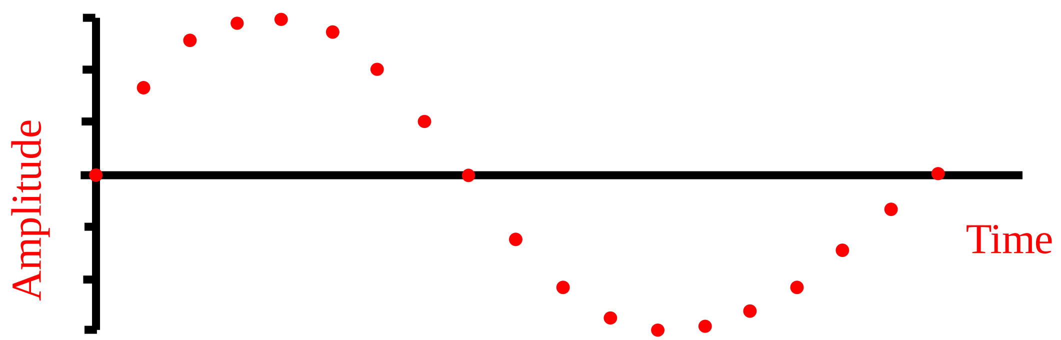


Sampling

Quantization

Digitization-Sampling

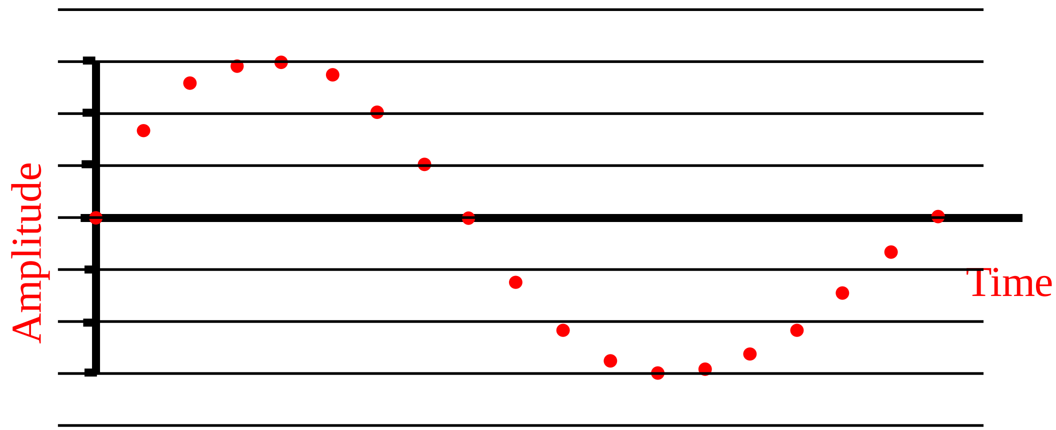
- Sampling
  - Divide the time axis into discrete pieces.
- Sampling rate
  - Number of samples per second (measured in Hz)



Sampling

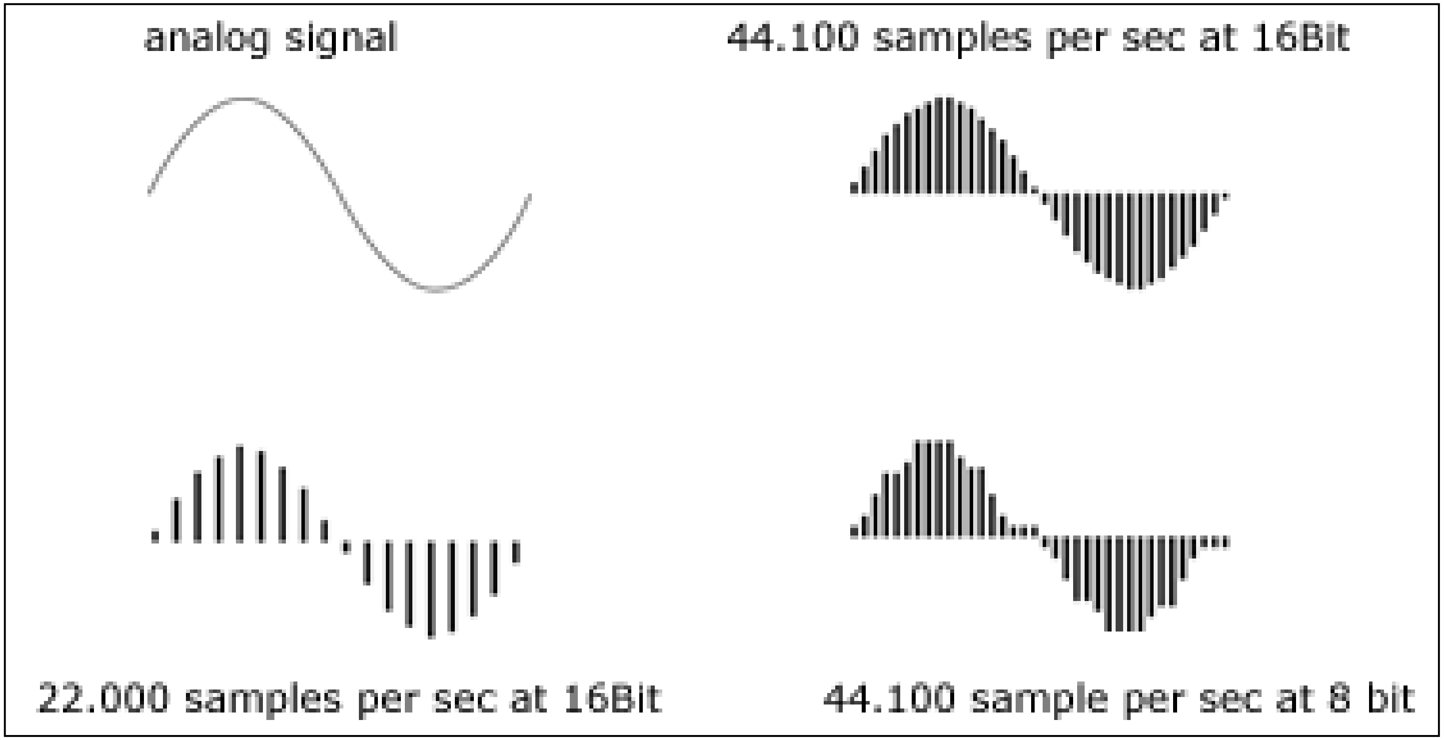
Digitization-Quantization

- Quantization
  - Divide the vertical axis (signal strength - voltage) into pieces
    - 8-bit quantization divides the vertical axis into 256 levels
    - 16-bit → 65536 levels.
  - The lower the quantization→ the lower the quality of the signal
- Example
  - 3-bit quantization →8 possible sample values



Quantization

Audio Digitization (PCM)



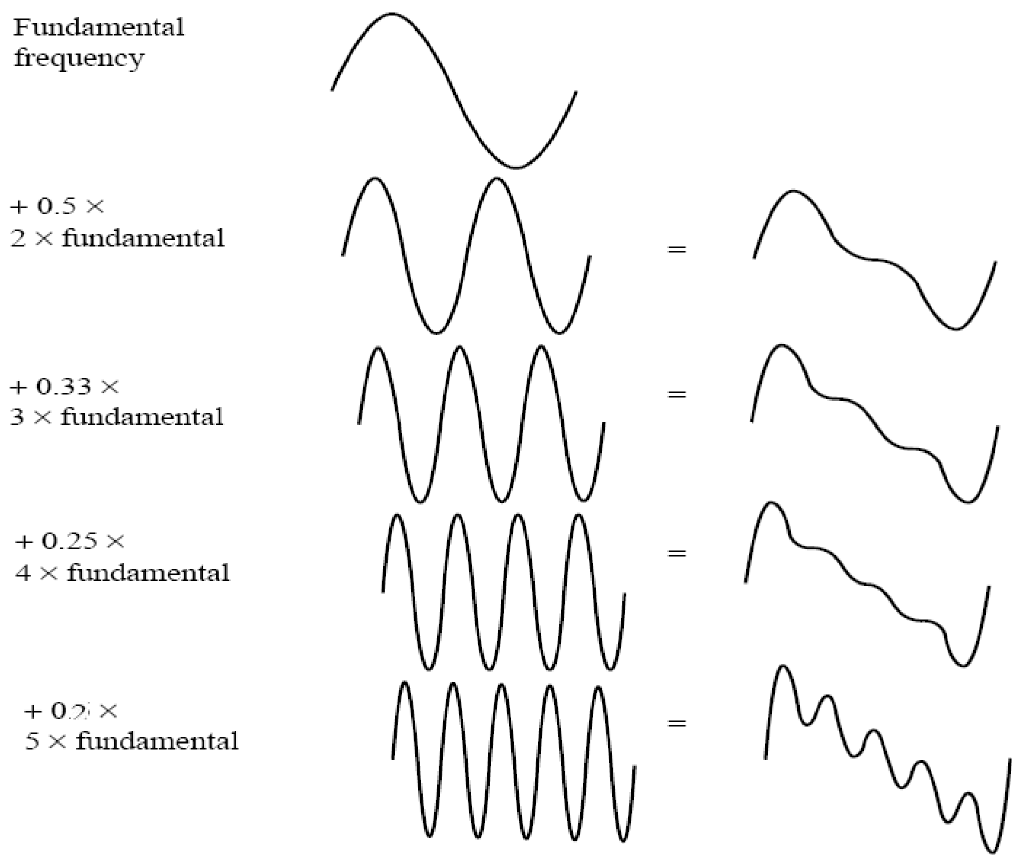
PCM: Pulse coded modulation

Parameters in Digitizing

- To decide how to digitize audio data we need to answer the following questions:
  1. What is the sampling rate?
  2. How finely is the data to be quantized, and is quantization uniform?
  3. How is audio data formatted? (file format)

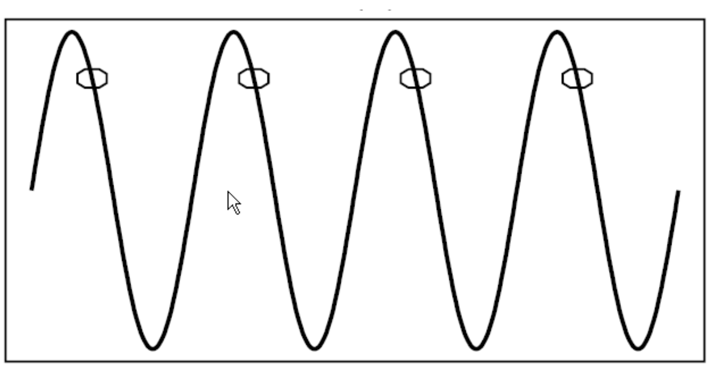
Sampling

- Signals can be decomposed into a sum of sinusoids.
- Weighted sinusoids can build up quite a complex signals.

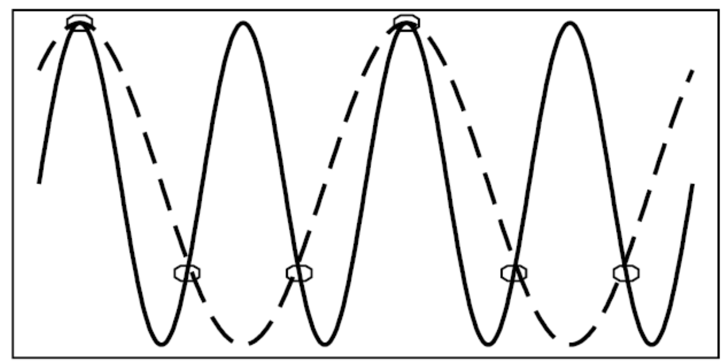


Sampling Rate cont'd

- If sampling rate just equals the actual frequency
    - a false signal (constant) is detected
- If sample at 1.5 times the actual frequency
    - an incorrect (**alias**) frequency that is lower than the correct one
        - it is half the correct one -- the wavelength, from peak to peak, is double that of the actual signal.



Sampling rate=f



Sampling rate=1.5*f

Nyquist Theorem

- For correct sampling we must use a sampling rate equal to **at least *twice the maximum frequency* content** in the signal, in order to recover it in the future. This rate is called the **Nyquist rate**.
- Sampling theory – Nyquist theorem

  If a signal is **band-limited**, i.e., there is a lower limit f_1 and an upper limit f_2 of frequency components in the signal, then the sampling rate should be at least $2(f_2 - f_1)$.

Example: CD

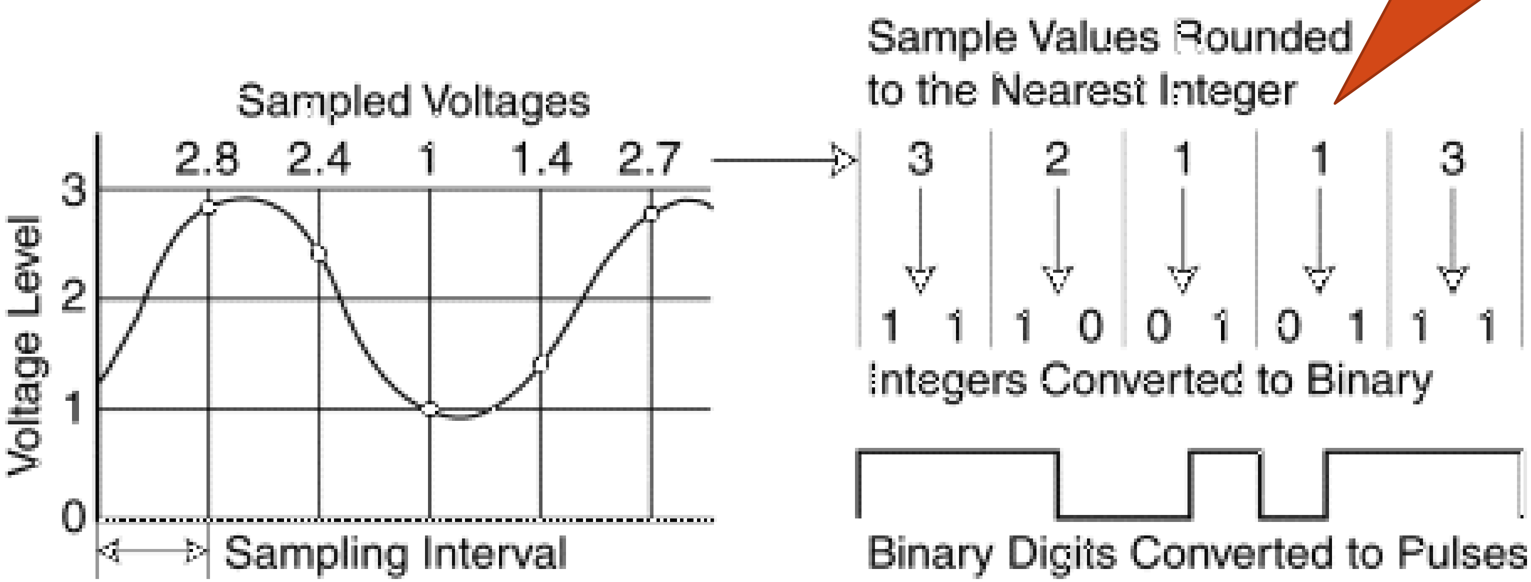
- CD audio is sampled at **44.1kHz**.
- Sampling relies on highly accurate clock pulses to determine the sample intervals.
  - If the clock drifts, Jitter (timing variations) occurs.
  - For CD quality, the jitter < 200 picoseconds.

- Frequencies greater than half the sampling rate are filtered out to avoid aliasing.

Quantization
(Pulse Code Modulation)

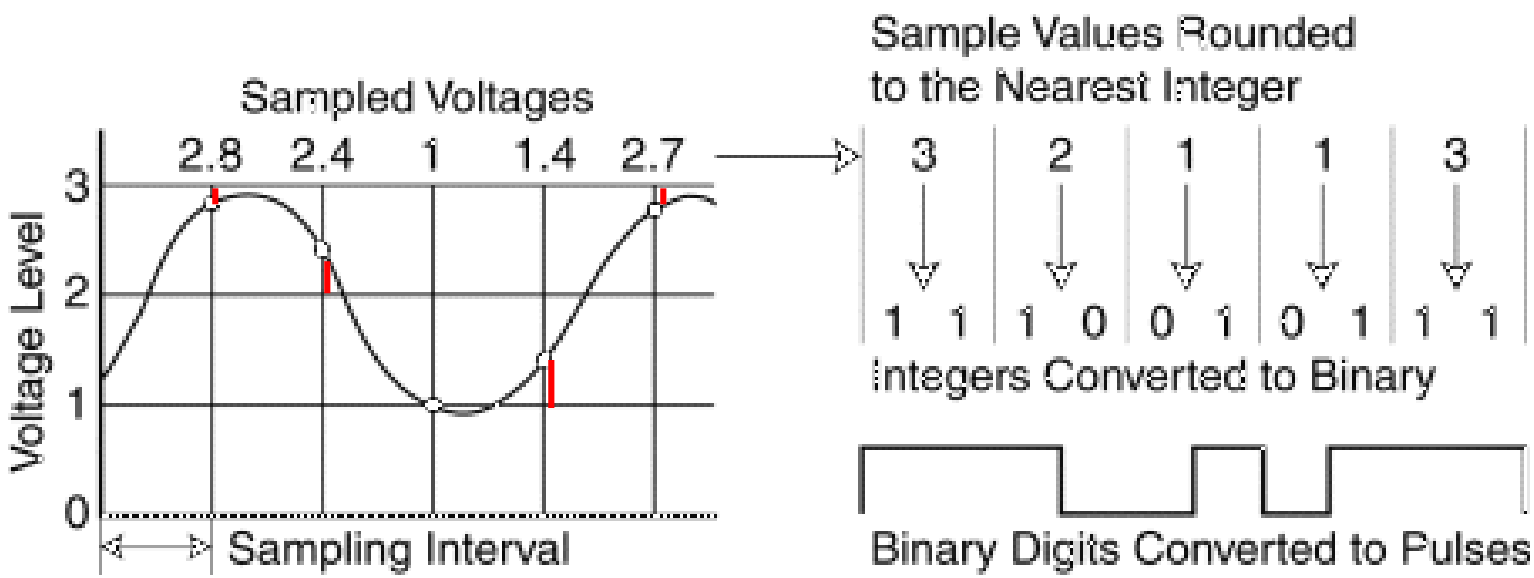
- At every time interval the sound is converted to a digital equivalent.
- Using 2 bits the following sound can be digitized

Quantization brings in Noise!



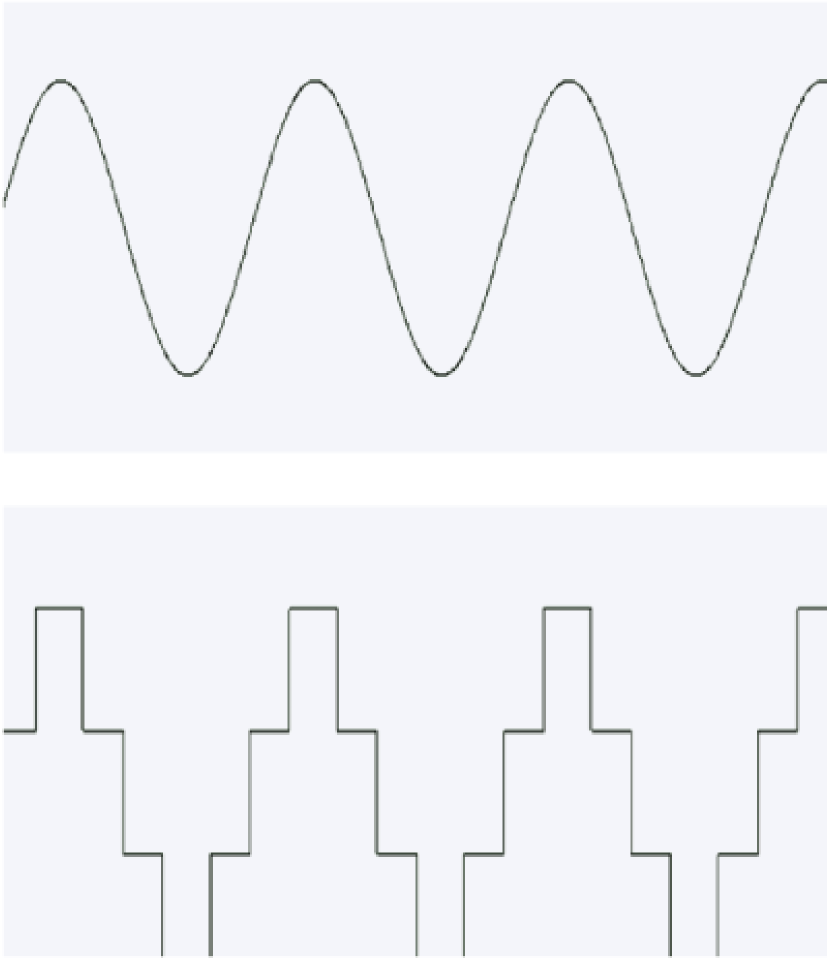
More on Quantization

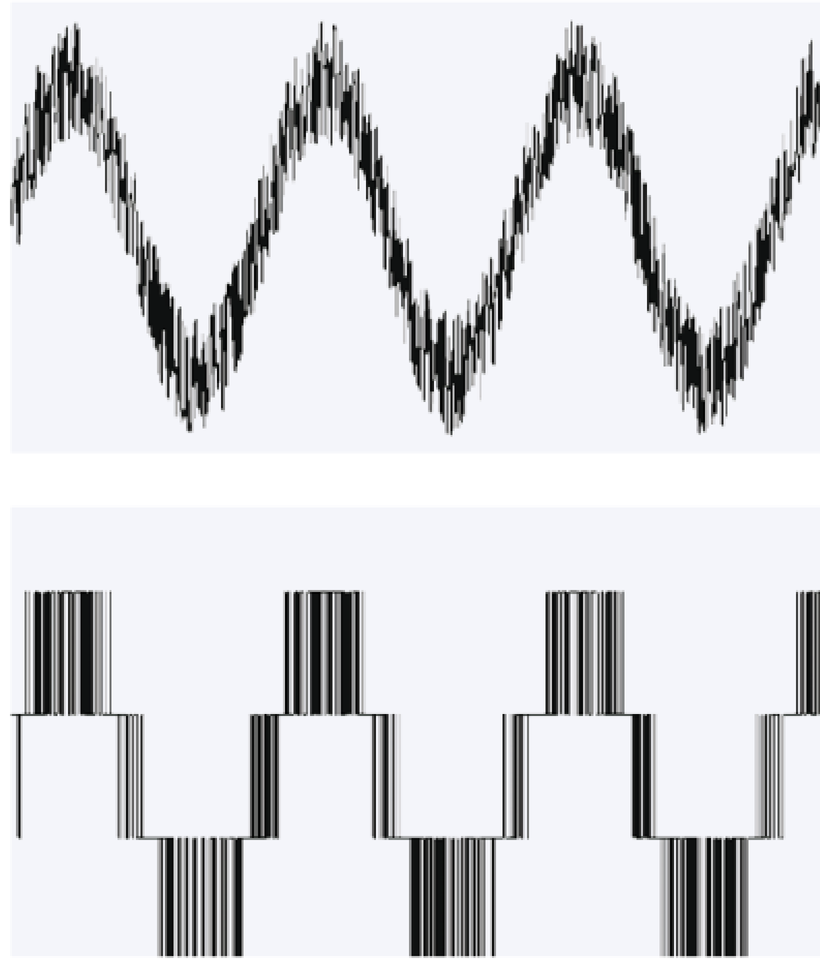
- Quantization is lossy
- Roundoff errors => **quantization noise/error**



Example: CD

- CD audio uses 16-bit samples to give 65,536 quantization levels.
- Quantization noise can be sufficiently eliminated by dithering.
  - i.e. Dithering here means adding a small amount of random noise which softens the sharp transitions of quantization noise.





Undersampling a pure sine wave

Dithering

Audio Quality vs. Data Rate

Quality	Sample Rate (KHz)	Bits per Sample	Mono/ Stereo	Data Rate (uncompressed) (kB/sec)	Frequency Band (KHz)
Telephone	8	8	Mono	8	0.200-3.4
AM Radio	11.025	8	Mono	11.0	0.1-5.5
FM Radio	22.05	16	Stereo	88.2	0.02-11
CD	44.1	16	Stereo	176.4	0.005-20
DAT	48	16	Stereo	192.0	0.005-20
DVD Audio	192 (max)	24 (max)	6 channels	1,200.0 (max)	0-96 (max)

Question

- For a sampling rate of r *Hz and sample size* of s *bits, each second of digitized sound will* occupy $rs/8$ *bytes.*
  - For CD quality, $r = 44.1 \times 10^3$ *and* $s = 16$, *so each* second occupies just over 88 kbytes (for a mono signal).

Question

An analog audio signal has a bandwidth from 10kHz-25kHz. To converting it to digital form:

1. What sampling rate should be used?
2. After sampling, if 8 bit quantization is used, what is the data rate of that digital signal?

- **Answer:**

  According to Nyquist Theorem, Nyquist rate=$2(f_2 - f_1)$=2*(25-10)=30kHz.

  30kHz sampling rate→ 30,000 samples/sec.

  8-bit quantization→8 bits/sample→256 quantization levels.

  The data rate= 30,000*8= 240,000 bps=240kbps.

- **Example rates**
  - CD: 1.411 Mbps
  - MP3: 96, 128, 160, 320 kbps
  - Internet telephony: 5.3 - 13 kbps

Compression

- Sound is difficult to compress using lossless methods, except for special cases.

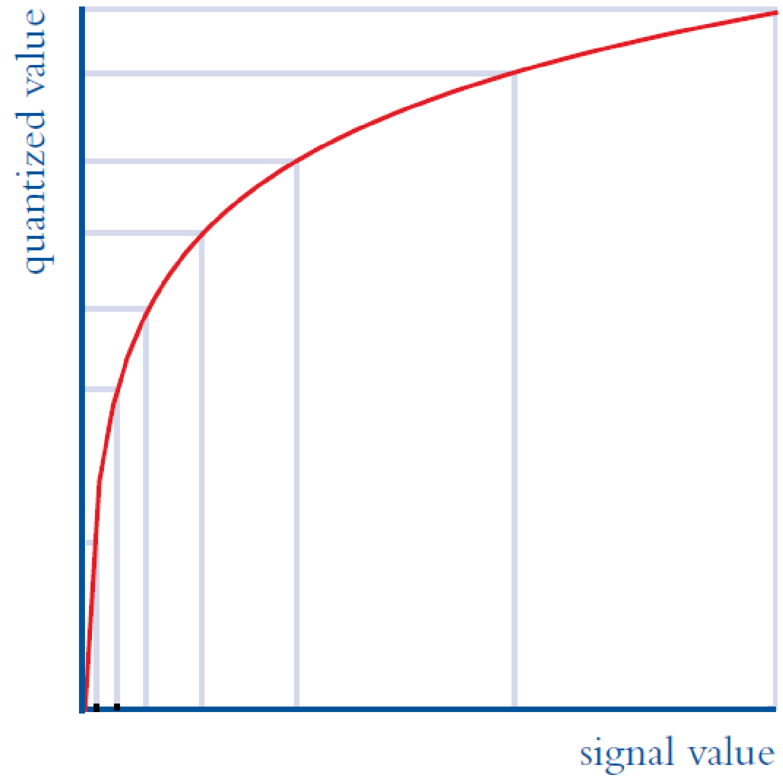
- Some compression of audio can be obtained by **run-length encoding** samples that fall below a threshold that can be considered to represent silence.
  - That is, instead of using 44,100 samples with the value of zero for each second of silence (assuming a 44.1 kHz sampling rate) we record the length of the silence.

Compression

- **Companding** uses non-linear quantization to compress speech.
  - Quiet sounds are represented in greater detail than louder ones.
- μ-law and A-law companding are used for telephony.

ITU recommendations for north America

ITU recommendations for the rest of the world



Non-linear quantization

Compression

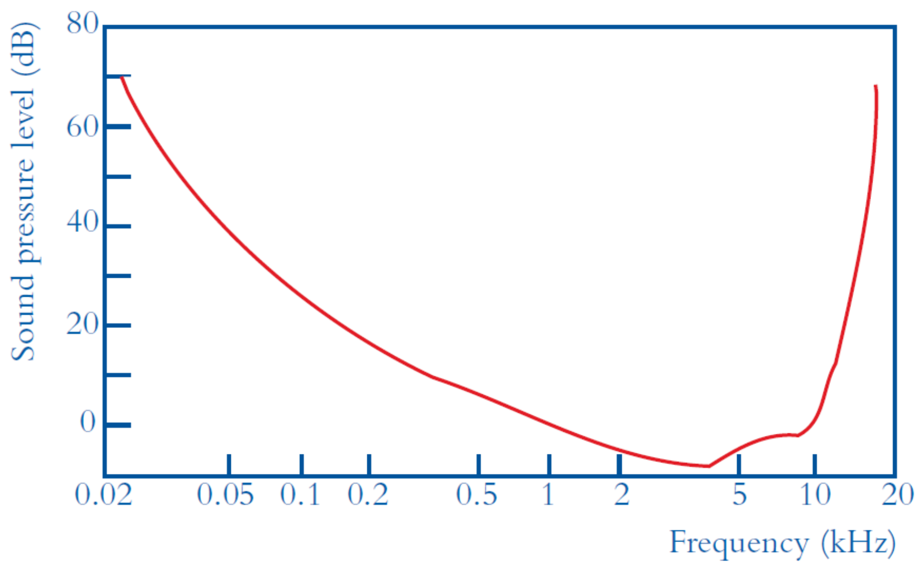
- **Adaptive Differential Pulse Code Modulation (ADPCM)**, which works by storing information about the difference between a sample and a value predicted from the preceding sample, is also used in telephony.

Compression

- **Perceptually-based compression** discards inaudible sounds.
  - Sounds may be too ‘quiet’ to be heard.
  - Sounds obscured by some other sounds.

Compression

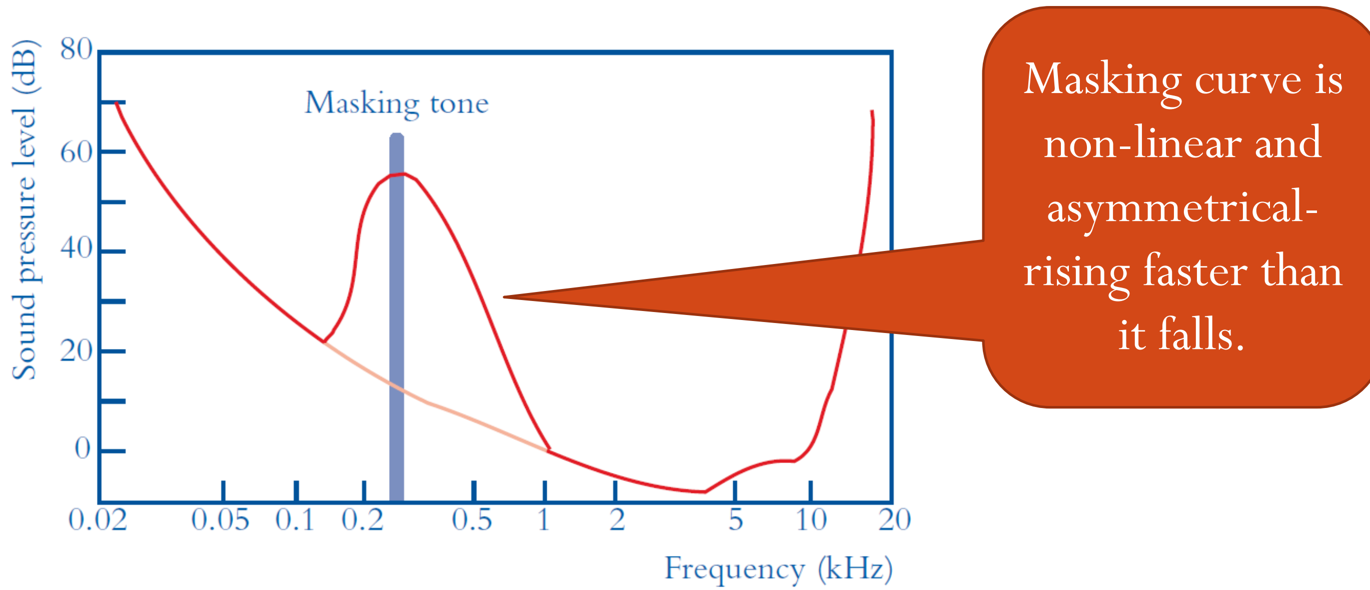
- A psycho-acoustical model describes how the threshold of hearing varies non-linearly with frequency.
  - The threshold of hearing is the minimum level at which a sound can be heard.
  - It varies non-linearly with frequency. A very-low or very high-frequency sound must be much louder than a mid-range tone in order to be heard. We are more sensitive to sounds in the frequency range that corresponds to human speech.



The threshold of hearing

Compression

- Loud tones can obscure softer tones that occur at similar time.
- **Masking** is a modification of the threshold of hearing curve in the region of a loud tone.
- The threshold is raised in the neighborhood of the masking tone.



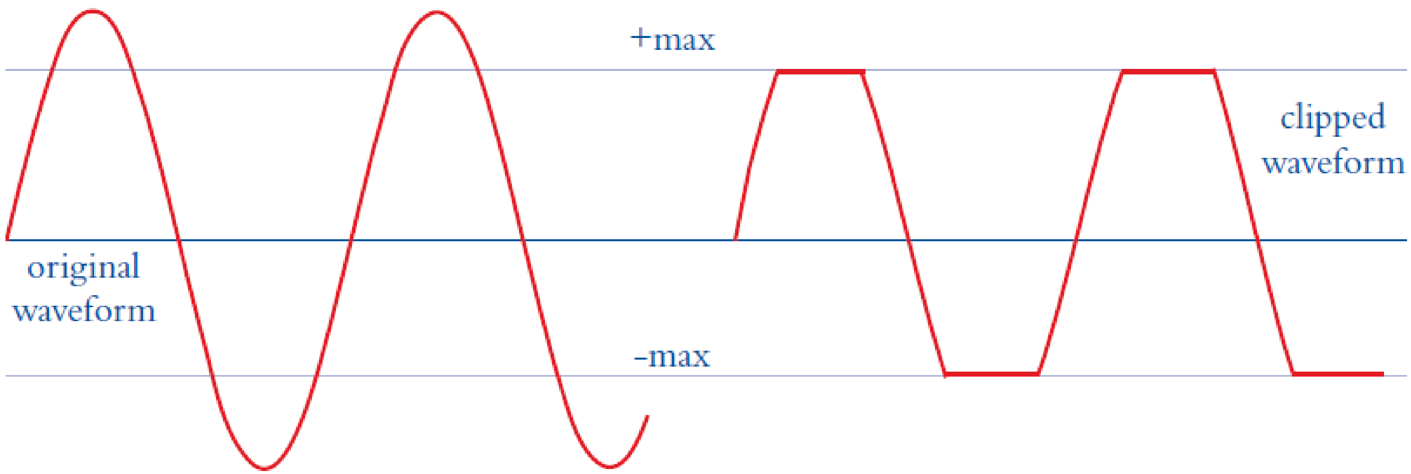
Modification of the threshold of hearing by a masking tone

Compression

- Filters are used to split a signal into 32 bands, and a masking level for each band is computed. Signals that fall below the level can be discarded.
- Practical implementations of perceptually based compression are the basis of *MP3* and *AAC* compression.

Sound Processing

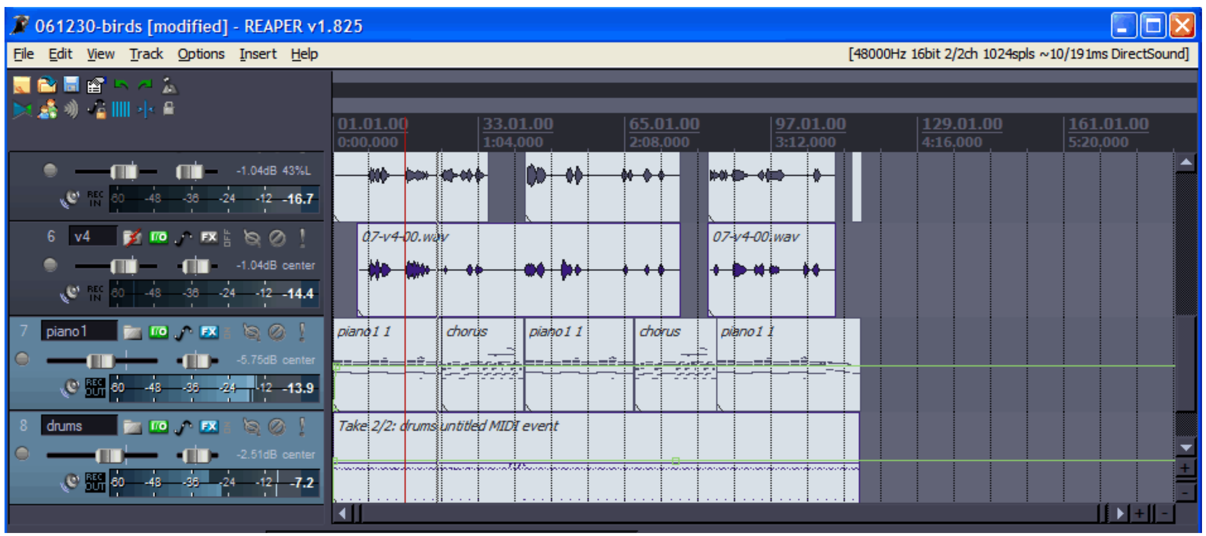
- If the recording level is too high, exceeding the max value that can be recorded, clipping will occur, causing distortion.



Clipping

Sound Processing

- Sound editing programs use a timeline interface, with multiple tracks (usually displayed as waveforms), which are mixed down to produce a stereo or mono output.



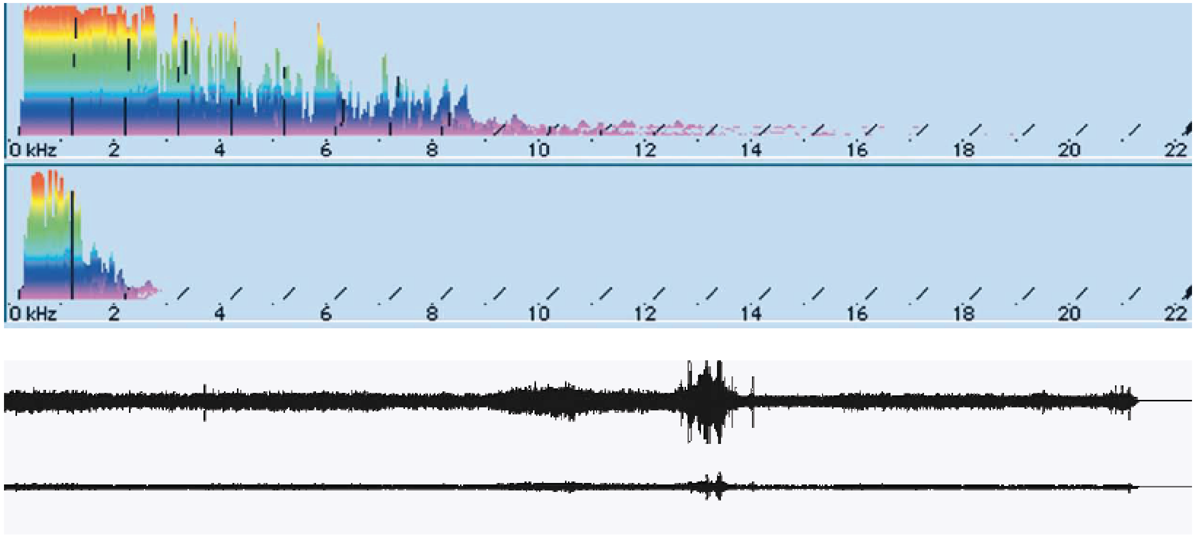
Sound Processing

- Short loops may be used to create voices for samplers; longer loops may be combined (e.g. in GarageBand) to build songs from repeating sections.

- Filters and gates are used to correct defects (e.g. remove noise) or to enhance or modify sounds (e.g. reverb).

Sound Processing

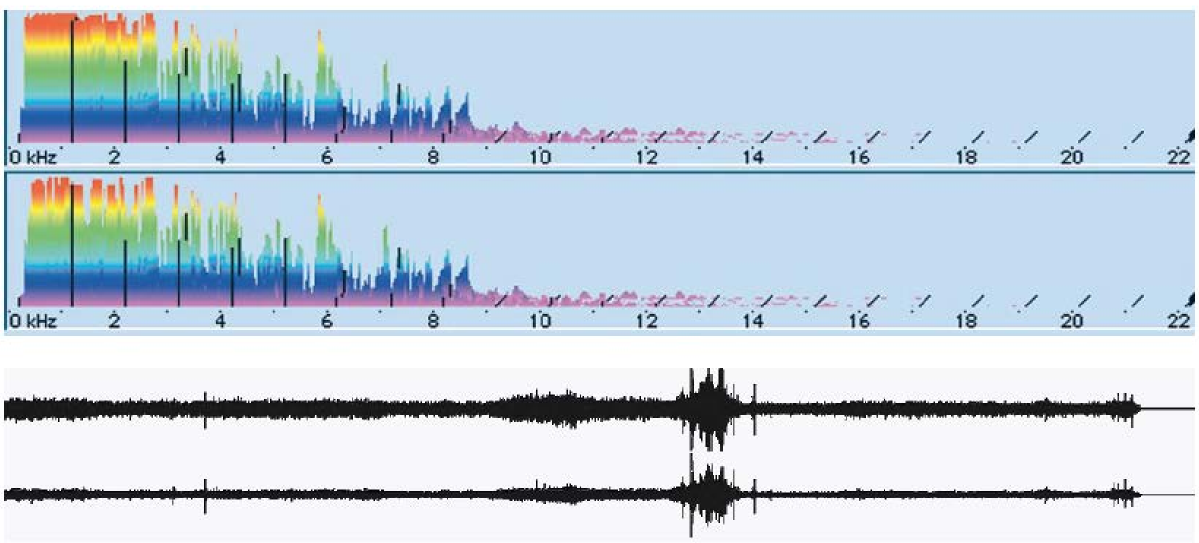
- Low pass filtering:
  - Allow low frequencies to pass through, remove high frequencies
  - Can be used to take out hiss



Low pass filtering

Processing Sound

- High pass filtering:
  - Allow high frequencies to pass through, remove low frequencies.
  - Can be used to take out low frequency '<u>rumble</u>' noise caused by mechanical vibrations.



High pass filtering

Processing Sound

- Notch filter
  - Remove a single narrow frequency band.
  - Example: remove hum picked up from the mains, typically a frequency 50 or 60Hz.

- Some sophisticated software supports:
  - Redraw a waveform.
  - Rub out the spikes corresponding to clicks, etc.
  - Not always easy, requiring considerable experience.

Processing Sound

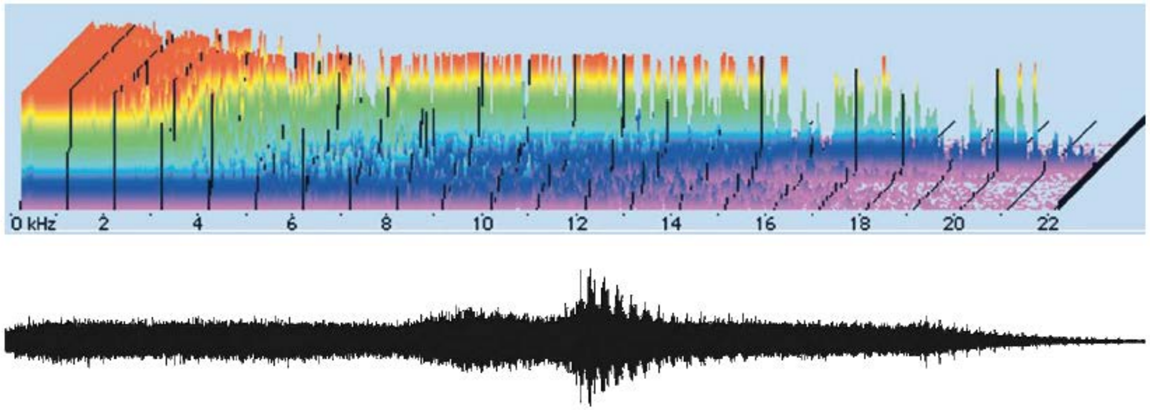
- Specialized filters:
  - De-esser
    - Remove the sibilance that results from speaking or singing into a microphone placed too close to the performer.
  - Click repairers
    - Remove clicks from recordings taken from damaged or dirty vinyl records.

Processing Sound

- All the filters are not infallible.
- The only sure way to get perfect sound is to start with a perfect take.
    - Microphones should be positioned to avoid sibilance
    - Kept away from fans and disk drives
    - Cables should be screened to avoid picking up hum, and so on.

Processing Sound

- Special effects:
  - Reverb effect:
    - Produced digitally by adding copies of a signal, delayed in time and attenuated, to the original.
    - These copies model reflections from surrounding surfaces.
    - The delay corresponding to the size of the enclosing space and the degree of attenuation modeling surfaces with different acoustic reflectivity.



Echo reverb

Processing Sound

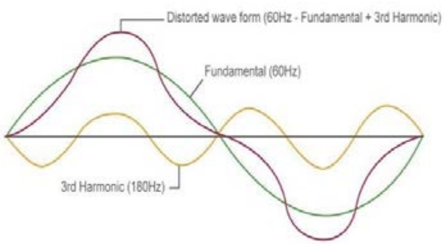
- Faders:
  - A sound’s volume gradually decreases or increases.
- Tremolo:
  - The amplitude to oscillate periodically from zero to its maximum
- Time stretching (slowing down and speeding up) and pitch alteration are more easily applied to digital audio than they were to analogue audio.
  - They are used for synchronization and for matching (e.g. when combining separately recorded loops).

Processing Sound

- Sound can be combined with pictures in a video editing program: sound tracks are displayed on the same timeline as video tracks, where they can be synchronized.
- Timecode is just a fiction when working with sound, owing to the high sampling rate, but it is valuable for synchronization.
- If sound and video are physically independent in a movie, synchronization may be lost, especially when it is sent over a network.

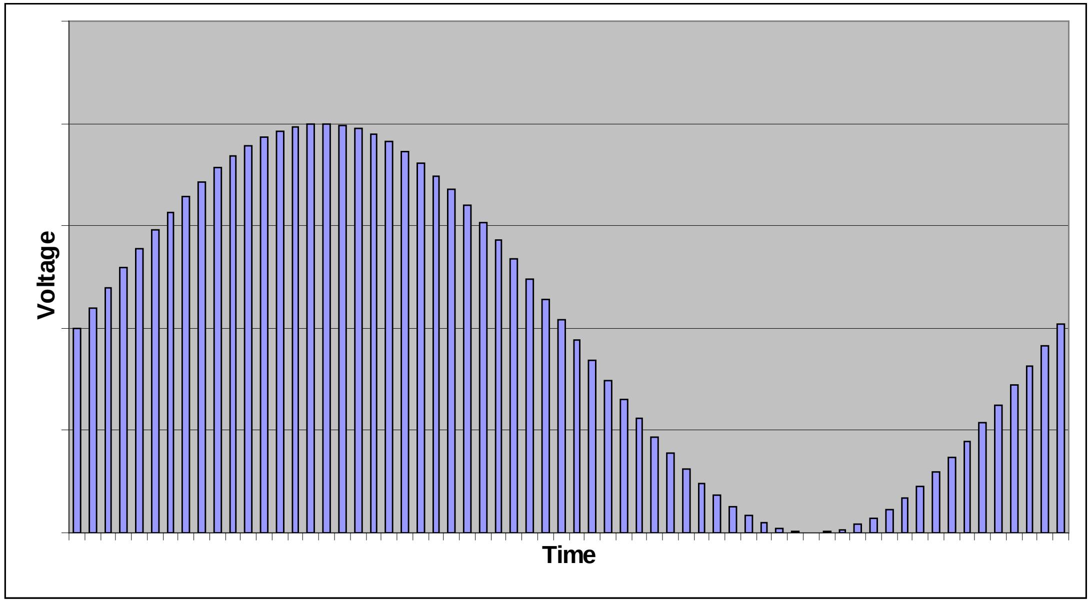
Revision

- When voice or music is captured by a microphone, it generates electrical signal.
- The signal consists of fundamental sine wave with certain **frequency** and **amplitude**.
- The fundamental sine wave is accompanied by harmonics.
- Adding the fundamental to harmonics, forms a composite sinusoidal signal that represents the original sound.
- Analog sinusoidal waveforms are converted to digital format by the **sampling** process.



Revision

- Sampling Process
  - The analog signal is sampled over time at regular intervals to obtain amplitudes of the signal at sampling time.
  - The interval at which sampling occurs is associated with **sampling rate**.



$$\text{Sampling rate} = \frac{1}{\text{sample time}} = \frac{1}{T}$$

T: Time interval between two samples

Revision

- Sampling Process (cont.)
  - The sample amplitude obtained at sampling instants is represented by an 8-bit value (one byte) or 16-bit (two bytes) value.
  - Higher values can also be used for higher resolution systems (high fidelity sound).
  - A composite signal of 11.025 kHz sampled 4 times every cycle will yield 44.1 kHz sampling rate. If you sample at higher rate, you need to store more samples.
  - For CD quality music at 44.1 kHz rate at 16-bit resolution, a one minute recording will require 44.1 x 1000 x 16 x 60 / 8 = 5.292 Mbytes.

Revision

- Audio objects generate a large volume of data. This poses two problems.
  - it requires a large storage
  - it takes a long time to transmit the data
- To solve these problems, the data is compressed.
  - Compression helps to shrink the storage and reduce network time.

- Audio industry uses 5.0125kHz, 22.05kHz and 44.1kHz as standard sampling frequencies. These frequencies are supported by most sound cards.

Sound Conversions

- Why considering sound conversions
  - **Tradeoff** between quality and storage

- Audio File Format
  - Uncompressed File Format
  - Format with Lossless Compression
  - Format with Lossy Compression
  - Other formats
  - Recommendations for WWW
- Codecs
- Audio Conversion Softwares

http://en.wikipedia.org/wiki/Audio_file_format

Sound Conversions

- Thinkings in mind…
  - Why are there so many audio formats?
  - What is the mathematical foundation of different audio files?
  - What is the benefit/shortcoming of each file format?
  - Which technique/format should I deploy in my application?

  - How to encode/decode each format?
  - Any non-technical restrictions in audio process

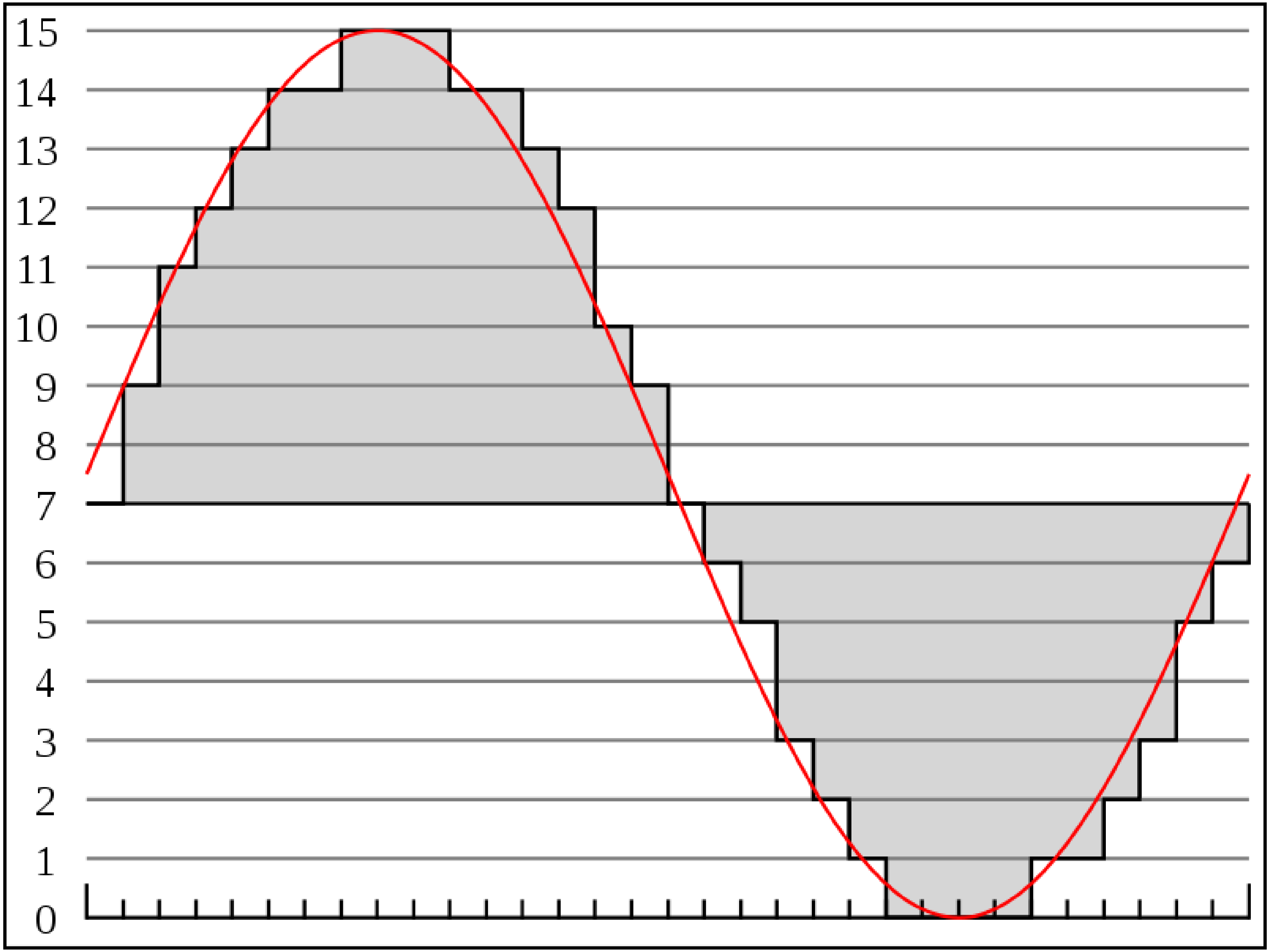
Uncompressed File Formats

- WAV
- AIFF
- AU
- Raw Audio Format, e.g., PCM
- ……

Pulse-Code Modulation

- PCM is a digital representation of an analog signal
  - the magnitude of the signal is sampled **regularly** at uniform intervals
  - then quantized to a series of symbols in a numeric (usually binary) code.

  - It is used in digital telephone systems.
  - The standard form for digital audio in computers and the compact disc format.
  - It is standard in digital video.
  - Uncompressed PCM is **NOT** typically used for video in standard definition consumer applications such as DVD because the bit rate required is far too high.

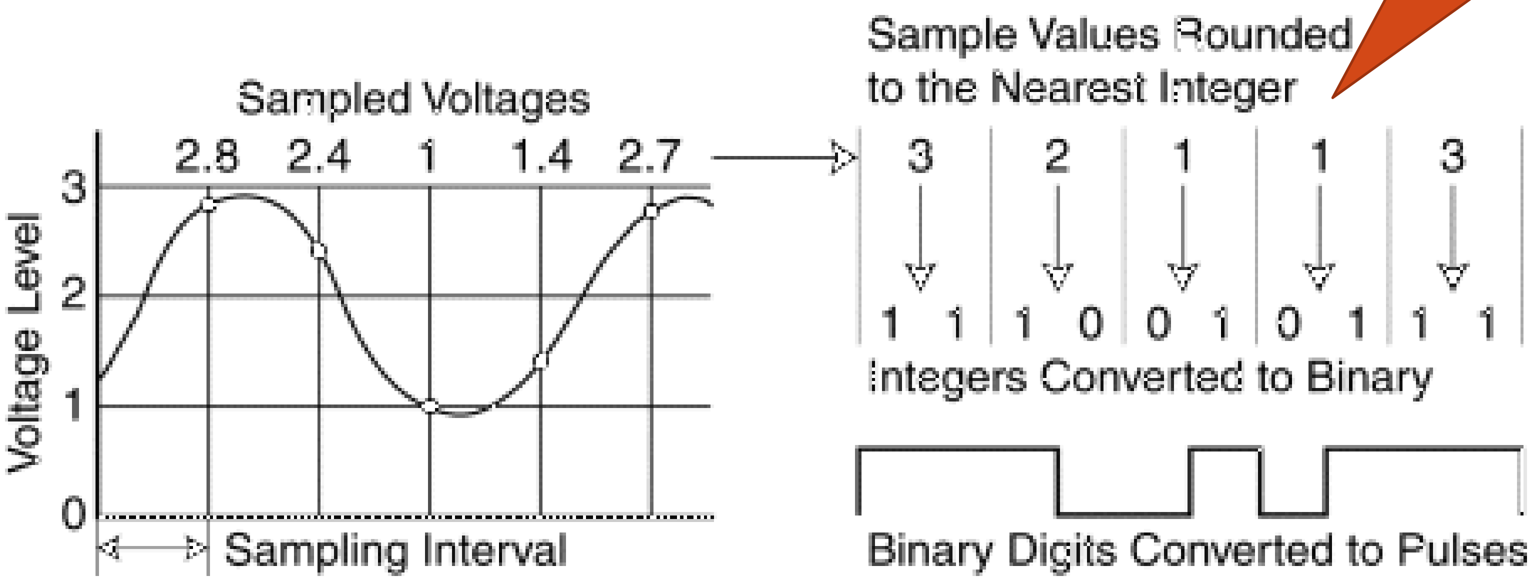


Sampling and quantization of a signal (red) for 4-bit PCM

Recap: Quantization (Pulse Code Modulation)

- At every time interval the sound is converted to a digital equivalent.
- Using 2 bits the following sound can be digitized

Quantization brings in Noise!

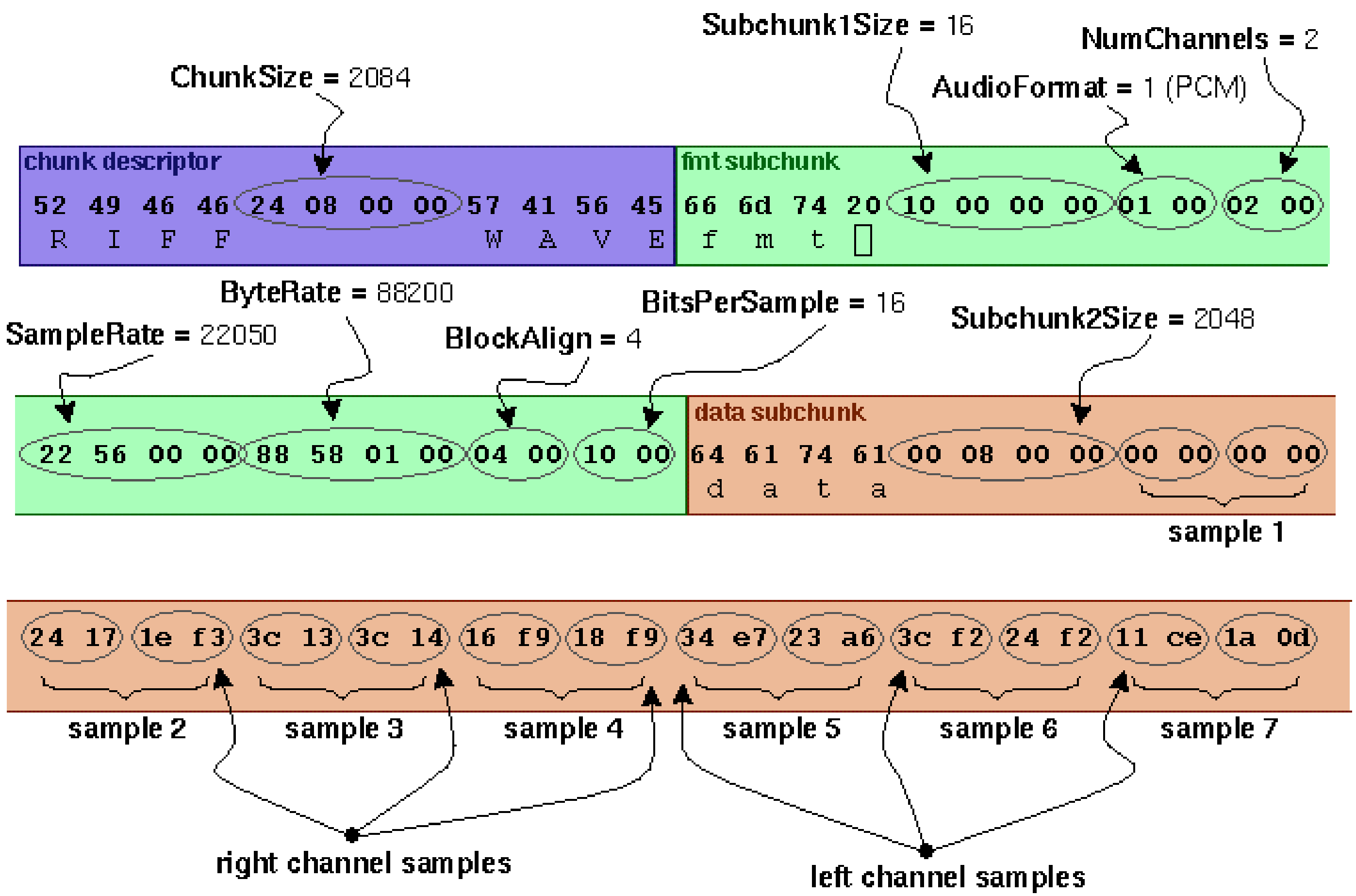


PCM

- PCM is usually stored as a .wav on Windows or as .aiff on Mac OS.
  - WAV and AIFF are flexible file formats designed to store more or less any combination of sampling rates or bitrates.
  - Suitable for storing and archiving an **original recording**.
  - AIFF is based on IFF (Interchange File Format by *Electric Arts*) format.
  - WAV is based on RIFF (Resource Interchange File Format) format, which is similar to IFF.

http://en.wikipedia.org/wiki/Interchange_File_Format
http://en.wikipedia.org/wiki/RIFF_%28File_format%29

Audio File Format: .WAV

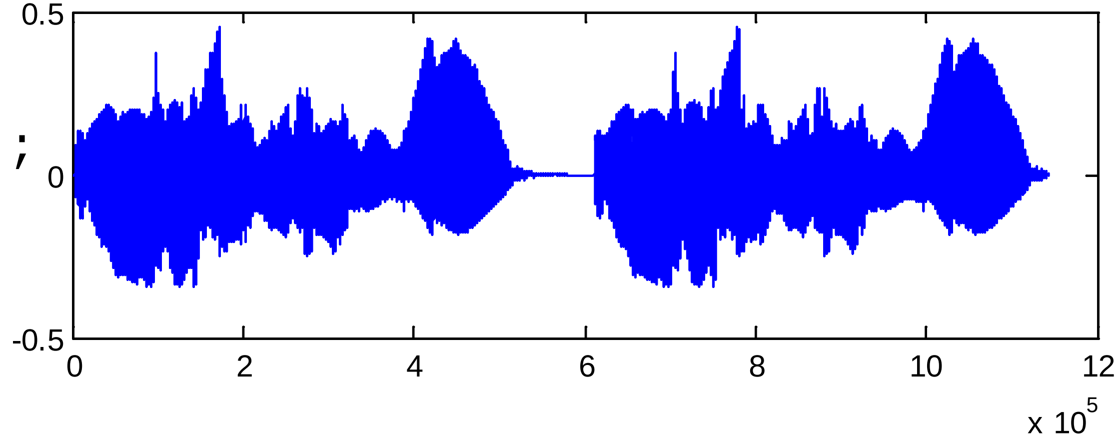


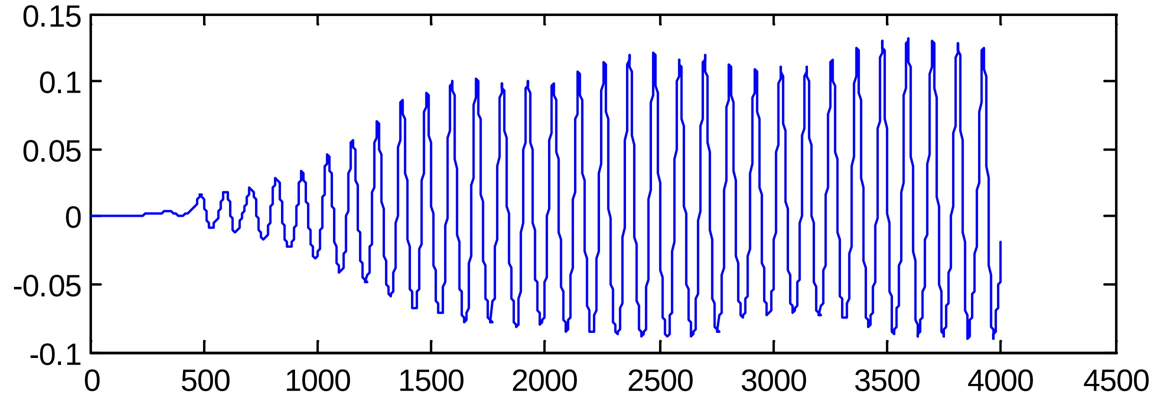
http://ccrma.stanford.edu/courses/422/projects/WaveFormat/

Example

Create this figure in Matlab:

```
x = wavread('horn.wav');
plot(x(:, 1));
plot(x(4000:10000, 1));
```





Note:
Wavread() normalizes the Samples to the range of [-1, 1].

Lossless Compressed Format

- Lossless compressed formats require more processing time than uncompressed formats but are more efficient in storage.
  - Uncompressed audio formats encode both sound and silence with the same number of bits per unit of time.
  - In a lossless compressed format, the music would occupy a smaller file and the silence take up almost no space at all.

- Lossless compression has a compression ratio of about 2:1
  - FLAC,
  - WavPack,
  - Monkey's Audio (APE),
  - ALAC/Apple Lossless
  - Shorten (SHN) …

Monkey’s Audio

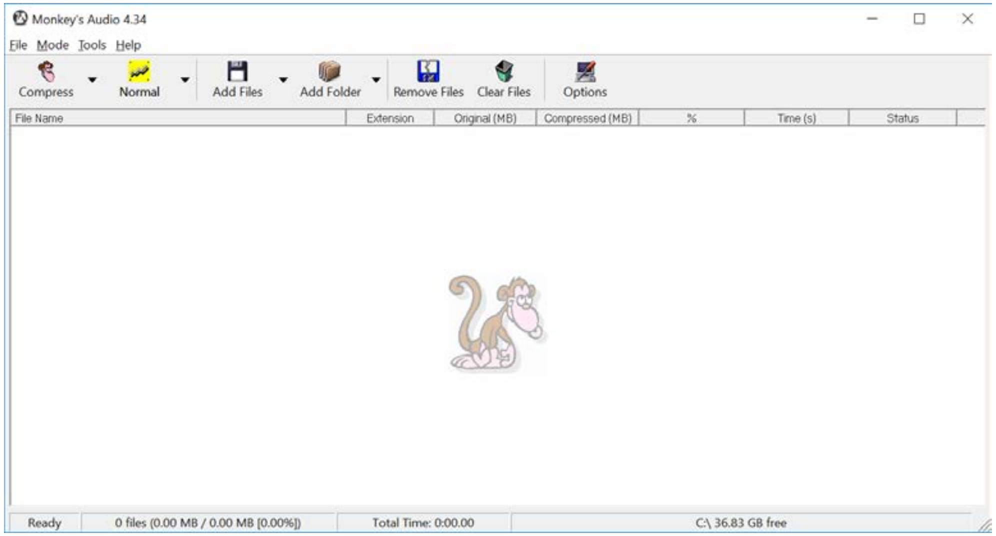
- Monkey's Audio is a lossless compression format for audio. It does not discard data during encoding.

- A digital recording (such as a CD) encoded to Monkey's Audio format can be decompressed into an identical copy of the original.

- Drawback: It is **proprietary** software, and has limited support on software platforms other than Windows. Alternatives such as FLAC may offer more options for some users.

- Filename extension: .ape for audio, and .apl for track metadata.

Monkey's Audio



Monkey’s Audio

- Storage:
  - Monkey's Audio compresses a little better than FLAC and a lot better than Shorten.
  - Encoding and decoding times are longer than FLAC and Shorten.
  - Like any lossless compression scheme, Monkey's Audio format takes up several times as much space as lossy compression formats like AAC, and MP3. A Monkey's Audio file is 3–5 times larger than a 192kbps bitrate MP3.
- Development:
  - The latest version of Monkey's Audio, ver. 4.34, was released on 2018-05-02.
  - The Shorten format is no longer in development.
  - FLAC has an active development community that continues to refine the format.
- Platform support:
  - Although Monkey's Audio is distributed as freeware, the license terms prevent most Linux distributions and other free software projects from including it.
  - Monkey's Audio is also supported on Linux and OS X using JRiver Media Center
  - FLAC has only open source licenses, so it comes pre-installed with most Linux distributions, is preferred by Linux users, and enjoys broad support in applications.

Lossy Compressed Formats

- MP3
- Vorbis
- Musepack
- AAC
- ATRAC
- RA
- lossy Windows Media Audio (WMA)
- …

Lossy Compressed Formats

- MP3
  - Designed by Moving Picture Experts Group as part of MPEG-1 standard.
  - The most popular format for downloading and storing music.

  - Eliminating portions of the audio file that are essentially inaudible.
  - A lossy compression algorithm to greatly reduce the amount of data required to represent the audio recording and still sound like a faithful reproduction of the original uncompressed audio for most listeners.
  - MP3 also has higher or lower bit rates, with higher or lower resulting quality.
  - Available sampling frequencies: 32, 44.1 and 48 kHz

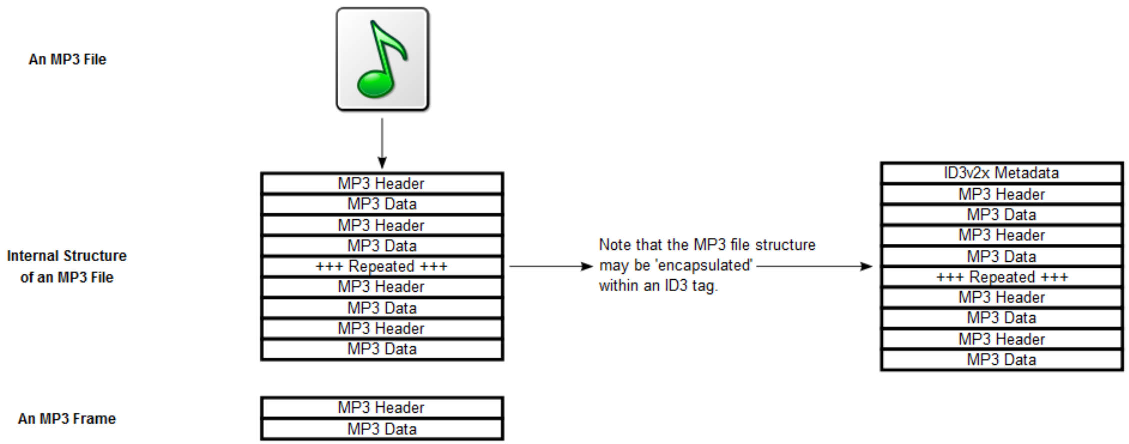
Lossy Compressed Formats

- Bitrates specified in MPEG-1 Audio Layer III standard:
  - 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 and 320 kbps
  - The bitrates 128, 160 and 192 kbps represent compression ratios of approximately 11:1, 9:1 and 7:1 respectively, comparing with CD.

- Constant Bit Rate Encoding (CBR)
  - one bit rate for the entire file
  - makes encoding simpler and faster

- Variable Bit Rate Encoding (VBR)
  - bit rate changes throughout the file
  - using a lower bit rate for the less complex passages and a higher one for the more complex parts

MP3 File Structure



Example MP3 Header

F F F B A 0 4 0 Colour-coding shows binary bit mapping to hex values below

Detail of an MP3 Header

Bits	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Binary	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Hex	F				F				F				B				A				0				4				0			
Meaning	MP3 Sync Word												Version	Layer		Error Protection	Bit Rate				Frequency		Pad. Bit	Priv. Bit	Mode		Mode Extension (Used With Joint Stereo)		Copy	Original	Emphasis	
Value	Sync Word												1 = MPEG	01 = Layer 3		1 = No	1010 = 160				00 = 44100 Hz		0 = Frame is not padded	Unknown	01 = Joint Stereo		0 = Intensity Stereo Off	0 = MS Stereo Off	0 = Not Copy-righted	0 = Copy Of Original Media	00 = None	

Lossy Compressed Formats

- WMA
  - Windows Media Audio format owned by Microsoft.
  - Designed with Digital Rights Management (DRM) abilities for copyright protection.
  - Parts of Windows media framework.
    - Windows media audio
    - Windows media audio professional
    - Windows media audio lossless
    - Windows media audio voice

Lossy Compressed Formats

- ATRAC (Adaptive Transform Acoustic Coding)
  - Developed by Sony.
  - It always has a .aa3 or a .oma file extension.
  - To open these files simply install the ATRAC3 drivers.
  - Used in Walkman and MD

Lossy Compressed Formats

- RA
  - Real Audio format designed for streaming audio over the Internet.
  - It can be played while it is downloading.
  - It is possible to stream RealAudio using HTTP, which works best with pre-recorded files.
  - Alternative protocols may work better for live broadcasts.

Other formats

- **MIDI (Musical Instrument Digital Interface)**
  - A simple scripting language and hardware setup
  - MIDI codes "events" stand for the production of sounds. e.g., a MIDI event might include values for the pitch of a single note, its duration, and its volume.
  - MIDI is a standard adopted by the electronic music industry for controlling devices, such as synthesizers and sound cards, that produce music.
  - Supported by most sound cards

Recommendations for WWW

- Portability between platforms, sound boards and software is determining factor
- Recommendation: audio/basic, 8-bit ISDN
- 8000 Hz sample rate, mono
- Also include MPEG audio version of audio clip for stereo support and CD-quality sound

Comparisons

- A detailed comparison of file formats is available from wiki:
  - http://en.wikipedia.org/wiki/Comparison_of_audio_formats

Question

- WAV → MP3, MP3 → WAV
  - Still the same quality?

File Format and Codec

- Difference between a file format and a codec.
  - A **codec** performs the encoding and decoding of the raw audio data;
  - The data itself is stored in a file with a specific audio file **format**.

Codec

- Codec: a device or computer program capable of encoding and/or decoding a digital data stream or signal.
  - The word codec is a blending of 'compressor-decompressor' or, more commonly, 'coder-decoder'.
- Lossless Codecs
  - archiving data in a compressed form while retaining all of the information present in the original stream
- Lossy Codecs
  - reduce quality by some amount in order to achieve compression

Audio Conversion Software

- Switch Audio Converter
  - http://www.nch.com.au/switch/
- Audio MP3 WAV WMA OGG Converter
  - http://www.audio-converter.com/
- Super Audio Converter
  - http://www.audioconverter.net/
- ……

Coding Algorithm

- Introduction
- Basic Concepts of Information Theory
- Run-Length Coding
- Variable-Length Coding (VLC)
- Dictionary-based Coding
- Arithmetic Coding
- Lossless Image Compression
- Further Exploration

Introduction

- **Compression**: the process of coding that will effectively reduce the total number of bits needed to represent certain information.

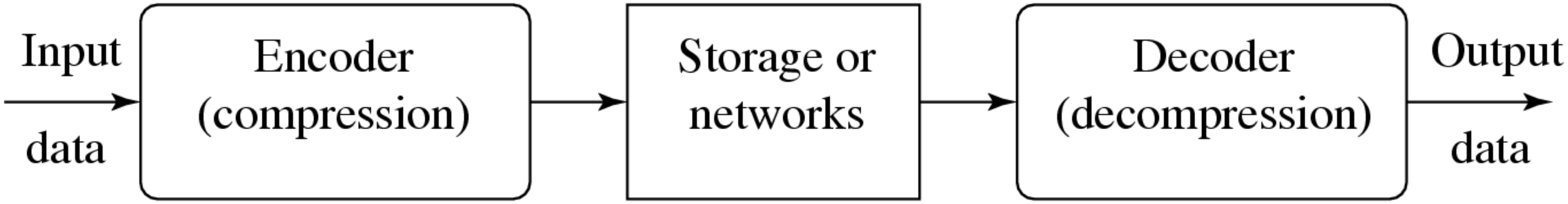


Fig. 7.1: A General Data Compression Scheme.

Introduction (cont'd)

- If the compression and decompression processes induce no information loss, then the compression scheme is **lossless**; otherwise, it is **lossy**.

Compression ratio:

$$compression\,ratio = \frac{B_0}{B_1} \qquad (7.1)$$

- B_0 – number of bits before compression
- B_1 – number of bits after compression

Basic Concepts of Information Theory

- The *entropy* η of an information *source* with alphabet $S = \{s_1, s_2, \ldots, s_n\}$ is:

$$\eta = H(S) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} \qquad (7.2)$$

$$= -\sum_{i=1}^{n} p_i \log_2 p_i \qquad (7.3)$$

- p_i – probability that symbol s_i will occur in S.
- $\log_2 \frac{1}{p_i}$ – indicates the amount of information (self-information as defined by Shannon) contained in s_i, which corresponds to the number of bits needed to encode s_i.

- In science, entropy is a measure of the disorder of a system.
- In information theory, entropy is a measure of **uncertainty**.
- The more entropy, the more uncertainty, the more information it carries.

- Example:
- Three types of languages are used in an essay and each letter/Hiragana/character appears averagely.
  - English-26 letters $$\eta = \sum_{i=1}^{26} \frac{1}{26} \log_2 26 = \boxed{4.7}$$
  - Japanese-50 Hiragana $$\eta = \sum_{i=1}^{50} \frac{1}{50} \log_2 50 = \boxed{5.64}$$
  - Chinese-2500 characters (frequently used)

$$\eta = \sum_{i=1}^{2500} \frac{1}{2500} \log_2 2500 = \boxed{11.3}$$

The information carried in a Chinese character is the biggest. Translation of an English essay into Chinese is shorter.

Distribution of Gray-Level Intensities

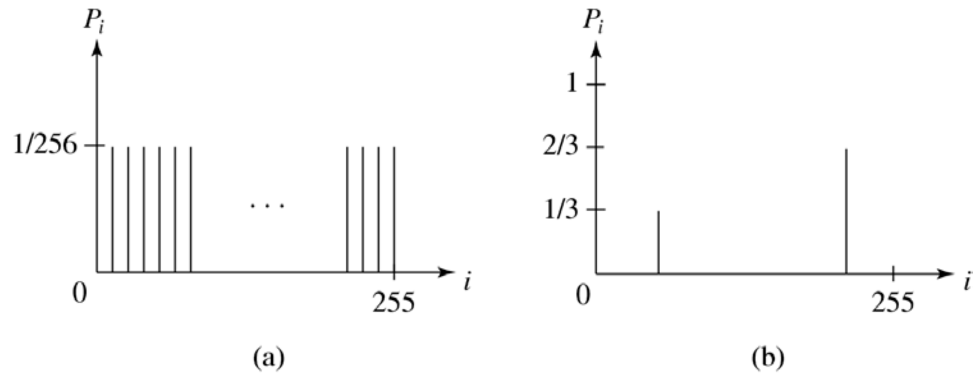


Fig. 7.2 Histograms for two gray-level images.

Fig. 7.2(a) shows the histogram of an image with *uniform* distribution of gray-level intensities. Its entropy is $\eta = 8$.

Fig. 7.2(b) shows the histogram of an image with two possible values. Its entropy is:

$$\eta = \frac{1}{3}\log_2 3 + \frac{2}{3}\log_2 \frac{3}{2} = 0.92 \qquad (7.4)$$

Exercises

1. For a string ‘xxzyyzzz’, calculate its entropy.
2. If a .BMP grayscale image with 800*400 pixels is converted to a .JPG image, the compression ratio is 1.2, calculate the file size of the .JPG image.

Exercises

1. For a string ‘xxzyyzzz’, calculate its entropy.

Probability: $P_x=1/4$, $P_z=1/2$, $P_y=1/4$.

$$\eta = -\frac{1}{4}\log_2\frac{1}{4} - \frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} = \frac{3}{2}\log_2 2 = \frac{3}{2}$$

Exercises

2. If a .BMP grayscale image with 800*400 pixels is converted to a .JPG image, the compression ratio is 1.2, calculate the file size of the .JPG image.

- Original .BMP image size=800*400*1 byte=320kB
- JPB image size=320kB/1.2=266.67kB

Entropy and Code Length

- As can be seen in Eq. (7.3): the entropy η is a weighted-sum of terms $\log_2 \frac{1}{p_i}$; hence it represents the *average* amount of information contained per symbol in the source S.
- The entropy η specifies the lower bound for the average number of bits to code each symbol in S, i.e.,

$$\eta \leq \bar{l} \tag{7.5}$$

$\bar{l}$ - the average length (measured in bits) of the codewords produced by the encoder.

Run-Length Coding

- **Memoryless Source**: an information source that is independently distributed. Namely, the value of the current symbol does not depend on the values of the previously appeared symbols.
- Instead of assuming memoryless source, *Run-Length Coding* (*RLC*) exploits memory present in the information source.
- **Rationale for RLC**: if the information source has the property that symbols tend to form continuous groups, then such symbol and the length of the group can be coded.

Run-Length Coding

- Example: If a scanline from a binary image is as below,

WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWW
WWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW

(67 characters)

- Using RLC, it can be rendered as below,

12W1B12W3B24W1B14W (18 characters)

- Compression ratio=3.72

Variable-Length Coding (VLC)

Shannon-Fano Algorithm

A **top-down** approach

1. Sort the symbols according to the **frequency count** of their occurrences.
2. Recursively divide the symbols into two parts, each with approximately the same number of counts, until all parts contain only one symbol.

A natural way of implementing the above procedure is to build a binary tree.

As a convention, assign bit 0 to its left branches and 1 to the right branches.

Shannon-Fano Algorithm

- Example: coding of "HELLO"

Symbol	H	E	L	O
Count	1	1	2	1

Frequency count of the symbols in "HELLO"

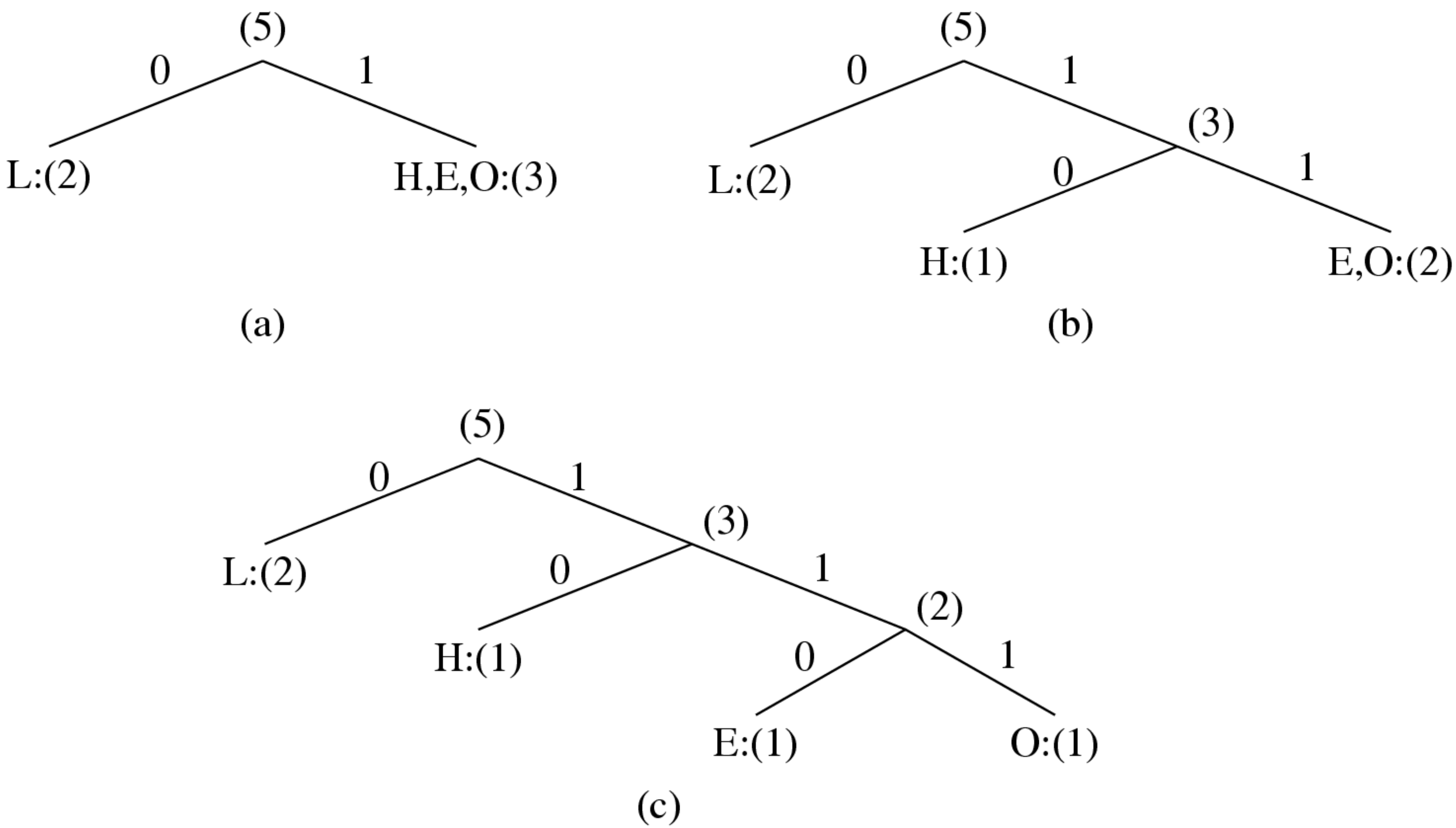


Fig. 7.3 Coding Tree for HELLO by Shannon-Fano.

Table 7.1: Result of Performing Shannon-Fano on HELLO

Symbol	Frequency	$\log_2(1/p_i)$	Code	# of bits used
L	2	1.32	0	2
H	1	2.32	10	2
E	1	2.32	110	3
O	1	2.32	111	3
			TOTAL # of bits:	10

$$\eta = 0.4 \times 1.32 + 0.2 \times 2.32 + 0.2 \times 2.32 + 0.2 \times 2.32 = 1.92$$

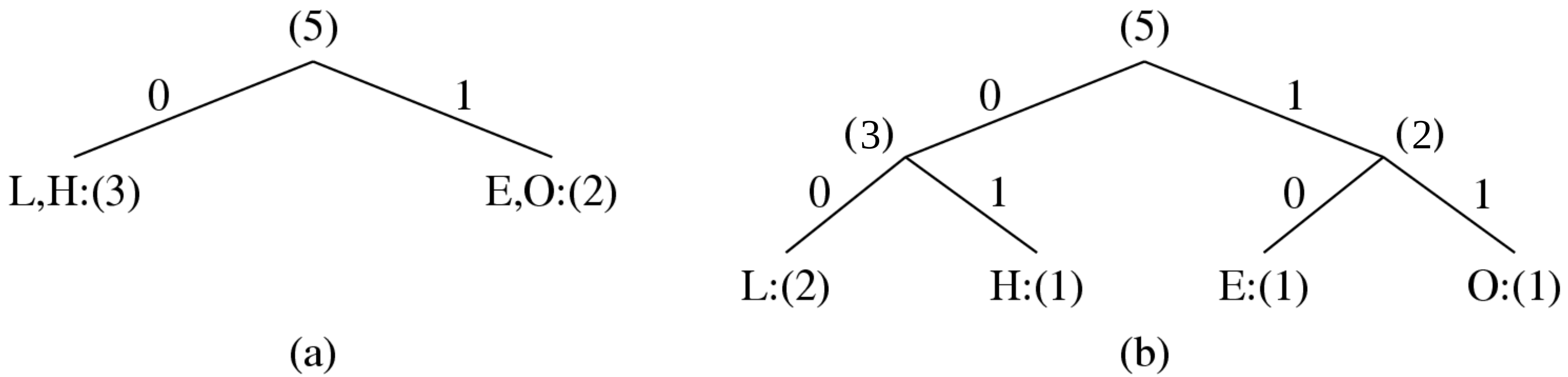


Fig. 7.4 Another coding tree for HELLO by Shannon-Fano.

Table 7.2: Another Result of Performing Shannon-Fano on HELLO (see Fig. 7.4)

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

Exercise

- Using Shannon-Fano Algorithm to encode the following symbols

Symbol	**a**	**b**	**c**	**d**	**e**
Counts	1	1	3	4	1

Huffman Coding

- A **bottom-up** approach
- **ALGORITHM 7.1-HUFFMAN CODING**

1. Initialization: put all symbols sorted according to their *frequency counts from lowest to highest*.
2. Repeat until the list has only one symbol left:
   a) From the list pick two symbols with the *lowest* frequency counts. Form a Huffman subtree that has these two symbols as child nodes and create a parent node.

   b) Assign the sum of the children's frequency counts to the parent and insert it into the list such that the order is maintained.

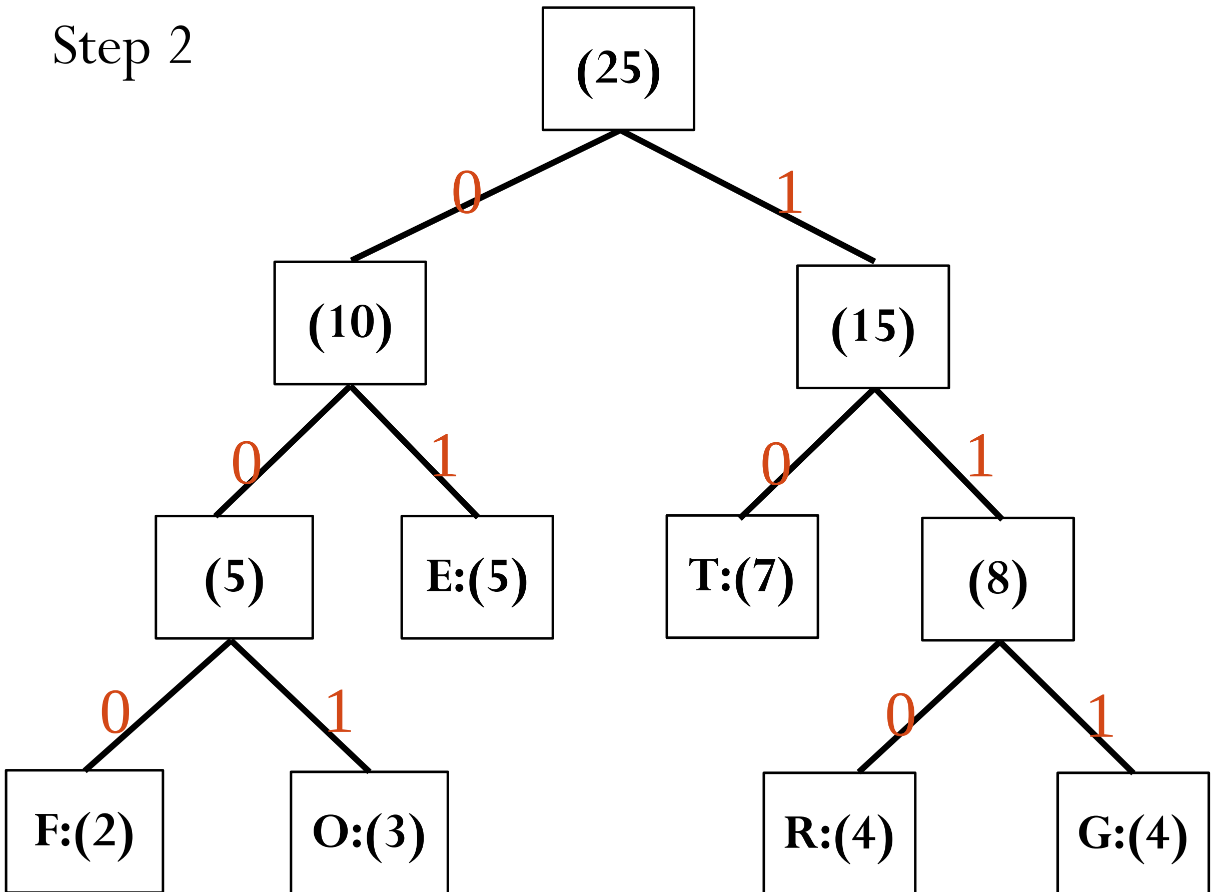
   c) Delete the children from the list.
3. Assign a codeword for each leaf based on the path **from the root**.

Example:

Step 1

Symbol	F	O	R	G	E	T
Frequency	**2**	**3**	**4**	**4**	**5**	**7**

Step 2



Frequency ranking list

Iter1: 2 3 4 4 5 7

Iter2: 4 4 5 5 7

Iter3: 5 5 7 8

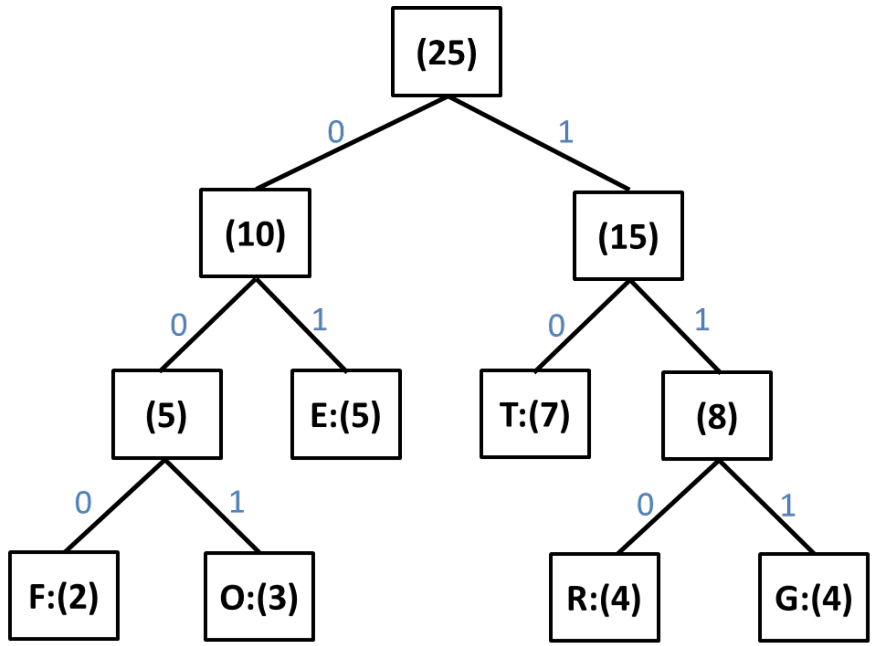
Iter4: 7 8 10

Iter5: 10 15

Iter6: 25

3 tips:

1. The frequency count on the same level of nodes should always be : small count on the left and big count on the right.
2. if the count are the same, doesn’t matter which is on the right or left.
3. Mark 0 on the left branch and mark 1 on the right branch.



Symbol	F	O	R	G	E	T
Frequency	2	3	4	4	5	7
Code	000	001	110	111	01	10
p_i	0.08	0.12	0.16	0.16	0.2	0.28
$\log_2(1/p_i)$	3.6439	3.0589	2.6439	2.6439	2.3219	1.8365

$$\eta = 2.4832$$

Properties of Huffman Coding

1. **Unique Prefix Property**: No Huffman code is a prefix of any other Huffman code - precludes any ambiguity in decoding.

2. **Optimality**: *minimum redundancy code* - proved optimal for a given data model (i.e., a given, accurate, probability distribution):

- The two least frequent symbols will have the same length for their Huffman codes, differing only at the last bit.
- Symbols that occur more frequently will have shorter Huffman codes than symbols that occur less frequently.
- The average code length for an information source S is strictly less than η + 1. Combined with Eq. (7.5), we have:

$$\eta < \bar{l} < \eta + 1 \tag{7.6}$$

Exercise

- Using Huffman coding Algorithm to encode the following symbols

Symbol	a	b	c	d	e
Counts	1	1	3	4	1

Extended Huffman Coding

- **Motivation**: All codewords in Huffman coding have integer bit lengths. It is wasteful when p_i is very large and hence $\log_2 \frac{1}{p_i}$ is close to 0. (Information carried in this symbol is little.)

- Why not group several symbols together and assign a single codeword to the group as a whole?

- **Extended Alphabet**: For alphabet $S = \{s_1, s_2, \ldots, s_n\}$, if k symbols are grouped together, then the *extended alphabet* is:

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

- — the size of the new alphabet $S^{(k)}$ is n^k.

Extended Huffman Coding (cont'd)

- It can be proven that the average # of bits for each symbol is:

$$\eta \le \bar{l} < \eta + \frac{1}{k} \qquad (7.7)$$

- An improvement over the original Huffman coding, but not much.

- **Problem**: If k is relatively large (e.g., $k \ge 3$), then for most practical applications where $n \gg 1$, n^k implies a huge symbol table — impractical.

Adaptive Huffman Coding

- **Adaptive Huffman Coding**: statistics are gathered and updated dynamically *as the data stream arrives*.

ENCODER

```
ENCODER
-------
Initial_code();

while not EOF
 {
       get(c);
       encode(c);
       update_tree(c);
 }
```

DECODER

```
DECODER
-------
Initial_code();

while not EOF
{
       decode(c);
       output(c);
       update_tree(c);
}
```

Adaptive Huffman Coding (Cont'd)

- `Initial_code` assigns symbols with some initially agreed upon codes, without any prior knowledge of the frequency counts.

- `update_tree` constructs an Adaptive Huffman tree. It basically does two things:

    a. increments the frequency counts for the symbols (including any new ones).
    b. updates the configuration of the tree.

- The *encoder* and *decoder* must use exactly the same `initial_code` and `update_tree` routines.

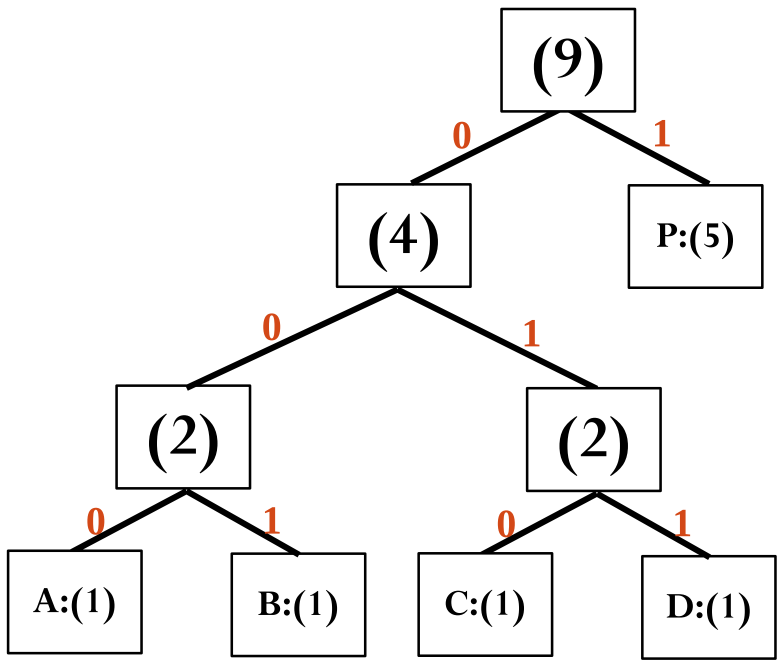
Notes on Adaptive Huffman Tree Updating

- Nodes are numbered in order from left to right, bottom to top. The numbers in parentheses indicates the count.
- The tree must always maintain its ***sibling* property**, i.e., all nodes (internal and leaf) are arranged in the order of increasing counts.
    - If the sibling property is about to be violated, a *swap* procedure is invoked to update the tree by rearranging the nodes.
- When a swap is necessary, the farthest node with count N is swapped with the node whose count has just been increased to $N+1$.

Adaptive Huffman Tree Updating

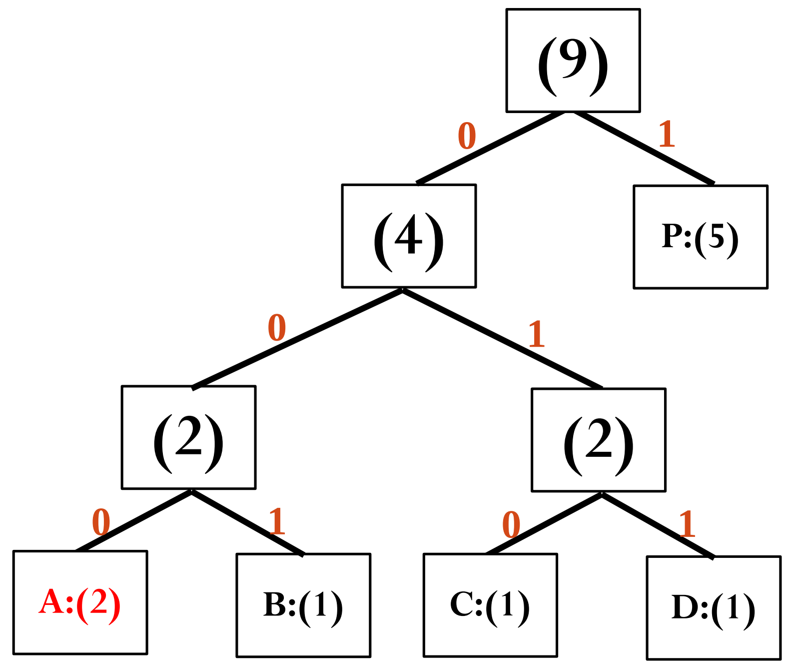
- Example: Symbol input: **PPDCPPBPA…**

Symbol	A	B	C	D	P
Frequency	1	1	1	1	5



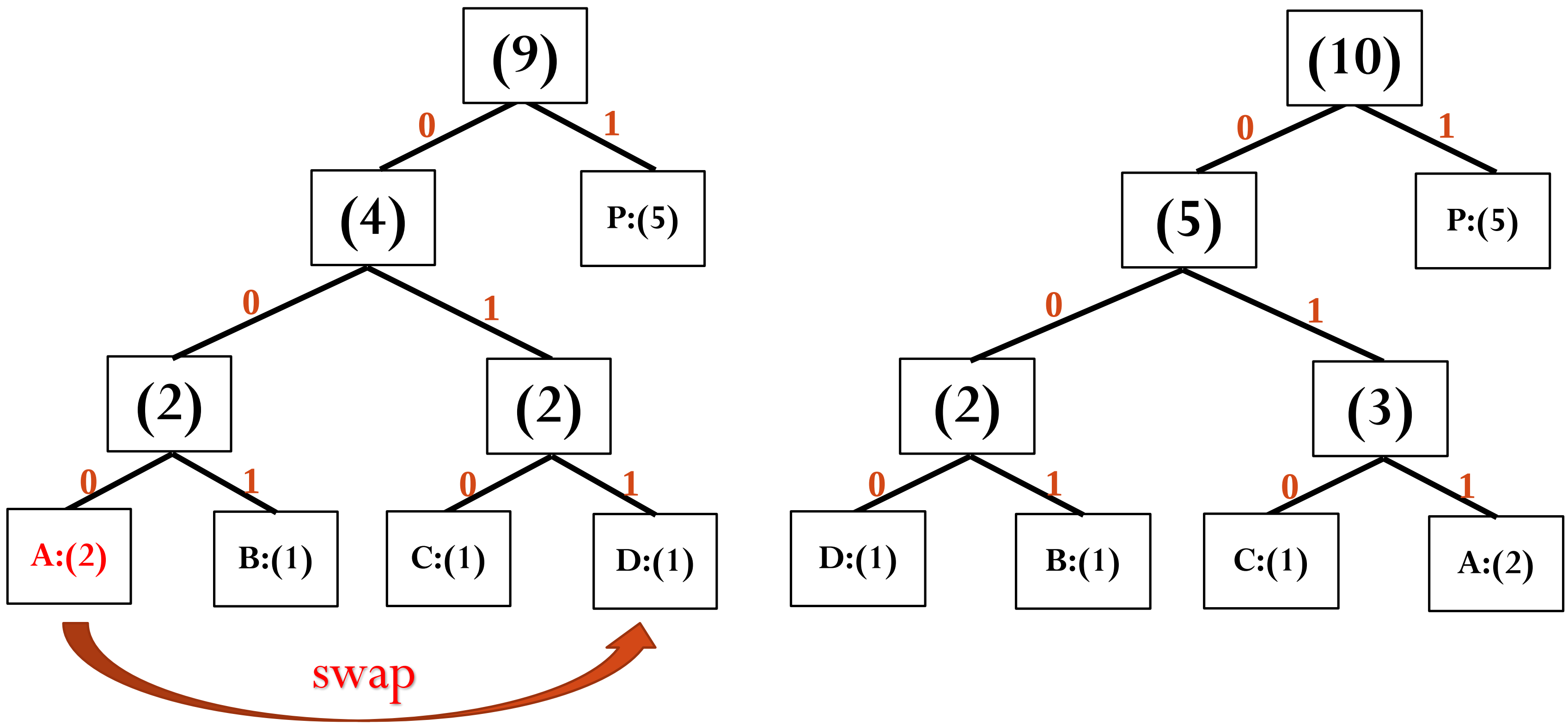
receiving 2^{nd} A

Symbol	D	B	C	A	P
Frequency	1	1	1	2	5



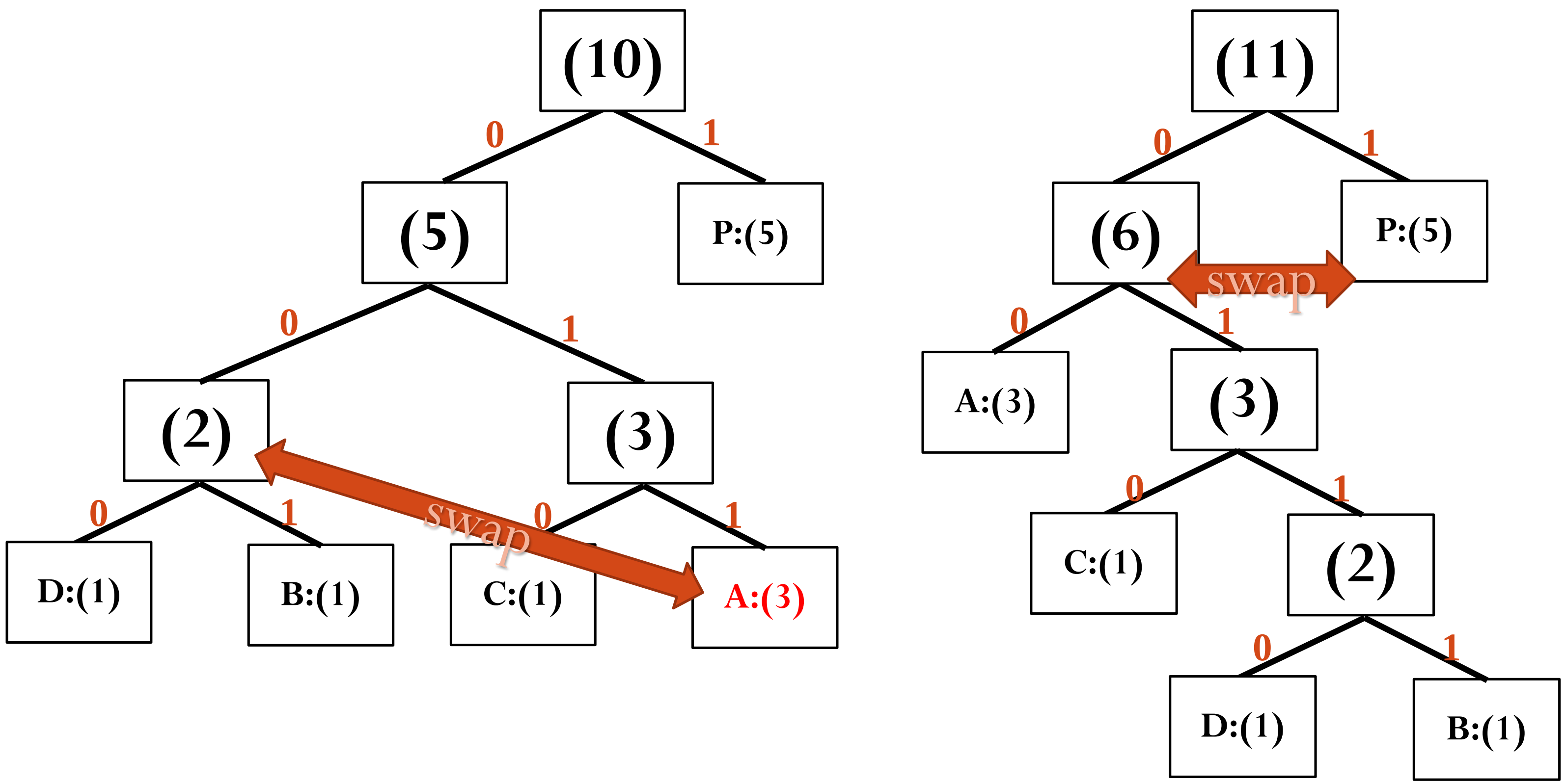
Receiving 2nd A

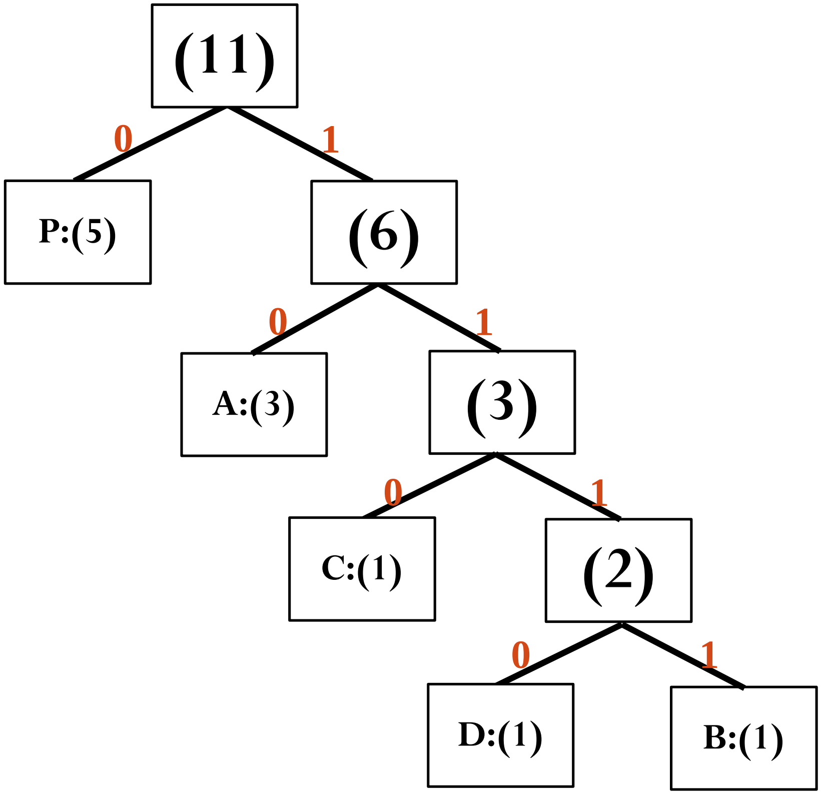
Symbol	D	B	C	A	P
Frequency	1	1	1	2	5



- Receiving 3rd A

Symbol	D	B	C	A	P
Frequency	1	1	1	3	5




(11)
0
1
P:(5)
(6)
0
1
A:(3)
(3)
0
1
C:(1)
(2)
0
1
D:(1)
B:(1)

Exercise

- Using Adaptive Huffman coding Algorithm to encode the following string:
- The initial input are ‘xxyzzz…’,work out the Huffman tree.
- When a second ‘y’ comes, how the Huffman tree changes.
- When a third ‘y’ comes, how the Huffman tree changes.
- When a fourth ‘y’ comes, how the Huffman tree changes.

Another Example: Adaptive Huffman Coding

- This is to clearly illustrate more implementation details. We show exactly what *bits* are sent, as opposed to simply stating how the tree is updated.

- An additional rule: if any character/symbol is to be sent the first time, it must be preceded by a special symbol, NEW. The initial code for NEW is 0. The *count* for NEW is always kept as 0 (the count is never increased); hence it is always denoted as NEW:(0).

Initial Code

NEW:	0
A:	00001
B:	00010
C:	00011
D:	00100
	. .
	. .
	. .

Table 7.3: Initial code assignment for AADCCDD using adaptive Huffman coding. (ASCII)

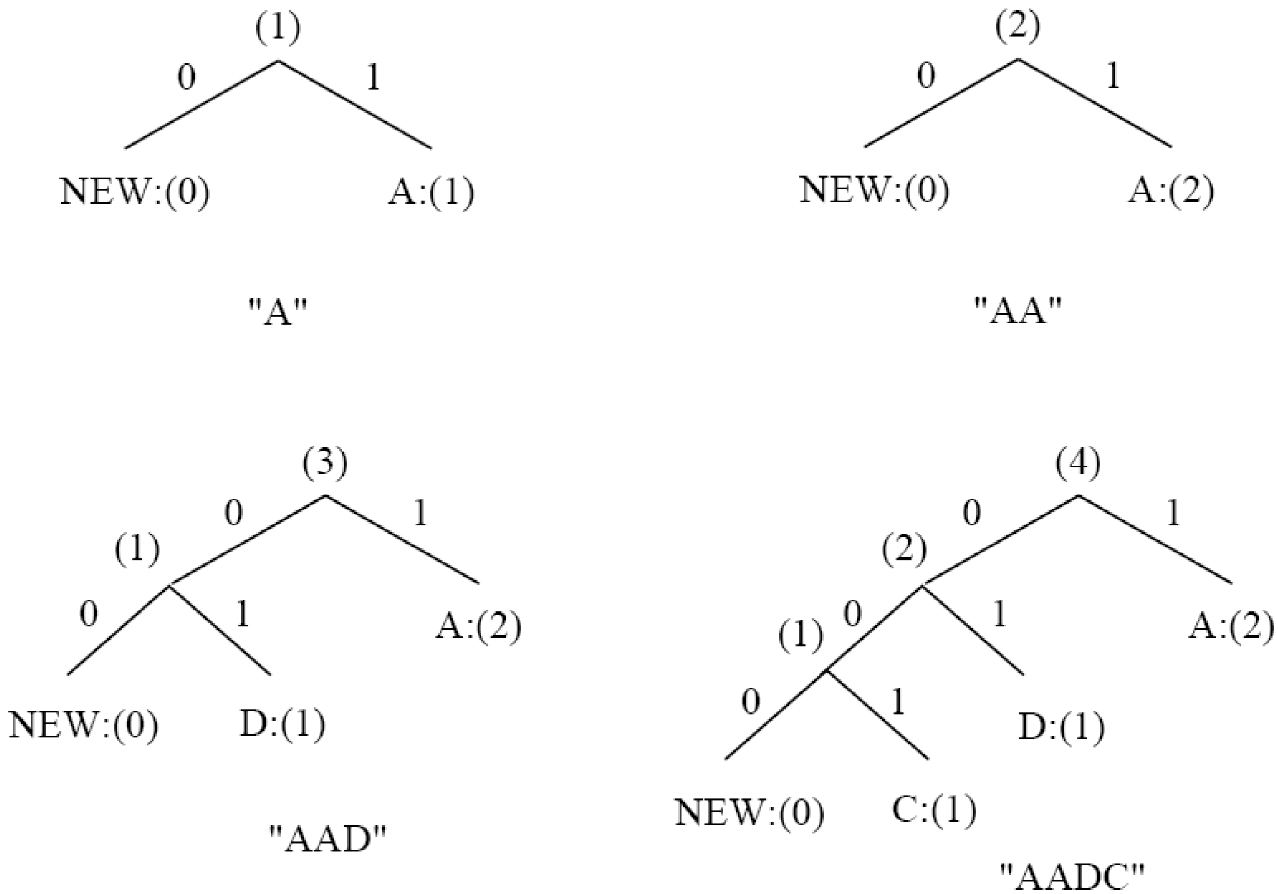


Fig. 7.7 Adaptive Huffman tree for AADCCDD.

Symbol	NEW	A	A	NEW	D	NEW	C	C	D	D
Code	0	00001	1	0	00100	00	00011	001	101	101

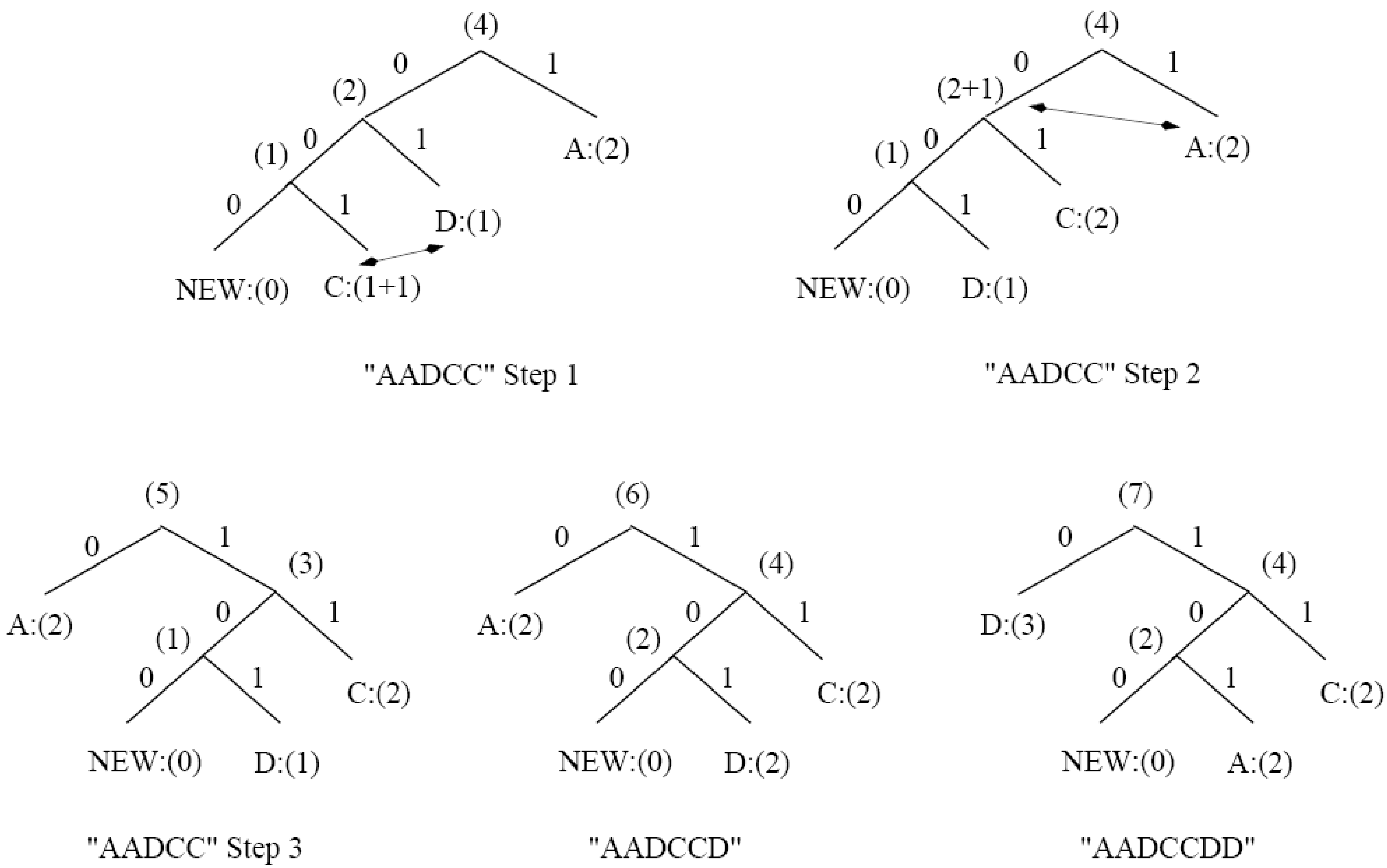


Fig. 7.7 (cont'd) Adaptive Huffman tree for AADCCDD.

Table 7.4 Sequence of symbols and codes sent to the decoder

Symbol	NEW	A	A	NEW	D	NEW	C	C	D	D
Code	0	00001	1	0	00100	00	00011	001	101	101

- It is important to emphasize that the code for a particular symbol changes during the adaptive Huffman coding process.

- For example, after AADCCDD, when the character D overtakes A as the most frequent symbol, its code changes from 101 to 0.

Exercise

- For ‘xxyzzz’ work out code by using adaptive Huffman coding with NEW symbol.

Initial Code

NEW:	0
x:	11000
y:	11001
z:	11010

Dictionary-based Coding

- LZW uses *fixed-length* codewords to represent variable-length strings of symbols/characters that commonly occur together, e.g., words in English text.

- The LZW encoder and decoder build up the *same* dictionary dynamically while receiving the data.

- LZW places longer and longer repeated entries into a dictionary, and then emits the *code* for an element, if the element has already been placed in the dictionary.

- **ALGORITHM 7.2 - LZW Compression**

```
BEGIN
       s = next input character;
       while not EOF
       {
               c = next input character;
               if s + c exists in the dictionary
               s = s + c;
               else
           {
               output the code for s;
               add string s + c to the dictionary with a new code;
               s = c;
           }
       }
       output the code for s;
END
```

- **Example:**

  LZW compression for string "ABABBABCABABBA"

- Let's start with a very simple dictionary (also referred to as a "string table"), initially containing only 3 characters, with codes as follows:

Code	String
1	A
2	B
3	C

- Now if the input string is "ABABBABCABABBA", the LZW compression algorithm works as follows:

Example:

LZW compression for string "ABABBABCABABBA"

s	c	s+c	Output	Code	String
				1 2 3	A B C
A	B	AB	1	4	AB
B	A	BA	2	5	BA
A	B	AB			
AB	B	ABB	4	6	ABB
B	A	BA			
BA	B	BAB	5	7	BAB
B	C	BC	2	8	BC
C	A	CA	3	9	CA
A	B	AB			
AB	A	ABA	4	10	ABA
A	B	AB			
AB	B	ABB			
ABB	A	ABBA	6	11	ABBA
A	EOF	A	1		

Example:

LZW compression for string "ABABBABCABABBA"

s	c	s+c	Output	Code	String
				1 2 3	A B C
A	B	AB	1	4	AB
B	A	BA	2	5	BA
A	B	AB			
AB	B	ABB	4	6	ABB
B	A	BA			
BA	B	BAB	5	7	BAB
B	C	BC	2	8	BC
C	A	CA	3	9	CA
A	B	AB			
AB	A	ABA	4	10	ABA
A	B	AB			
AB	B	ABB			
ABB	A	ABBA	6	11	ABBA
A	EOF	A	1		

- The output codes are: **1 2 4 5 2 3 4 6 1**. Instead of sending 14 characters, only 9 codes need to be sent (compression ratio = 14/9 = 1.56).

Input String "ABCBCABCBBCAB" (13 characters)

s	C	s+c	Output	Code	String
				1 2 3	A B C
A	B	AB	1	4	AB
B	C	BC	2	5	BC
C	B	CB	3	6	CB
B	C	BC			
BC	A	BCA	5	7	BCA
A	B	AB			
AB	C	ABC	4	8	ABC
C	B	CB			
CB	B	CBB	6	9	CBB
B	C	BC			
BC	A	BCA			
BCA	B	BCAB	7	10	BCAB
B	EOF		2		

Output String "1 2 3 5 4 6 7 2" (8 characters)

- **ALGORITHM 7.3 LZW Decompression (simple version)**

```
BEGIN
    s = NIL;
    while not EOF
    {
          k = next input code;
          entry = dictionary entry for k;
          output entry;
          if (s != NIL)
                add string s + entry[0] to
                dictionary with a new code;
          s = entry;
    }
END
```

Example 7.3:

Input codes to the decoder are 1 2 4 5 2 3 4 6 1. The initial string table is identical to what is used by the encoder.

The LZW decompression algorithm then works as follows:

S	K	Entry/output	Code	String
			1 2 3	A B C
NIL	**1**	A		
A	**2**	B	4	AB
B	**4**	AB	5	BA
AB	**5**	BA	6	ABB
BA	**2**	B	7	BAB
B	**3**	C	8	BC
C	**4**	AB	9	CA
AB	**6**	ABB	10	ABA
ABB	**1**	A	11	ABBA
A	**EOF**			

The LZW decompression algorithm then works as follows:

S	K	Entry/output	Code	String
			1	A
			2	B
			3	C
NIL	1	A		
A	2	B	4	AB
B	4	AB	5	BA
AB	5	BA	6	ABB
BA	2	B	7	BAB
B	3	C	8	BC
C	4	AB	9	CA
AB	6	ABB	10	ABA
ABB	1	A	11	ABBA
A	EOF			

- Apparently, the output string is “ABABBABCABABBA”, a truly lossless result!

ALGORITHM 7.4 LZW Decompression (modified)

```
BEGIN
   s = NIL;
   while not EOF
   {
         k = next input code;
         entry = dictionary entry for k;

         /* exception handler */
         if (entry == NULL)
                 entry = s + s[0];

         output entry;
         if (s != NIL)
                 add string s + entry[0] to dictionary with a new
   code;
         s = entry;
   }
END
```

LZW Coding (cont'd)

- In real applications, the code length l is kept in the range of $[l_0, l_{max}]$. The dictionary initially has a size of 2^{l_0}. When it is filled up, the code length will be increased by 1; this is allowed to repeat until $l = l_{max}$.

- When l_{max} is reached and the dictionary is filled up, it needs to be flushed (as in Unix *compress*, or to have the LRU (least recently used) entries removed).

Exercise

- Use the LZW method to compress the string "ABCBBAACABAC" with the following initial dictionary.
- Show the compression process step by step and give the code result.

Code	String
1	A
2	B
3	C

Arithmetic Coding

- Arithmetic coding is a more modern coding method that usually **out-performs** Huffman coding.
- Huffman coding assigns each symbol a codeword which has an integral bit length. Arithmetic coding can treat the whole message as one unit.
- A message is represented by a half-open interval $[a, b)$ where a and b are real numbers between 0 and 1. Initially, the interval is $[0, 1)$. When the message becomes longer, the length of the interval shortens and the number of bits needed to represent the interval increases.

ALGORITHM 7.5 Arithmetic Coding Encoder

```
BEGIN
  low = 0.0;    high = 1.0;     range = 1.0;

  while (symbol != terminator)
  {
      get (symbol);
      low = low + range * Range_low(symbol);
      high = low + range * Range_high(symbol);
      range = high - low;
  }

  output a code so that low <= code < high;
END
```

Example: Encoding in Arithmetic Coding

Symbol	Probability	Range
A	0.2	[0, 0.2)
B	0.1	[0.2, 0.3)
C	0.2	[0.3, 0.5)
D	0.05	[0.5, 0.55)
E	0.3	[0.55, 0.85)
F	0.05	[0.85, 0.9)
$	0.1	[0.9, 1.0)

Probability distribution of symbols.

Fig. 7.8: Arithmetic Coding: Encode Symbols "CAEE$"

Encode Symbols “CAEE$”

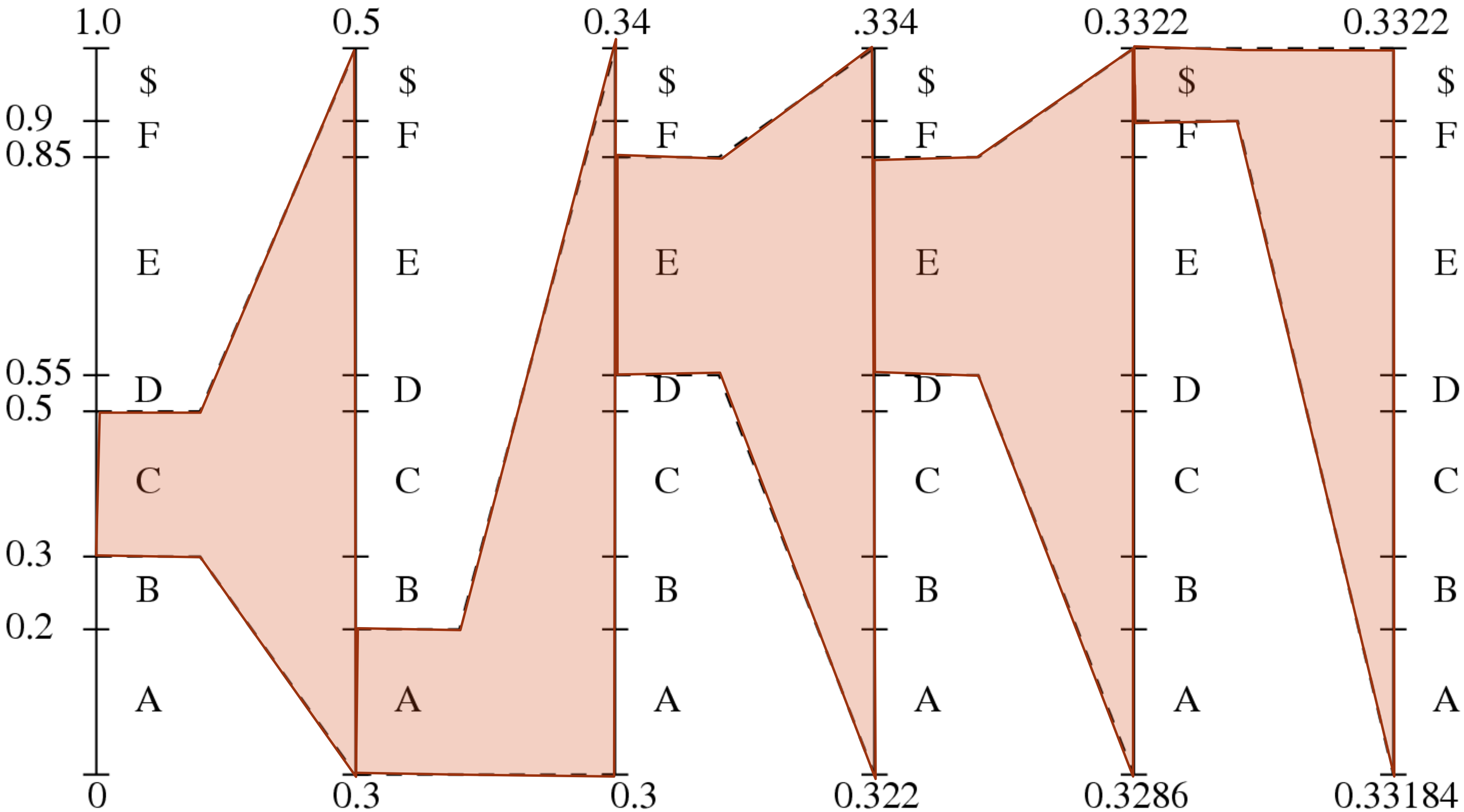


Fig. 7.8(b) Graphical display of shrinking ranges.

Example: Encoding in Arithmetic Coding

Symbol	Low	High	Range
	0	1.0	1.0
C	0.3	0.5	0.2
A	0.30	0.34	0.04
E	0.322	0.334	0.012
E	0.3286	0.3322	0.0036
$	0.33184	0.33220	0.00036

New *low*, *high*, and *range* generated.

Fig. 7.8 (cont'd): Arithmetic Coding: Encode Symbols "CAEE$"

$$Range = P_C \times P_A \times P_E \times P_E \times P_S = 0.2 \times 0.2 \times 0.3 \times 0.3 \times 0.1 = 0.00036$$

- **PROCEDURE 7.2 Generating Codeword for Encoder**

```
BEGIN
   code = 0;
   k = 1;
   while (value(code) < low)
   {
       assign 1 to the kth binary fraction bit
       if (value(code) > high)
       replace the kth bit by 0
       k = k + 1;
   }
END
```

- The final step in Arithmetic encoding calls for the generation of a number that falls within the range [*low*, *high*). The above algorithm will ensure that the shortest binary codeword is found.

ALGORITHM 7.6 Arithmetic Coding Decoder

```
BEGIN
  get binary code and convert to
  decimal value = value(code);
  Do
  {
      find a symbol s so that
            Range_low(s) <= value < Range_high(s);
      output s;
      low = Rang_low(s);
      high = Range_high(s);
      range = high - low;
      value = [value - low] / range;
  }
  Until symbol s is a terminator
END
```

Table 7.5 Arithmetic coding: decode symbols “CAEE$”

Value	Output Symbol	Low	High	Range
0.33203125	C	0.3	0.5	0.2
0.16015625	A	0.0	0.2	0.2
0.80078125	E	0.55	0.85	0.3
0.8359375	E	0.55	0.85	0.3
0.953125	$	0.9	1.0	0.1

Lossless Image Compression

- **Approaches of Differential Coding of Images:**
  - Given an original image $I(x, y)$, using a simple difference operator we can define a difference image $d(x, y)$ as follows:

$$d(x, y) = I(x, y) - I(x - 1, y) \qquad (7.9)$$

or use the discrete version of the 2D Laplacian operator to define a difference image $d(x, y)$ as

$$d(x, y) = 4\, I(x, y) - I(x, y - 1) - I(x, y + 1) - I(x + 1, y) - I(x - 1, y) \qquad (7.10)$$

- Due to *spatial redundancy* existed in normal images I, the difference image d will have a narrower histogram and hence a smaller entropy, as shown in Fig. 7.9.

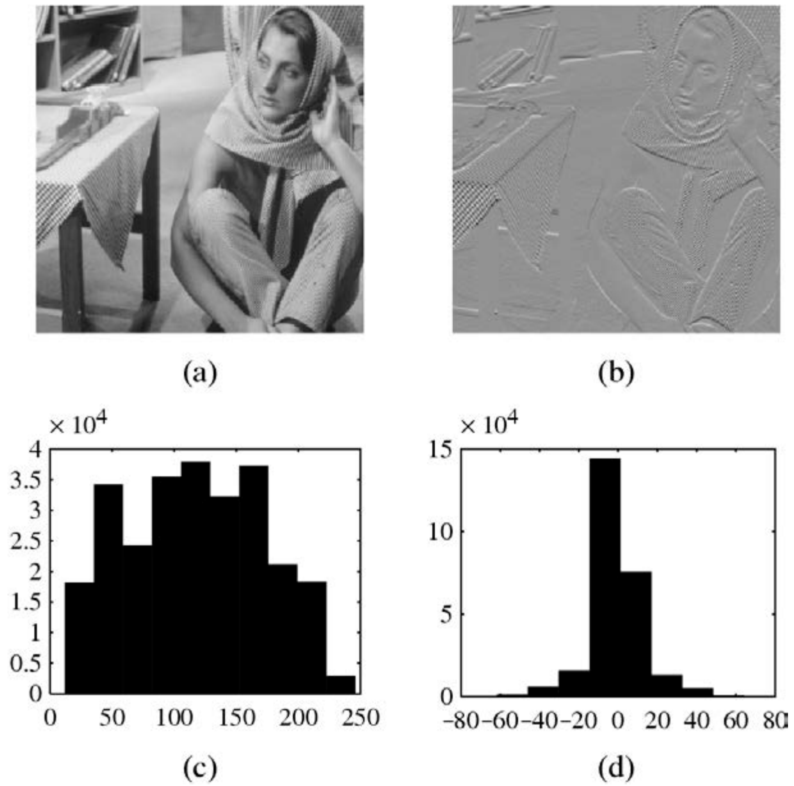


Fig. 7.9: Distributions for Original versus Derivative Images.

(a), (b): Original gray-level image and its partial derivative image

(c), (d): Histograms for original and derivative images.

Recall: Distribution of Gray-Level Intensities

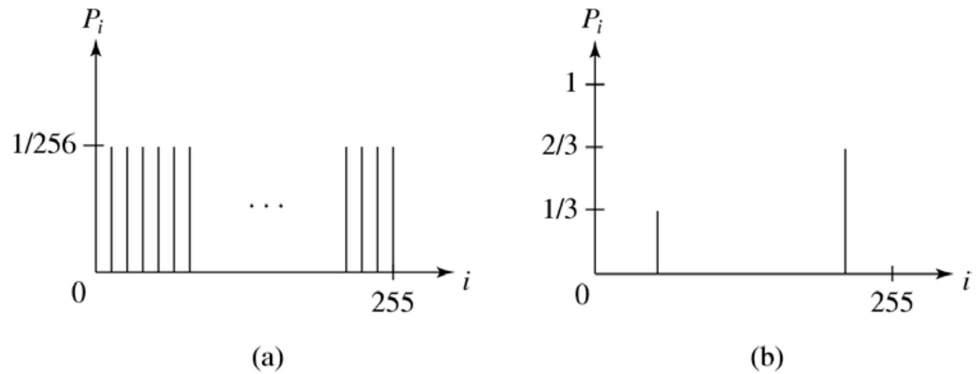


Fig. 7.2 Histograms for two gray-level images.

Fig. 7.2(a) shows the histogram of an image with *uniform* distribution of gray-level intensities. Its entropy is .

Fig. 7.2(b) shows the histogram of an image with two possible values. Its entropy is:

$$\eta = 8 \tag{7.4}$$

$$\eta = \frac{1}{3}\log_2 3 + \frac{2}{3}\log_2 \frac{3}{2} = 0.92$$

Lossless JPEG

- **Lossless JPEG**: A special case of the JPEG image compression.

The Predictive method

1. **Forming a differential prediction:**
    - A predictor combines the values of up to three neighboring pixels as the predicted value for the current pixel, indicated by 'X' in Fig. 7.10.
    - The predictor can use any one of the seven schemes listed in Table 7.6.
2. **Encoding:**
    - The encoder compares the prediction with the actual pixel value at the position 'X' and encodes the difference using one of the lossless compression techniques we have discussed, e.g., the Huffman coding scheme.

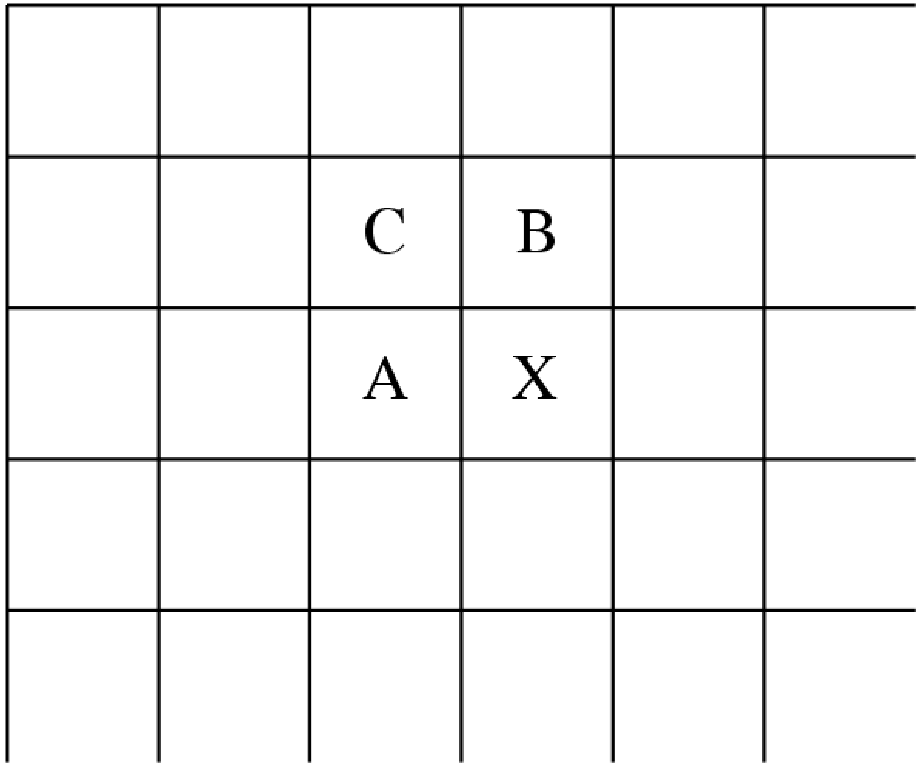


Fig. 7.10: Neighboring Pixels for Predictors in Lossless JPEG.

- **Note**: Any of A, B, or C has already been decoded before it is used in the predictor, on the decoder side of an encode-decode cycle.

Table 7.6: Predictors for Lossless JPEG

Predictor	Prediction
P1	A
P2	B
P3	C
P4	A + B – C
P5	A + (B – C) / 2
P6	B + (A – C) / 2
P7	(A + B) / 2

Table 7.7: Comparison with other lossless compression programs

Compression Program	Compression Ratio			
	Lena	Football	F-18	Flowers
Lossless JPEG	1.45	1.54	2.29	1.26
Optimal Lossless JPEG	1.49	1.67	2.71	1.33
Compress (LZW)	0.86	1.24	2.21	0.87
Gzip (LZ77)	1.08	1.36	3.10	1.05
Gzip -9 (optimal LZ77)	1.08	1.36	3.13	1.05
Pack(Huffman coding)	1.02	1.12	1.19	1.00

Further Exploration

- **Text books:**
  - *The Data Compression Book* by M. Nelson
  - *Introduction to Data Compression* by K. Sayood

Computer and Multimedia Networks

- Basics of Computer and Multimedia Networks
- Multiplexing Technologies
- Quality of Multimedia Data Transmission
- Multimedia over IP
- Multimedia over ATM Networks
- Transport of MPEG-4
- Media-on-Demand (MOD)

Basics of Computer and Multimedia Networks

- Computer networks are essential to modern computing.

- Multimedia networks share all major issues and technologies of computer networks.

- The ever-growing needs for various multimedia communications have made networks one of the most active areas for research and development.

- Various high-speed networks are becoming a central part of most contemporary multimedia systems.

OSI Network Layers

OSI Reference Model has the following network layers:

1. **Physical Layer**: Defines electrical and mechanical properties of the physical interface, and specifies the functions and procedural sequences performed by circuits of the physical interface.

2. **Data Link Layer**: Specifies the ways to establish, maintain and terminate a link, e.g., transmission and synchronization of data frames, error detection and correction, and access protocol to the Physical layer.

3. **Network Layer**: Defines the routing of data from one end to the other across the network. Provides services such as addressing, internetworking, error handling, congestion control, and sequencing of packets.

OSI Network Layers (Cont'd)

4. **Transport Layer**: Provides end-to-end communication between *end systems* that support end-user applications or services. Supports either *connection-oriented* or *connectionless* protocols. Provides error recovery and flow control.

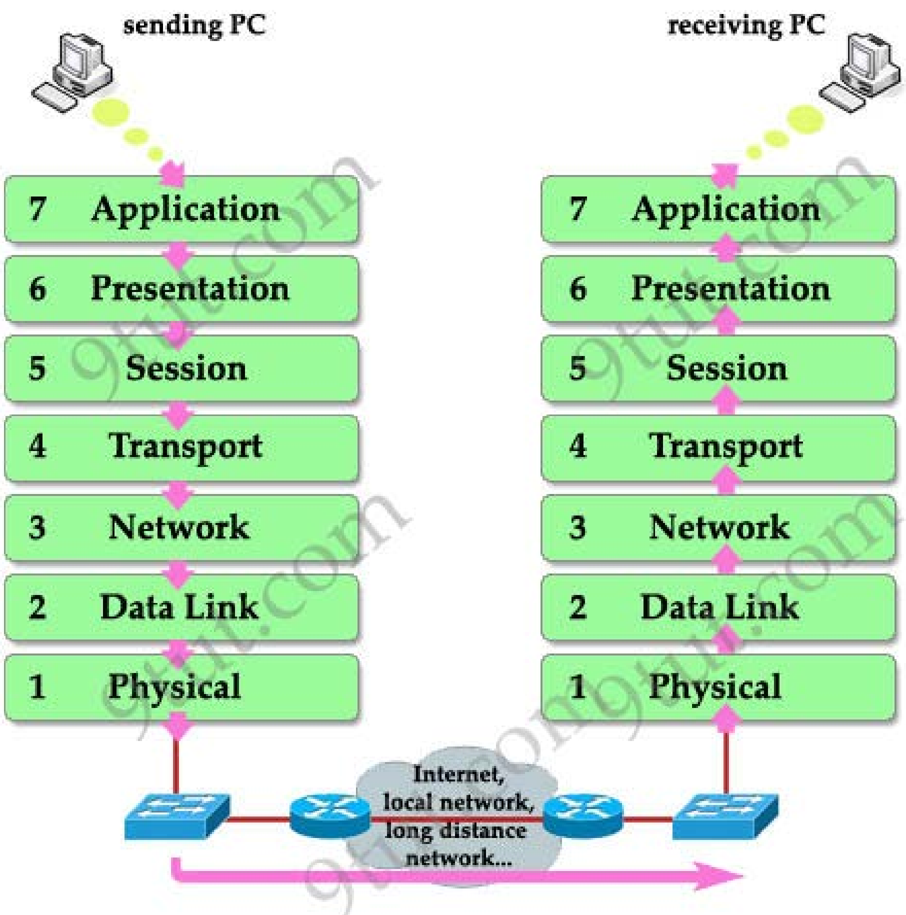
5. **Session Layer**: Coordinates interaction between user applications on different hosts, manages sessions (connections), e.g., completion of long file transfers.

6. **Presentation Layer**: Deals with the syntax of transmitted data, e.g., conversion of different data formats and codes due to different conventions, compression or encryption.

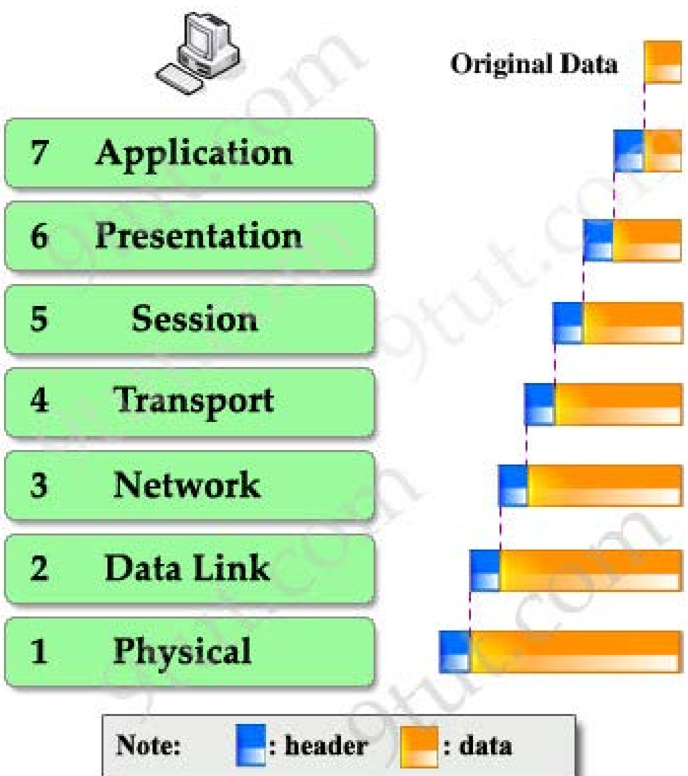
7. **Application Layer**: Supports various application programs and protocols, e.g., FTP, Telnet, HTTP, SNMP, SMTP/MIME, etc.

OSI Network Layers (Cont'd)

Please Do Not Throw Sausage Pizza Away



OSI Network Layers (Cont'd)



- When the information goes down through layers, a header is added to it. This is called "encapsulation".
- Each header can be understood only by the corresponding layer at the receiving side. Other layers only see that layer's header as a part of data.
- At the receiving side, corresponding header is stripped off in the same layer it was attached. This process is called "decapsulation".

http://www.9tut.com/osi-model-tutorial

TCP/IP Protocols

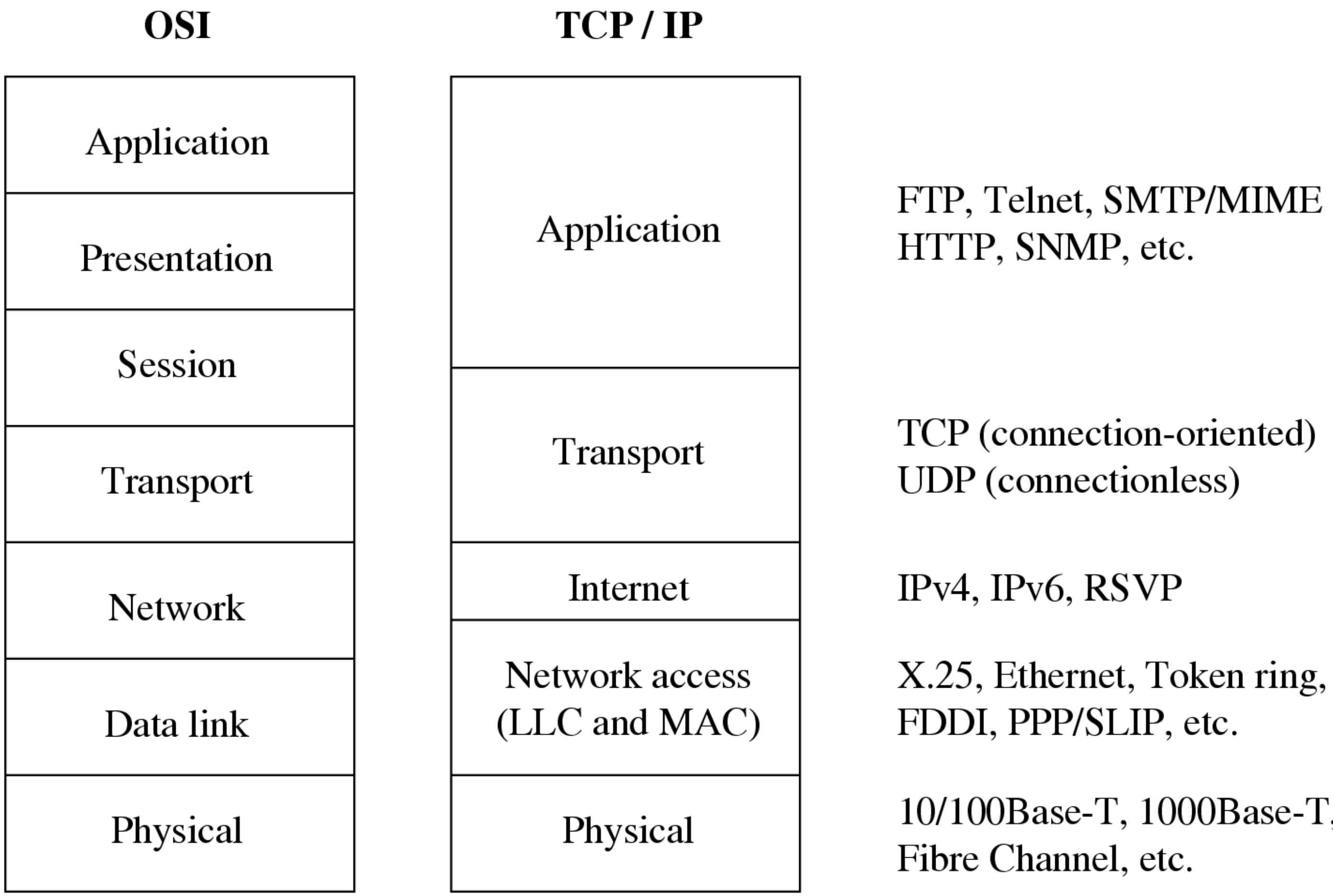


Fig: Comparison of OSI and TCP/IP protocol architectures

Transport Layer- TCP (Transmission Control Protocol)

- *Connection-oriented:* it provides reliable data transfer between pairs of communicating processes across the network.
- Established for packet switched networks only; data have to be packetized.
- Relies on the IP layer for delivering the message to the destination computer specified by its IP address.
- Provides message packetizing, error detection, retransmission, packet resequencing and multiplexing.
- Although reliable, the overhead of retransmission in TCP may be too high for many real-time multimedia applications such as streaming video —UDP can be used instead.

Transport Layer – TCP (Cont'd)

- Each TCP datagram header contains the source and destination port, sequence number, check sum, window field, acknowledgement number and other fields.

TCP header:

<table>
<tr><td>00</td><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td><td>08</td><td>09</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">Source Port</td><td colspan="16">Destination Port</td></tr>
<tr><td colspan="32">Sequence Number</td></tr>
<tr><td colspan="32">Acknowledgment Number</td></tr>
<tr><td colspan="4">Data Offset</td><td colspan="3">reserved</td><td colspan="3">ECN</td><td colspan="6">Control Bits</td><td colspan="16">Window</td></tr>
<tr><td colspan="16">Checksum</td><td colspan="16">Urgent Pointer</td></tr>
<tr><td colspan="32">Options and padding :::</td></tr>
<tr><td colspan="32">Data :::</td></tr>
</table>

http://www.networksorcery.com/enp/protocol/tcp.htm

Transport Layer - UDP (User Datagram Protocol)

- *Connectionless*: the message to be sent is a single Datagram.
- The only thing UDP provides is multiplexing and error detection through a Checksum.
- The source port number in UDP header is optional since there is no acknowledgment.
- Much faster than TCP, however it is unreliable:
  - In most real-time multimedia applications (e.g., streaming video or audio), packets that arrive late are simply discarded.
  - Flow control, and congestion avoidance, more realistically error concealment must be explored for acceptable Quality of Service (QoS).

Network Layer - IP (Internet Protocol)

- Two basic services: **packet addressing** and **packet fragmentation**.

- **Packet addressing**:

  - The IP protocol provides for a global addressing of computers across all interconnected networks.

  - For an IP packet to be transmitted within LANs, either broadcast based on hubs or point-to-point transmission based on switch is used.

  - For an IP packet to be transmitted across WANs, Gateways or routers are employed, which use routing tables to direct the messages according to destination IP addresses.

Network Layer — IP (Internet Protocol) (Cont'd)

- The IP layer also has to:
  - translate the destination IP address of incoming packets to the appropriate network address.
  - identify for each destination IP the next best router IP through which the packet should travel based on routing table.

- Since the best route can change depending on node availability, network congestion and other factors, routers have to communicate with each other to determine the best route for groups of IPs. The communication is done using *Internet Control Message Protocol (ICMP)*.

- IP is *connectionless* — provides no end-to-end flow control, packets could be received out of order, and dropped or duplicated.

Network Layer — IP (Internet Protocol) (Cont'd)

- **Packet fragmentation**: performed when a packet travels over a network that only accepts packets of a smaller size.
  - In that case, IP packets are split into the required smaller size, sent over the network to the next hop, and reassembled and resequenced.
- IP versions:
  - IPv4 (IP version 4): IP addresses are *32* bit numbers, usually specified using *dotted decimal notation* (e.g. 128.77.149.63) — running out of new IP addresses soon (projected in year 2008).
  - IPv6 (IP version 6): The *next generation IP (IPng)* - adopts 128-bit addresses, allowing $2^{128} \approx 3.4 \times 10^{38}$ addresses.

Multiplexing Technologies

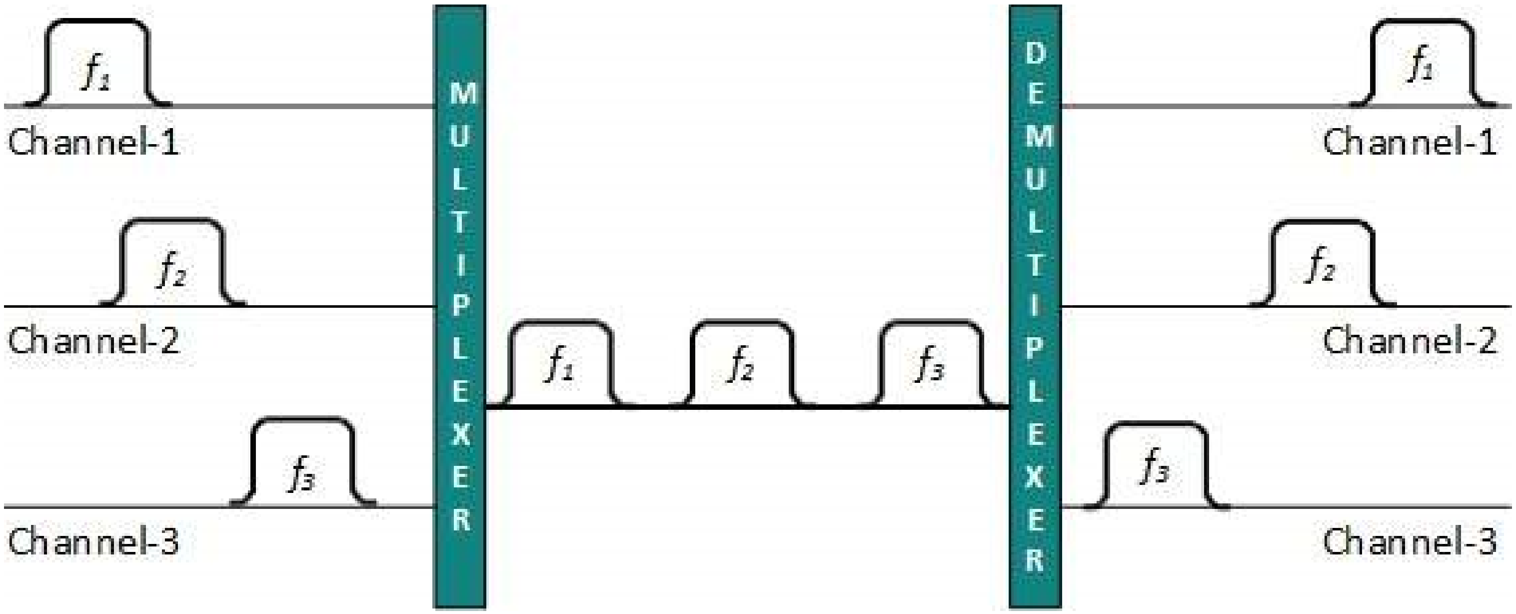
- **Basics of Multiplexing**

1. **FDM (Frequency Division Multiplexing)** — Multiple *channels* are arranged according to their frequency (e.g. radios and TVs):

   - For FDM to work properly, analog signals must be *modulated* so that the signal occupies a bandwidth B_s centered at f_c — *carrier* frequency unique for each channel.

   - The receiver uses a band-pass filter tuned for the particular channel-of-interest to capture the signal, and then uses a demodulator to decode it.

   - Basic modulation techniques: *Amplitude Modulation (AM)*, *Frequency Modulation (FM)*, *Phase Modulation (PM)*, and *Quadrature Amplitude Modulation (QAM)*.


f_1
f_2
f_3
Channel-1
Channel-2
Channel-3
MULTIPLEXER
DEMULTIPLEXER

2. **WDM (Wavelength Division Multiplexing)**: A variation of FDM for data transmission in optical fibers:

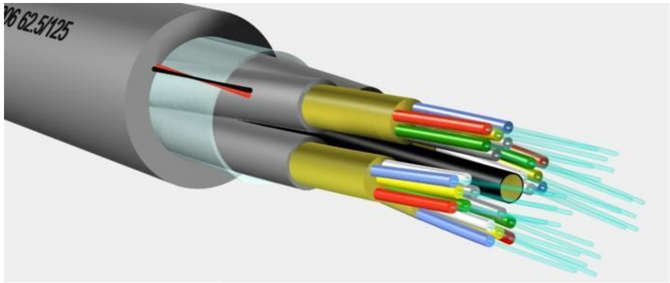
- Light beams representing channels of different wave-lengths are combined at the source, and split again at the receiver.

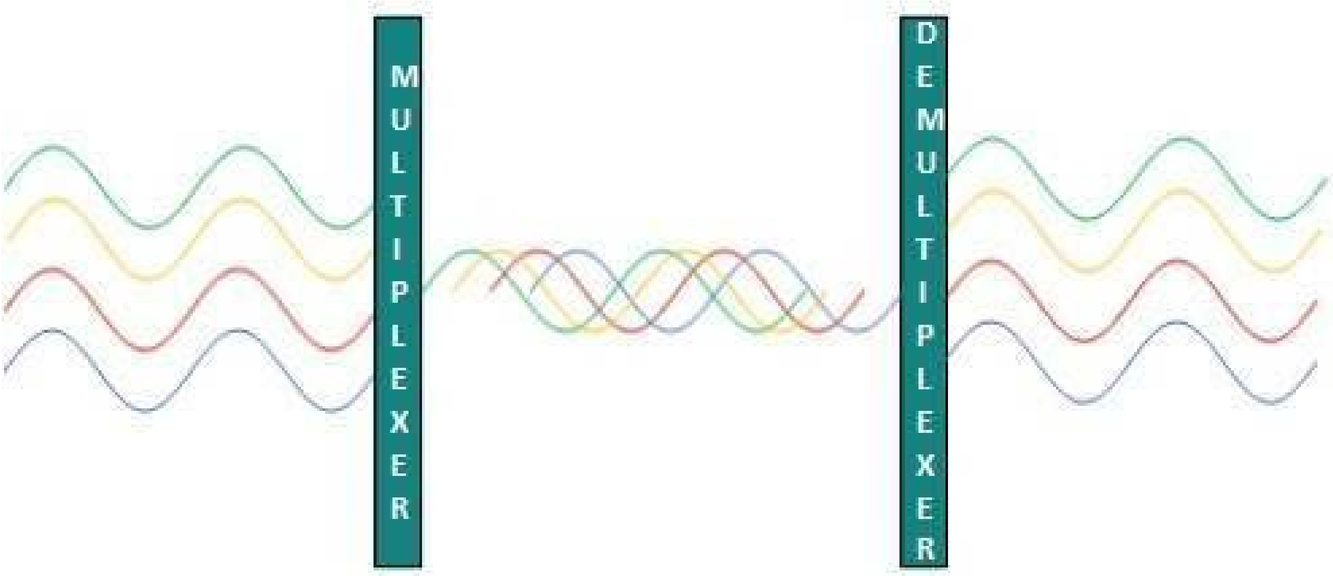
- The capacity of WDM is tremendous — a huge number of channels can be multiplexed (aggregate bit-rate can be up to dozens of terabits per second).

- Two variations of WDM:

  a) **DWDM** (Dense WDM): employs densely spaced wavelengths so as to allow a larger number of channels than WDM (e.g., more than 32).

  b) **WWDM** (Wideband WDM): allows the transmission of color lights with a wider range of wavelengths (e.g., 1310 to 1557 nm for long reach and 850 nm for short reach) to achieve a larger capacity than WDM.



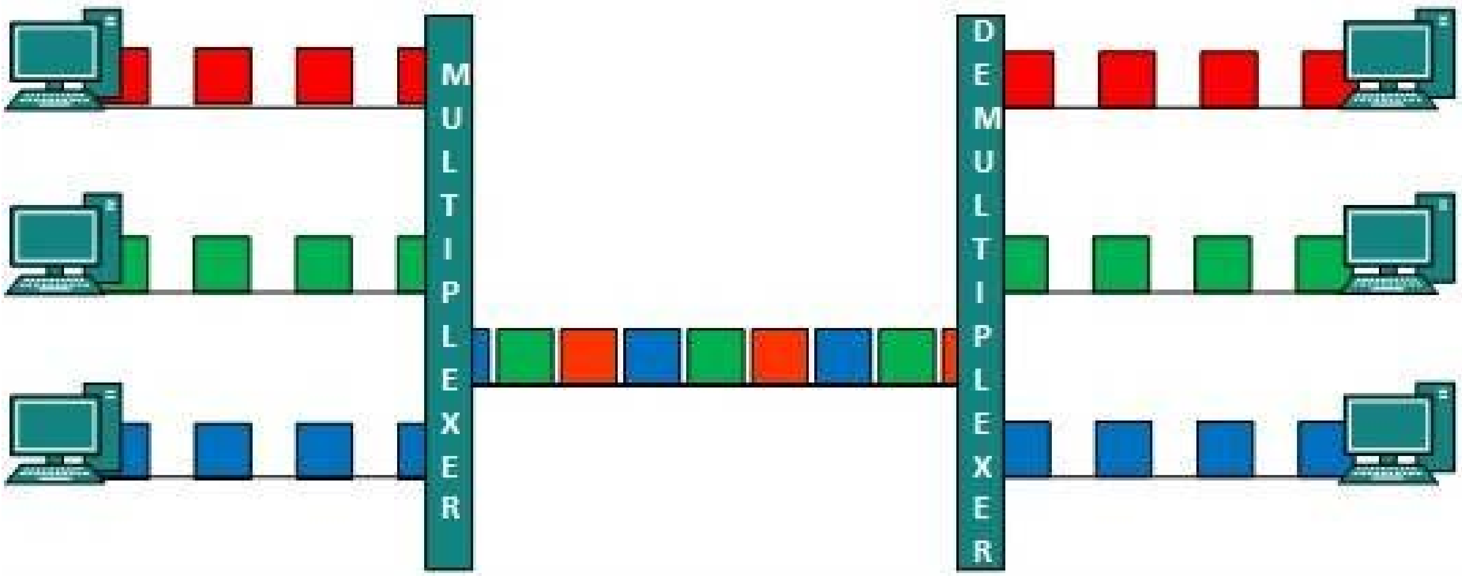

M
U
L
T
I
P
L
E
X
E
R
D
E
M
U
L
T
I
P
L
E
X
E
R

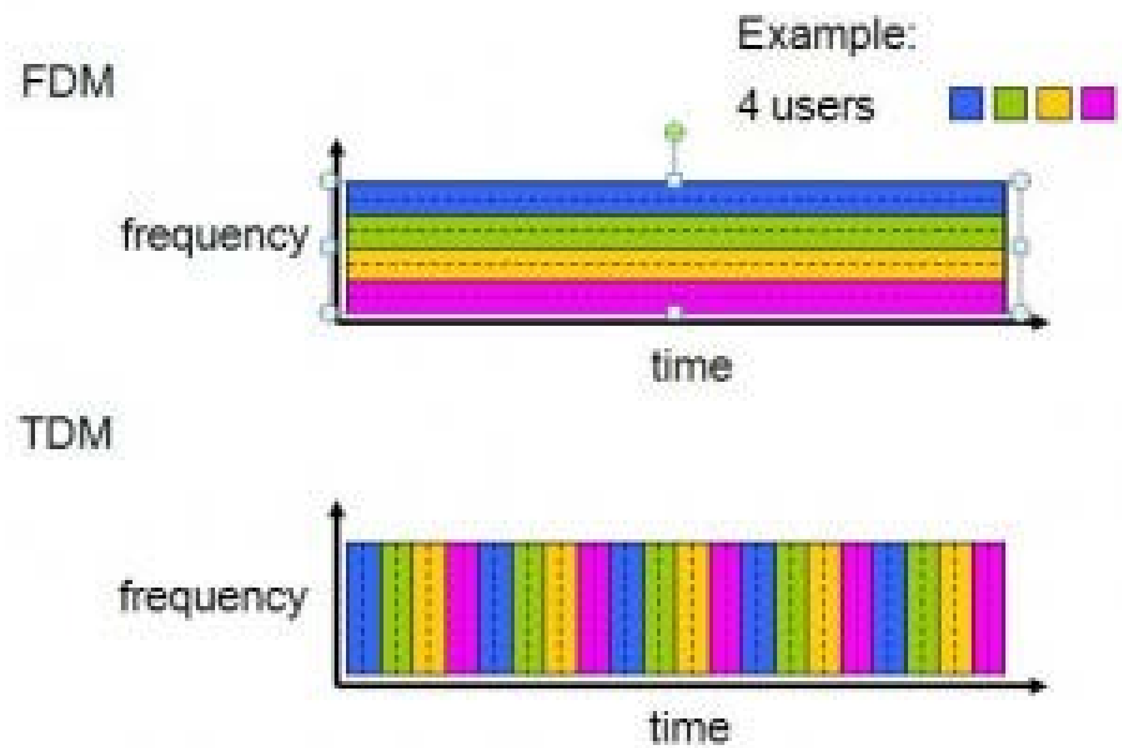
3. **TDM (Time Division Multiplexing)** — A technology for directly multiplexing digital data:

- If the source data is analog, it must first be digitized and converted into PCM (Pulse Code Modulation).
- Multiplexing is performed along the time (t) dimension.
  Multiple buffers are used for m ($m > 1$) channels.
- Two variations of TDM:

  a) **Synchronous TDM**: Each of the m buffers is scanned in turn and treated equally. If, at a given time slot, some sources (accordingly buffers) do not have data to transmit the slot is wasted.

  b) **Asynchronous TDM**: Only assign k ($k < m$) time slots to scan the k buffers that are likely to have data to send (based on statistics) — has the potential of having a higher throughput given the same carrier data rate.


M U L T I P L E X E R
D E M U L T I P L E X E R



FDM vs. TDM

ADSL (Asymmetric Digital Subscriber Line)

- Adopts a higher data rate downstream and lower data rate upstream, hence *asymmetric*.
- Makes use of existing telephone twisted-pair lines to transmit QAM (Quadrature Amplitude Modulated) digital signals.
- Bandwidth on ADSL lines: 1 MHz or higher.
- ADSL uses FDM to multiplex three channels:
  a) High speed (1.5 to 9 Mbps) downstream channel at the high end of the spectrum
  b) Medium speed (16 to 640 kbps) duplex channel.
  c) POTS (Plain Old Telephone Service) channel at the low end (next to DC, 0-4 kHz) of the spectrum.

ADSL (Asymmetric Digital Subscriber Line)

- Because signals attenuate quickly on twisted-pair lines, and noise increases with line length, ADSL has the distance limitation when using only ordinary twisted-pair copper wires.

Data rate	Wire Size	Distance
1.544 Mbps	0.5 mm	5.5km
1.544 Mbps	0.4 mm	4.6km
6.1 Mbps	0.5 mm	3.7km
6.1 Mbps	0.4 mm	2.7km

Table. History of Digital Subscriber Lines

Name	Meaning	Data Rate	Mode
V.32 or V.34	Voice Band Modems	1.2 to 56 kbps	Duplex
DSL	Digital Subscriber Line	160 kbps	Duplex
HDSL	High Data Rate Digital Subscriber Line	1.544 Mbps or 2.048 Mbps	Duplex
SDSL	Single Line Digital Subscriber Line	1.544 Mbps or 2.048 Mbps	Duplex
ADSL	Asymmetric Digital Subscriber Line	1.5 to 9 Mbps 16 to 640 kbps	Down Up
VDSL	Very high data rate Digital Subscriber Line	13 to 52 Mbps 1.5 to 2.3 Mbps	Down Up

Table offers a brief history of various digital subscriber lines (**xDSL**).

Characteristics of Multimedia Data

- **Voluminous** — they demand very high data rates, possibly dozens or hundreds of Mbps.

- **Real-time and interactive** — they demand low delay and synchronization between audio and video for “lip sync”. In addition, applications such as video conferencing and interactive multimedia also require two-way traffic.

- **Sometimes bursty** — data rates fluctuate drastically, e.g., no traffic most of the time but burst to high volume in video-on-demand.

Quality of Multimedia Data Transmission

- **Quality of Service (QoS)** depends on many parameters:
  - **Data rate**: a measure of transmission speed.
  - **Latency (maximum frame/packet delay)**: maximum time needed from transmission to reception.
  - **Packet loss or error**: a measure (in percentage) of error rate of the packetized data transmission.
  - **Jitter**: a measure of smoothness of the audio/video playback, related to the variance of frame/packet delays.
  - **Sync skew**: a measure of multimedia data synchronization.

Multimedia Service Classes

- **Real-Time** (also *Conversational*): two-way traffic, low latency and jitter, possibly with prioritized delivery, e.g., voice telephony and video telephony.
- **Priority Data**: two-way traffic, low loss and low latency, with prioritized delivery, e.g., E-commerce applications.
- **Silver**: moderate latency and jitter, strict ordering and sync. One-way traffic, e.g., streaming video, or two-way traffic (also *Interactive*), e.g., web surfing, Internet games.
- **Best Effort** (also *Background*): no real-time requirement, e.g., downloading or transferring large files (movies).
- **Bronze**: no guarantees for transmission.

Requirement on Network Bandwidth / Bit-rate

Application	**Speed Requirement**
Telephone	16 kbps
Audio-conferencing	32 kbps
CD-quality audio	128–192 kbps
Digital music (QoS)	64–640 kbps
H. 261	64 kbps–2 Mbps
H. 263	$<$ 64 kbps
DVI video	1.2–1.5 Mbps
MPEG-1 video	1.2–1.5 Mbps
MPEG-2 video	4–60 Mbps
HDTV (compressed)	$>$ 20 Mbps
HDTV (uncompressed)	$>$ 1 Gbps
MPEG-4 video-on-demand (QoS)	250–750 kbps
Videoconferencing (QoS)	384 kbps–2 Mbps

Tolerance of Latency and Jitter in Digital Audio and Video

Application	Avg Latency Tolerance (msec)	Avg Jitter Tolerance (msec)
Low-end videoconf. (64 kbps)	300	130
Compressed voice (16 kbps)	30	130
MPEG NTSC video (1.5 Mbps)	5	7
MPEG audio (256 kbps)	7	9
HDTV video (20 Mbps)	0.8	1

Perceived QoS

- Although QoS is commonly measured by the above technical parameters, QoS itself is a "**collective effect of service performances that determine the degree of satisfaction of the user of that service**".
- In other words, it has everything to do with how the user *perceives* it. For example, in real-time multimedia:
  - Regularity is more important than latency (i.e., jitter and quality fluctuation are more annoying than slightly longer waiting).
  - Temporal correctness is more important than the sound and picture quality (i.e., ordering and synchronization of audio and video are of primary importance).
  - Humans tend to focus on one subject at a time. User focus is usually at the center of the screen, and it takes time to refocus especially after a scene change.

QoS for IP Protocols

- **IP** is a *best-effort* communications technology-hard to provide QoS over IP by current routing methods.
  - Abundant bandwidth improves QoS, but unlikely to be available everywhere over a complex networks.

Sol1:

- **DiffServ** (Differentiated Service) uses DiffServ code [*TOS* (Type of Service) octet in IPv4 packet, and Traffic Class octet in IPv6 packet] to classify packets to enable their differentiated treatment.
  - Widely deployed in intra-domain networks and enterprise networks as it is simpler and scales well.
  - Emerging as the QoS technology in conjunction with other QoS.

QoS for IP Protocols (Cont'd)

Sol2:

- **MPLS** (*Multiple Protocol Label Switching*) facilitates the marriage of IP to OSI Layer 2 technologies.
  - Creates tunnels: **Label Switched Paths (LSP)** — IP network becomes connection-oriented.
  - Main advantages of MPLS:
    a) Support Traffic Engineering (TE), which is used essentially to control traffic flow.
    b) Support VPN (Virtual Private Network).
    c) Both TE and VPN help delivery of QoS for multimedia data.

Prioritized Delivery

- Used to alleviate the perceived deterioration (high packet loss or error rate) in network congestion.
- **Prioritization for types of media**:
  - Transmission algorithms can provide prioritized delivery to different media.
- **Prioritization for uncompressed audio**:
  - PCM audio bitstreams can be broken into groups of every n^{th} sample.
- **Prioritization for JPEG image**:
  - The different *scans* in Progressive JPEG and different resolutions of the image in Hierarchical JPEG can be given different priorities.
- **Prioritization for compressed video**:
  - Set priorities to minimize playback delay and jitter by giving highest priority to I-frames for their reception, and lowest priority to B-frames.

Multimedia over IP

- A *broadcast* message is sent to all nodes in the domain, a *unicast* message is sent to only one node, and a *multicast* message is sent to a set of specified nodes.
- IP-multicast is vital for applications such as mailing list, bulletin boards, group file transfer, audio/video on demand, audio/videoconferencing

- **MBone** (Multicast Backbone) — based on the IP-multicast technology:
  - Used for audio and video conferencing on the Internet .

  - Uses a subnetwork of routers (*mrouters*) that support multicast to forward multicast packets.

Multimedia over IP

- **IP-Multicast**:
  - Anonymous membership: the source host multicasts to one of the IP-multicast addresses — doesn't know who will receive.
  - Potential problem: too many packets will be traveling and alive in the network — use time-to-live (TTL) in each IP packet.
  - based on UDP-limited reliability.

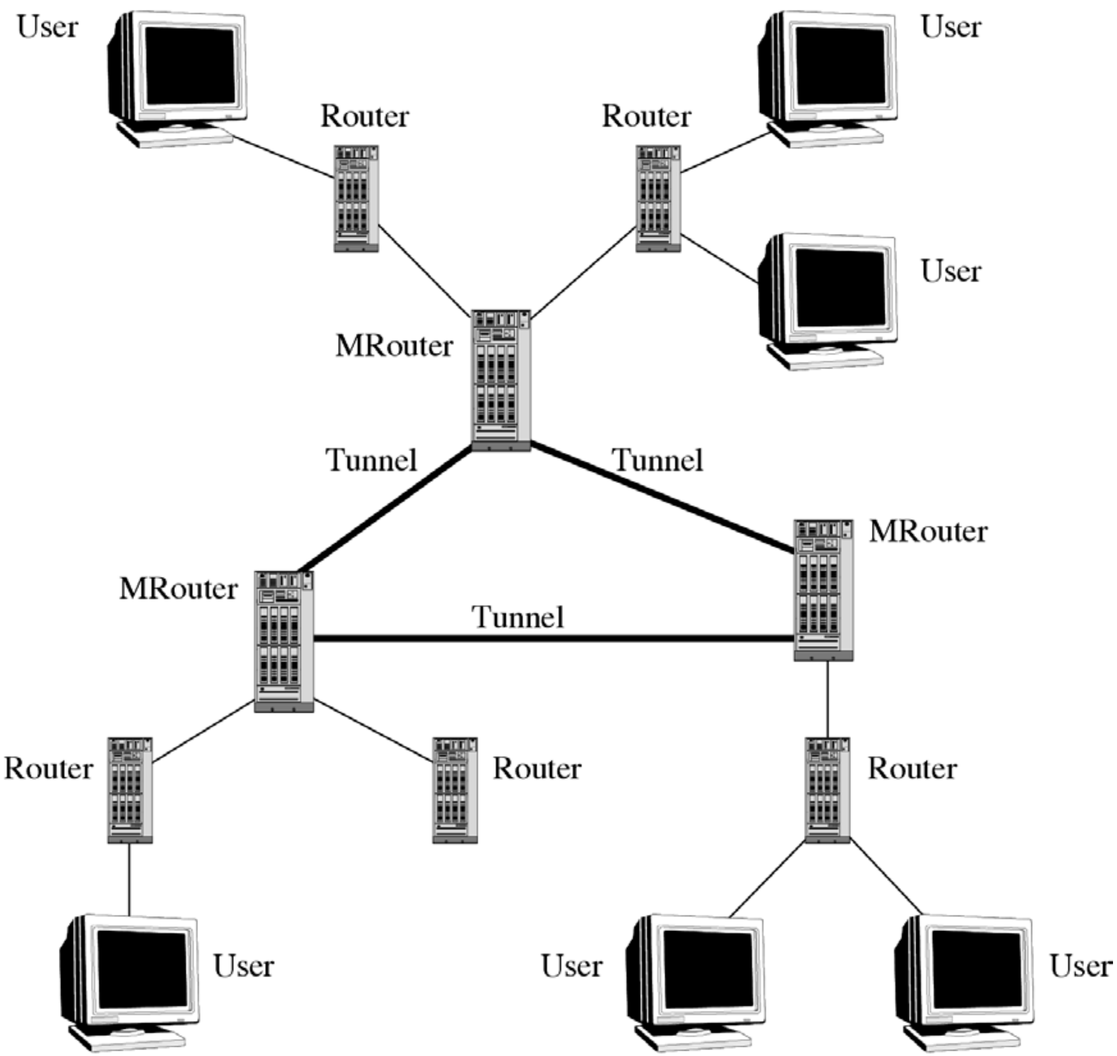


Fig: Tunnels for IP Multicast in MBone.

Internet Group Management Protocol (IGMP)

- Designed to help the maintenance of multicast groups.

- Two special types of IGMP messages are used:
  - **Query** messages are multicast by routers to all local hosts to inquire group membership.
  - **Report** is used to respond to a query and to join groups.

- On receiving a query, members wait for a random time before responding.

- Routers periodically query group membership, and declare themselves group members if they get a response to at least one query. If no responses occur after a while, they declare themselves as non members.

RTP (Real-time Transport Protocol)

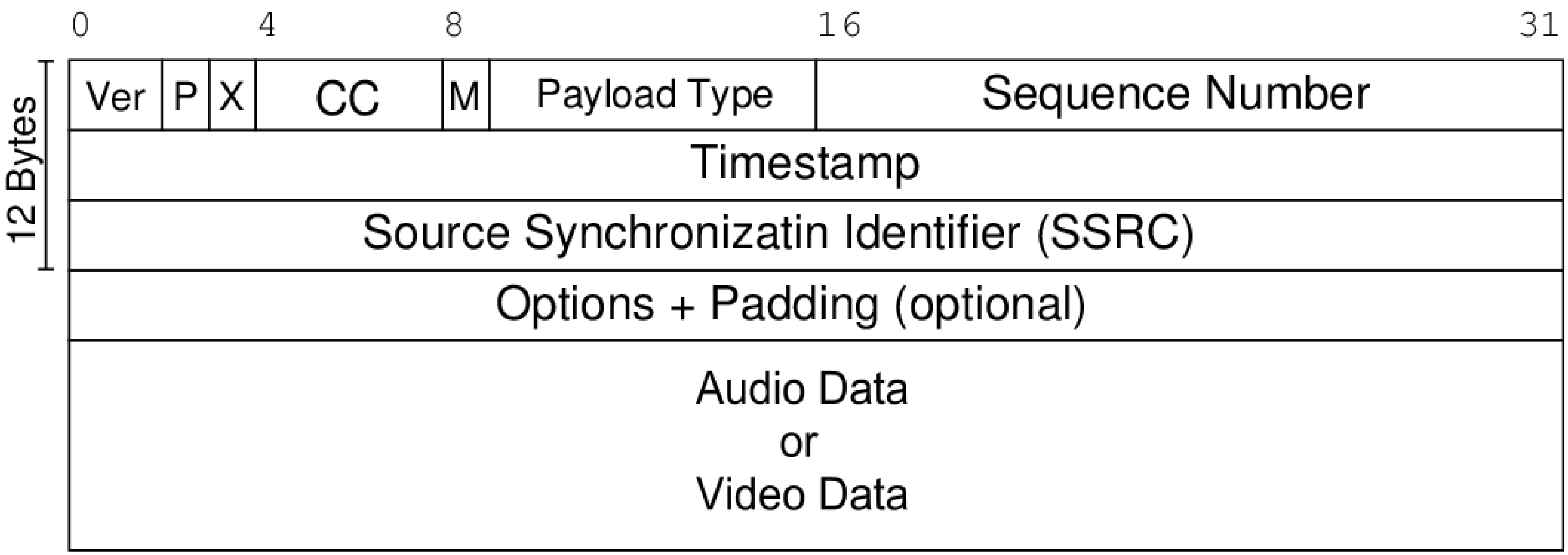
- Designed for the transport of real-time data, such as audio and video streams:
  - Primarily intended for multicast.
- Usually runs on top of UDP which provides efficient (but less reliable) connectionless datagram service:
  - RTP must create its own *timestamping* and *sequencing* mechanisms to ensure the ordering.

RTCP (Real Time Control Protocol)

- A companion protocol of RTP:
  - Monitors QoS in providing feedback to the server (sender) and conveys information about the participants of a multi-party conference.
  - Provides the necessary information for audio and video synchronization.

- RTP and RTCP packets are sent to the same IP address (multicast or unicast) but on different ports.

RTCP (Real Time Control Protocol)



RTP packet header

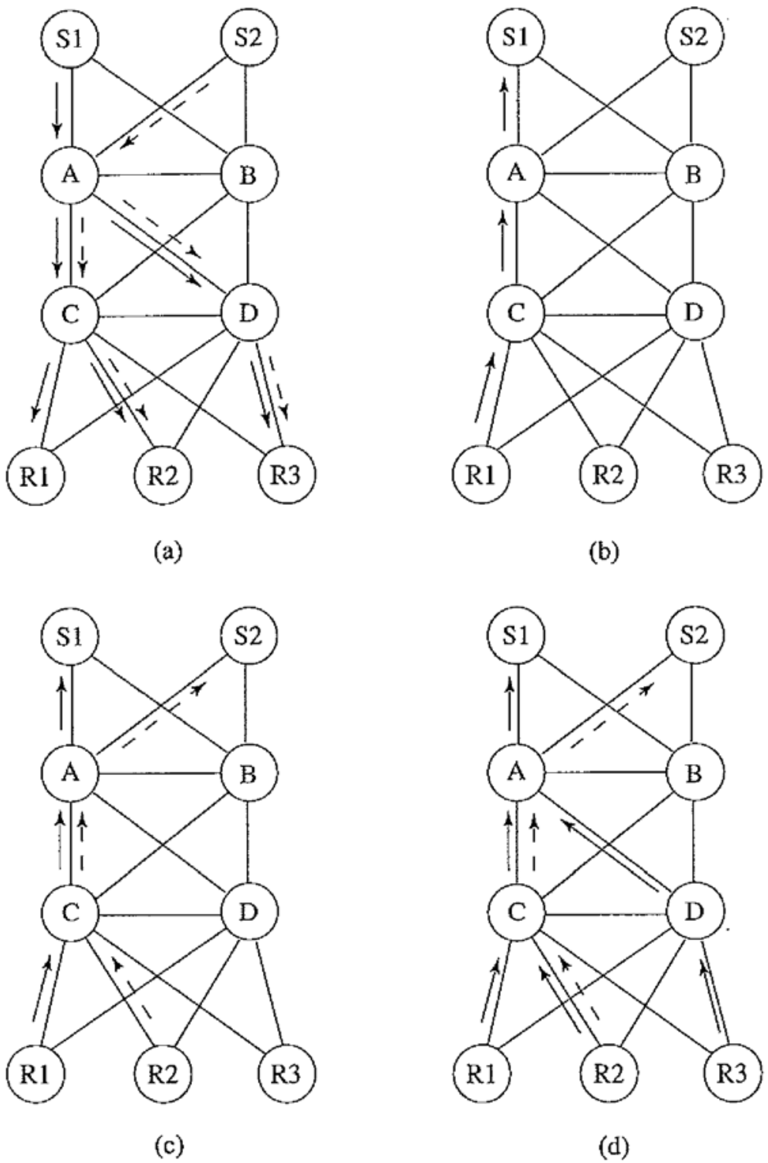
RSVP (Resource ReSerVation Protocol)

- Aim to guarantee desirable QoS, mostly for multicast but also applicable to unicast.

  - A general communication model supported by RSVP consists of *m* senders and *n* receivers, possibly in various multicast groups.

- The most important messages of RSVP:
  1. A **Path** message is initiated by the sender, and contains information about the sender and the path (e.g., the previous RSVP hop).
  2. A **Resv** message is sent by a receiver that wishes to make a reservation.

RSVP



Main Challenges of RSVP

- Plenty of senders and receivers compete for the limited network bandwidth.

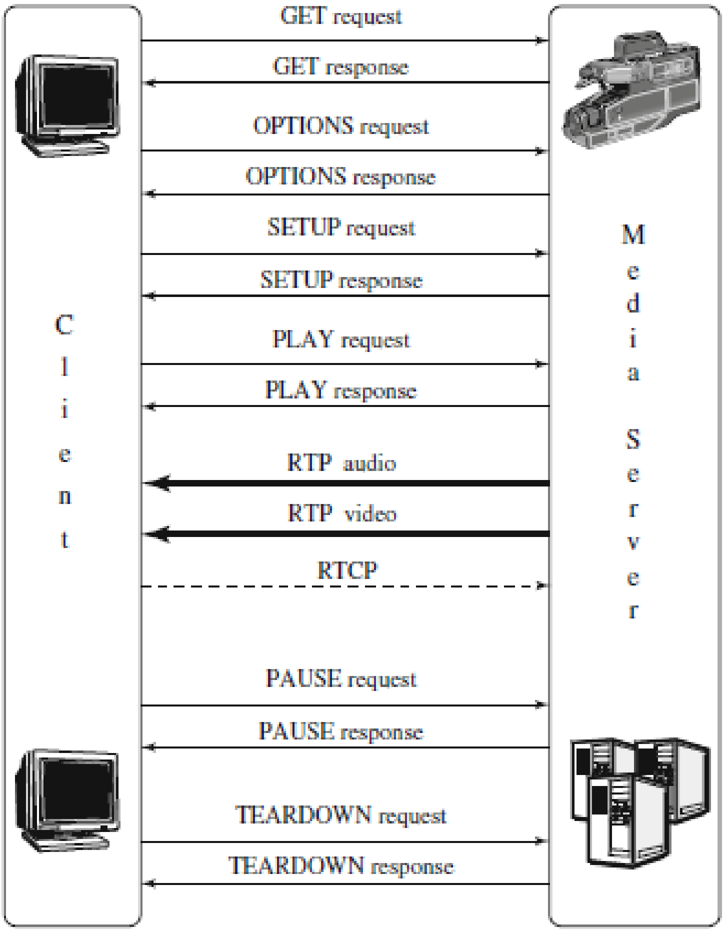
- The receivers can be heterogeneous in demanding different contents with different QoS.

- They can be dynamic by joining or quitting multicast groups at any time.

RTSP (Real Time Streaming Protocol)

- **Streaming Audio and Video**: Audio and video data that are transmitted from a stored media server to the client in a data stream that is almost instantly decoded.
- **RTSP Protocol**: for communication between a client and a stored media server.
  1. **Requesting presentation description**: the client issues a DESCRIBE request to the Stored Media Server to obtain the presentation description — media types, frame rate, resolution, codec, etc.
  2. **Session setup**: the client issues a SETUP to inform the server of the destination IP address, port number, protocols, TTL (for multicast).
  3. **Requesting and receiving media**: after receiving a PLAY, the server started to transmit streaming audio/video data using RTP.
  4. **Session closure**: TEARDOWN closes the session.

RTSP (Real Time Streaming Protocol)



A possible scenario of RTSP operations

Internet Telephony

- Main advantages of Internet telephony over *POTS* (*Plain Old Telephone Service*):
  - Uses *packet-switching* — network usage is much more efficient (voice communication is bursty and VBR-encoded).
  - With the technologies of *multicast or multipoint* communication, multi-party calls are not much more difficult than two-party calls.
  - With advanced multimedia data compression techniques, various degrees of *QoS* can be supported and dynamically adjusted according to the network traffic.
  - Good *graphics user interfaces* can be developed to show available features and services, monitor call status and progress, etc.

Internet Telephony (Cont'd)

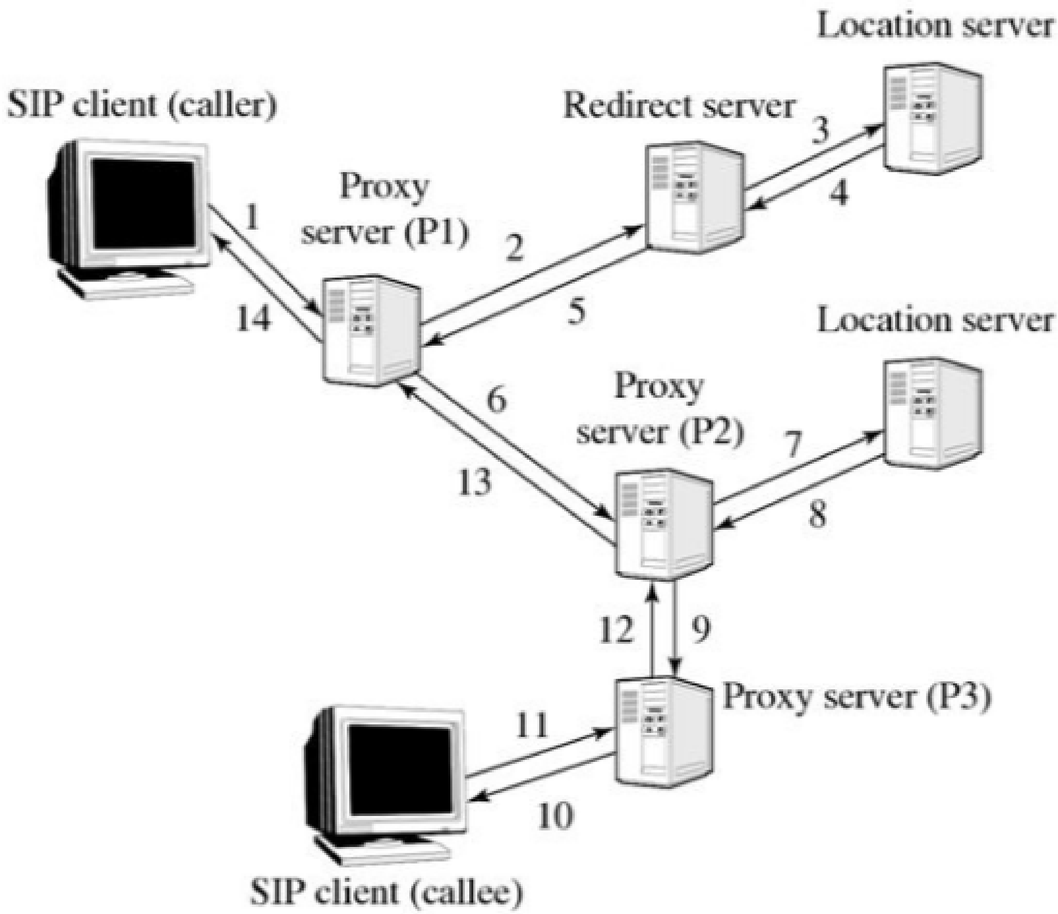
- The transport of real-time audio (and video) in Internet telephony is supported by RTP (whose control protocol is RTCP).

- Streaming media is handled by RTSP and Internet resource reservation is taken care of by RSVP.

SIP (Session Initiation Protocol)

- An *application-layer* control protocol in charge of the establishment and termination of sessions in Internet telephony.
  - SIP is a text-based protocol, also a client-server protocol.
- SIP can advertise its session using email, news group, web pages or directories, or *SAP* — a multicast protocol.
- The *methods* (commands) for clients to invoke:
  - **INVITE**: invites callee(s) to participate in a call.
  - **ACK**: acknowledges the invitation.
  - **OPTIONS**: enquires media capabilities without setting up a call.
  - **CANCEL**: terminates the invitation.
  - **BYE**: terminates a call.
  - **REGISTER**: sends user's location info to a Registrar (a SIP server).

SIP



Location server
SIP client (caller)
Redirect server
3
Proxy
server (P1)
1
4
2
14
5
Location server
6
Proxy
server (P2)
7
13
8
12
9
Proxy server (P3)
11
10
SIP client (callee)

Multimedia over ATM Networks

- ATM forum supports various **Video Bit-rates:**
  - **CBR** (Constant Bit Rate): if the allocated bit-rate of CBR is too low, then cell loss and distortion of the video content are inevitable.
  - **VBR** (Variable Bit Rate): the most commonly used video bit-rate for compressed video, can be further divided into:

    * rt-VBR (real-time Variable Bit Rate): for compressed video.

    * nrt-VBR (non real-time Variable Bit Rate): for specified QoS.
  - **ABR** (Available Bit Rate): data transmission can be backed off or buffered due to congestion. Cell loss rate and minimum cell data rate can sometimes be specified.
  - **UBR** (Unspecified Bit Rate): no guarantee on any quality parameter.

Multicast over ATM

- Multicast in ATM networks had several challenges:
  - ATM is connection-oriented; hence ATM multicasting needs to set up all multipoint connections.
  - QoS in ATM must be negotiated at the connection set-up time and be known to all switches.
- It is difficult to support multipoint-to-point or multipoint-to-multipoint connections in ATM
  - Do not keep track of multiplexer number or sequence number. It cannot reassemble the data correctly at the receiver side if cells from different senders are interleaved at their reception.

Transport of MPEG-4

- **DMIF**: An interface between multimedia applications and their transport. It supports:

  1. Remote interactive network access (IP, ATM, PSTN, ISDN, mobile).
  2. Broadcast media (cable or satellite).
  3. Local media on disks.

- The following fig. shows the integration of delivery through three types of communication mediums.

- **MPEG-4 over IP**: MPEG-4 sessions can be carried over IP-based protocols such as RTP, RTSP, and HTTP.

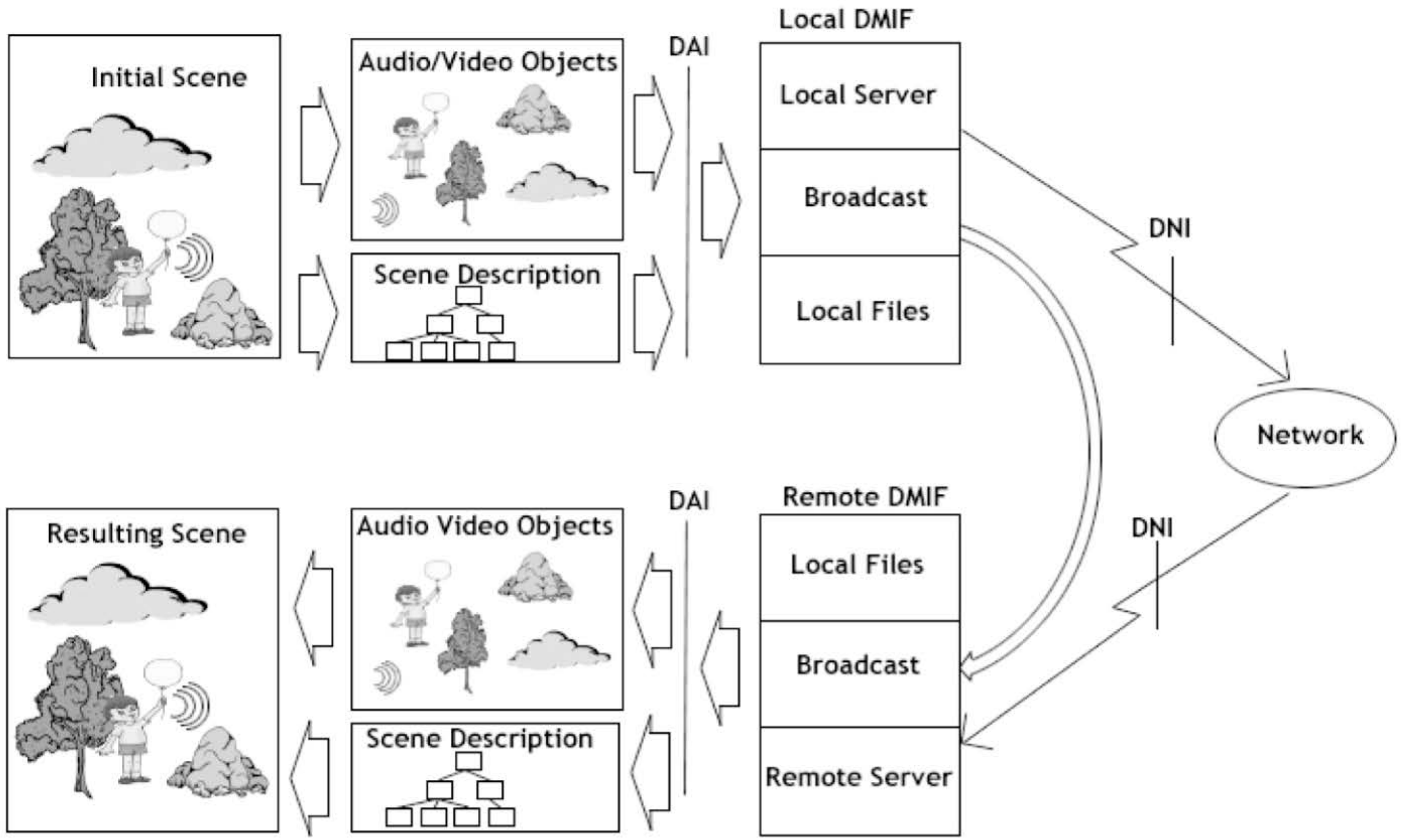


Fig. DMIF — the multimedia content delivery integration framework.

Media-on-Demand (MOD)

- **Interactive TV (ITV) and Set-top Box (STB)**
  - ITV supports activities such as:
    1. TV (basic, subscription, pay-per-view).
    2. Video-on-demand (VOD).
    3. Information services (news, weather, magazines, sports events, etc.).
    4. Interactive entertainment (Internet games, etc.).
    5. E-commerce (on-line shopping, stock trading).
    6. Access to digital libraries and educational materials.

Set-top Box (STB)

- Set-top Box (STB) generally has the following components:

1. **Network Interface and Communication Unit**: including tuner and demodulator, security devices, and a communication channel.

2. **Processing Unit**: including CPU, memory, and special-purpose operating system for the STB.

3. **Audio/Video Unit**: including audio and video (MPEG-2 and 4) decoders, DSP (Digital Signal Processor), buffers, and D/A converters.

4. **Graphics Unit**: supporting real-time 3D graphics for animations and games.

5. **Peripheral Control Unit**: controllers for disks, audio and video I/O devices (e.g., digital video cameras), CD/DVD reader and writer, etc.

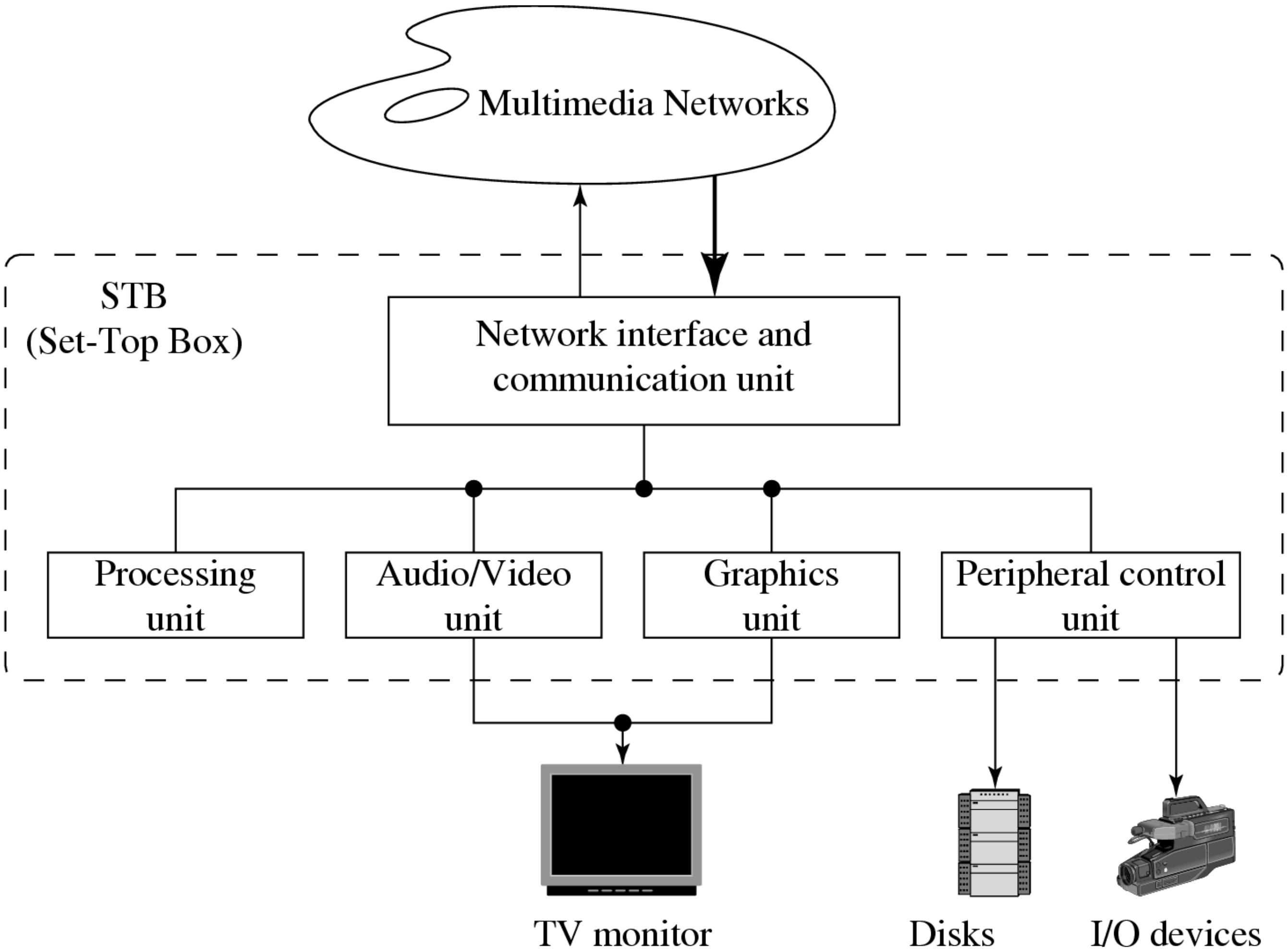


Fig: General Architecture of STB (Set-top Box).

Further Exploration

- **Text books:**
  - *Computer Networks* by A.S. Tanenbaum
  - *Data & Computer Communications* by W. Stalling

- **Further Topics**:
  - SONET FAQ, etc.
  - xDSL introductions at DSL Forum website.
  - Introductions and White Papers on ATM.
  - FAQ and White Papers on 10 Gigabit Ethernet at the Alliance website.
  - IEEE 802 standards.
  - IETF RFCs: IPv6 (Internet Protocol, Version 6).