Sardar Patel University,Balaghat(M.P)
School of Engineering & Technology
Department of CSE
Subject Name: Computer Graphics & Multimedia
Subject Code: CSE504
Semester: 5th

Unit-I

Computer Graphics

The computer graphics is one of the most effective and commonly used ways to communicate the processed information to the users. The term computer graphics includes almost everything on computers that is not text or sound. Today almost every computer can do some graphics, and people have even come to expect to control their computer through icons and pictures rather than just by typing. In computer graphics we draw picture on computers also called rendering. The pictures can be photographs, drawings, movies, or simulations - pictures of things, which do not yet exist and maybe could never exist. Or they may be pictures from places we cannot see directly, such as medical images from inside your body. We spend much of our time improving the way computer pictures can simulate real world scenes. We want images on computers to not just look more realistic, but also to be more realistic in their colors, the way objects and rooms are lighted, and the way different materials appear. We call this work “realistic image synthesis”.

Application of Computer Graphics:

- Computer-aided design for engineering and architectural systems etc.
- Presentation graphics
- Medical applications
- Office automation and desktop publishing
- Computer art and entertainment
- Education and training
- Visualization and image processing
- Graphical user interface
- Education and training
- Cartography

There are two techniques for producing images on the CRT screen:

1. Random scan / Vector scan display
2. Raster scan display

Raster Scan Display:

This is the most common method of drawing images on the CRT screen. In this method horizontal and vertical deflection signals are generated to move the beam all over the screen in a pattern as shown in figure i.e. the electron beam is swept across the screen one row at a time from top to bottom.
Here the beam is swept back and forth from left to right across the screen. When the beam is moved from left to right it is ON. The beam is OFF when it moves from right to left and is called as Horizontal Retrace and is shown by dotted lines.
When the beam reaches the bottom of the screen it is turned off and is rapidly retraced back to the top to start again. This is called the Vertical Retrace. In the Raster Scan display the screen image is maintained by repeating scanning of the same image. The process is shown as Refreshing of screen.

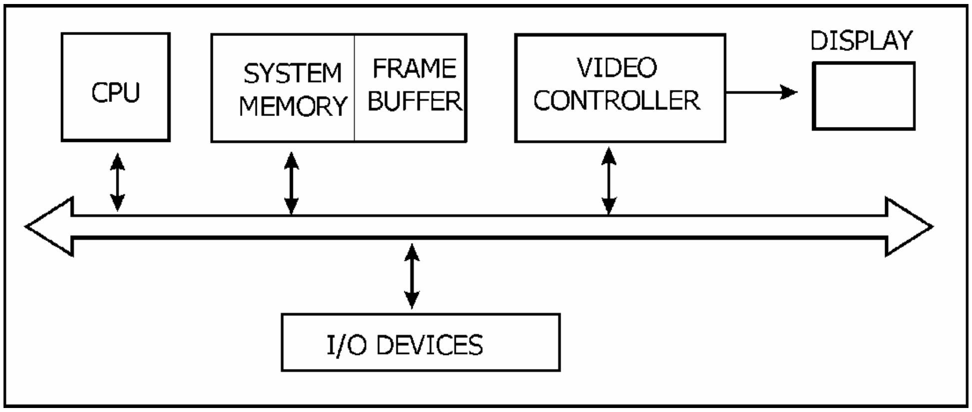


Figure 1.1 Internal operations of raster scan displays

Picture definition is stored in a memory area called Refresh buffer or Frame Buffer for which a fined area of the system memory is kept reserved. The frame buffer holds the set of intensity values for all the screen points. Stored intensity values are then retrieved from the refresh buffer and displayed on the screen one row at a time. Video controller has direct access to memory locations in the frame buffer. It is responsible for retrieving data from the frame buffer and passing it to the display device.

Frame buffer maps the screen into Cartesian co-ordinates. Generally, the screen co- ordinates are taken as positive (x, y) plane. Hence the screen is continuously refreshed by scanning from maximum value of y-coordinate down to y = 0.

Some basic definitions:

Pixel: The pixel is the smallest addressable screen element. It is the smallest piece of the display screen which we can control. The control is achieved by setting the intensity and color of the pixel which compose the screen

Frame buffer: Picture definition is stored in a memory area called the frame buffer. This frame buffer stores the intensity values for all the screen points.

Persistence: The major difference between phosphors is their persistence. It defines how long they continue to emit light after the electron beam is removed.

Horizontal retrace: The return to the left of the screen after refreshing each scan line.

Vertical retrace: After the end of each frame the electron beam returns to the top left corner of the screen.

Resolution: The maximum number of points that can be displayed on the screen.

Aspect ratio: The ratio of vertical points to the number of points present in the horizontal line. An aspect ratio of 4/5 means that a vertical line plotted with four points has the same length as a horizontal line plotted with five points.

Aliasing: In a line drawing algorithm, we have seen that all rasterized locations do not match with the true line and we have to select the optimum raster locations to represent a straight line. This problem is severe in low resolution screen.

1. Staircase A common example of aliasing effects is the staircase of jagged appearance, we see when scan converting a primitive such as a line of a circle. We also see the stair steps of jaggiest along the border of a filled region.

2. Unequal Brightness Another side effect that is less noticeable is the unequal brightness of lines of different orientation. A slanted line appears dimmer than a horizontal of vertical line although all are presented at the same intensity level. The horizontal line is placed one unit apart, whereas those on the diagonal line are approximately 1.414 units apart. This difference in density produces the perceived difference in brightness.

3. Picket Fence Problem The picket fence problem occurs when an object is not aligned with of does not fit into the pixel grid properly. In the picket fence the distance between two adjacent pickets is not a multiple of the unit distance between pixels. Scan converting it normally into the image space will result in uneven distances between pickets since the endpoints will have to be snapped to pixel coordinates.

Flickering: When the resolution is decaying by the 1/10th of its original resolution the phenomenon whereby a display screen appears to flicker. Screen flicker results from a variety of factors, the most important of which is the monitor's refresh rate, the speed with which the screen is redrawn. If the refresh rate is too slow, the screen will appear to glimmer. Another factor that affects screen flicker is the persistence of the screen phosphors. Low-persistence phosphors fade more quickly than high-persistence monitors, making screen flicker more likely. Screen flicker can also be affected by lighting. Finally, screen flicker is a subjective perception that affects people differently. Some people perceive screen flicker where others do not.

Vector & Character Generation:

1. Stroke method

a. This method uses small line segments to generate a character.
b. The small series of line segments are drawn like a stroke of a pen to form a character as shown in figure.
c. We can build our own stroke method.
   1. By calling a line drawing algorithm.
   2. Here it is necessary to decide which line segments are needed for each character and
   3. Then drawing these segments using line drawing algo.
d. This method supports scaling of the character.
e. It does this by changing the length of the line segments used for character drawing.

Figure 1.2 Character generations by stroke method

2. Starbust method

a. In this method, a fix pattern of line segments is used to generate characters.
b. As shown in figure, there are 24 line segments.
c. Out of 24 line segments, segments required to display for particular character, are highlighted
d. This method is called starbust method because of its characteristic appearance.
e. Fig shows the starbust patterns for character A and M.
f. The patterns for particular characters are stored in the form of 24 bits' code.
g. Each bit representing one line segment.
h. The bit is set to one to highlight the line segment otherwise it is set to zero.

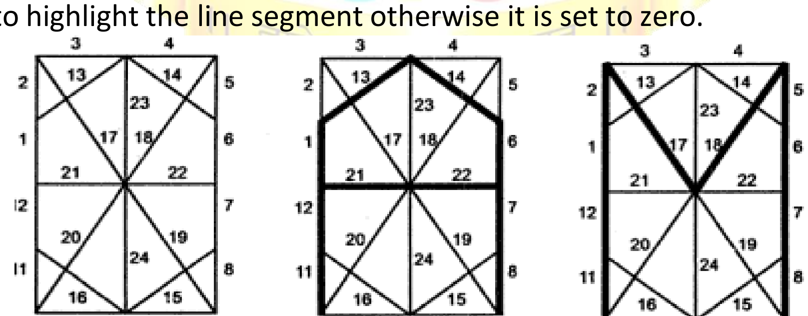


Figure 1.3 Character generations by starbust method

This method of character generation is not used now a day because of following disadvantages:

1. The 24-bits are required to represent a character. Hence more memory is required.
2. Requires code conversion software to display character from its 24 bits' code.
3. Character quality poor. Worst for curve shaped character.

3. Bitmap Method

a. The third method for character generation.
b. Also, known as dot matrix because in this method characters are represented by an array of dots in the matrix form.
c. It's a two-dimensional array having columns and rows: 5 X 7 as shown in figure.
d. Each dot in the matrix is a pixel.
e. The character is placed on the screen by copying pixel values from the character array.
f. Value of the pixel controls the intensity of the pixel.
g. Usually the dot patterns for all characters are stored in the hardware device.
h. This chip accepts address for the character and gives the bit pattern.
i. Here the size of the pixel is fixed and hence the size of the dot.
j. Disadvantage:
k. Antialiasing is possible in this method.

Random scan display directly traces out only the desired lines on the CRT tube i.e. a CRT has the electron beam directed only to the parts of the screen where a picture is to be drawn. Random scan monitors draws a picture one line at a time and for this reason is also known as random scan displays.
If we want a line connecting point A with B on the vector graphics display, we simply drive the beam deflection circuiting, which will cause the beam to go directly from point A to B. If we want to move the beam from point A to point B without showing a line between points, we can blank the beam as we move it.

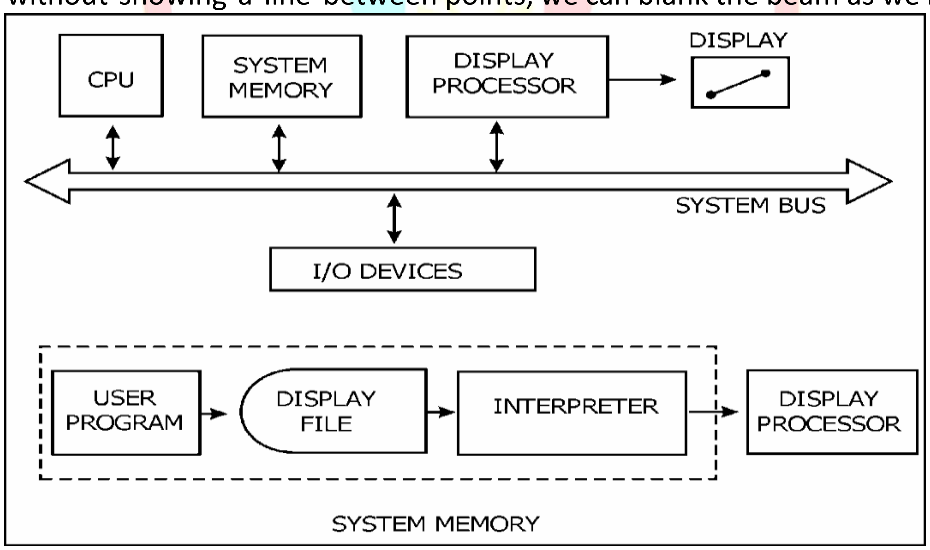


Figure 1.4 Internal operations of random scan displays

An application program along with graphics package is present in the system memory. The graphic commands are the program and translated by the graphics package and stored in the display file. In other records the file used to store the commands necessary for drawing the line segments is called the Display file. The program which converts these commands into actual picture is called the Display File Interpreter. It is an interface between the graphics representation in the display file and display device. The display processor is connected to the display device, processes and manages the display process.

The commands present in the display file contain two fields, an operation code (opcode) and operands. Opcode identifies the commands such as line draw, move cursors, etc and the operands provide the co-ordinate of a point to process the commands.
One of the ways to store opcode and operands of series of commands is to use to separate arrays, one for opcode, and one for x-coordinate and one for y-co-ordinate of the operand. It is also necessary to assign meanings to the possible opcodes before we can proceed to interpret them.
The figure below shows how these commands are interpreted and plotted.

Display Devices:

Some common type of display devices are:
1. Random scan displays.
2. Raster scan displays.
3. Direct view storage tube (DVST).
4. Flat panel displays.

The operation of most of the video monitors is based on CRT design.

Cathode Ray Tube (CRT):

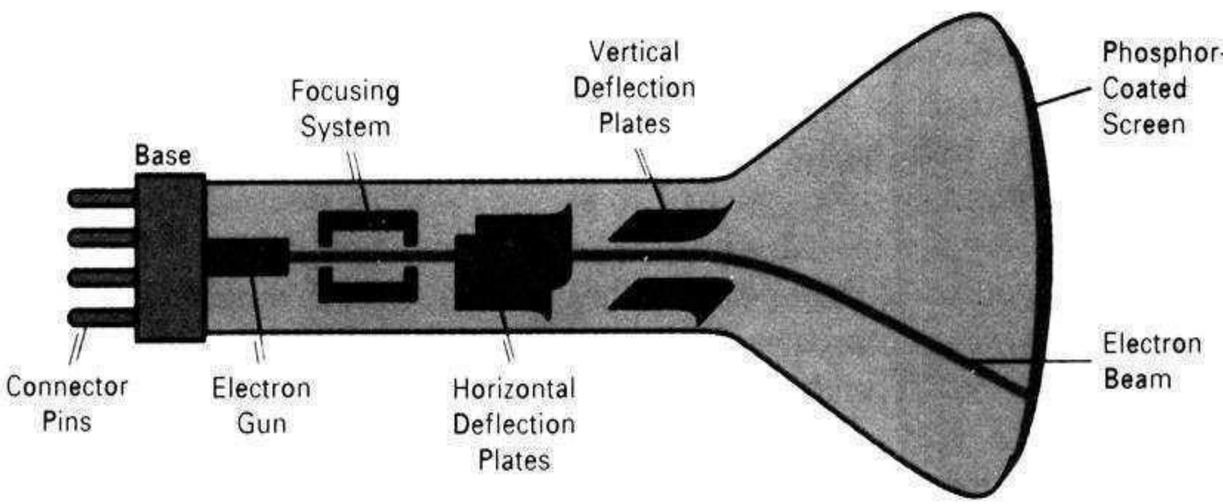


Figure 1.5 Electrostatic deflection of the electron beam in a CRT

Basic Operation:

A beam of electrons (i.e. cathode rays) is emitted by the electron gun, it passes through focusing and deflection systems that direct the beam towards the specified position on the phosphor screen. The phosphor then emits a small spot of light at every position contacted by the electron beam. Since light emitted by the phosphor fades very quickly some method is needed for maintaining the screen picture. One of the simplest ways to maintain pictures on the screen is to redraw the image rapidly. This type of display is called Refresh CRT.

Primary components of CRT are as follows: -

1. Heating element and cathode

Heat is supplied to the cathode by passing current through heater element. Cathode is cylindrical metallic structure which is rich in electrons. On heating the electrons are released from cathode surface.

2. Control grids

It is the next element which follows cathode. It almost covers cathode leaving small opening for electrons to come out. Intensity of the electron beam is controlled by setting voltage levels on the control grid. A high negative cottage applied to the control grid will shut off the beam by repelling electrons and stopping them from passing through the small hole at the end of control grid structure. A smaller negative voltage on the control grid will simply decrease the number of electrons passing through. Thus, we can control the brightness of a display by varying the voltage on the control grid.

3. Accelerating anode

They are positively charged anodes who accelerate the electrons towards phosphor screen.

4. Focusing & deflection coils

They are together needed to force the electron beam to converge into a small spot as it strikes the screen otherwise the electrons would repel each other and the beam would spread out as it approaches the screen. Electrostatic focusing is commonly used in television and computer graphics monitor.

5. Phosphor Coating

When the accelerating electron beam (collides) is incident on the phosphor cooling, a part of Kinetic Energy is converted into light and heat. When the electrons in the beam collide with the phosphor coating they are stopped and their kinetic energy is absorbed by the phosphor.

Direct View Storage Tubes (DVST):

a. It gives the alternative method of maintaining the screen image. A DVST uses the storage grid which stores the picture information as a charge distribution just behind the phosphor-coated screen.
b. It consists of two electron guns: a primary gun and a flood gun.
c. A primary gun stores the picture pattern and the flood gun maintains the picture display.
d. A primary gun produces high speed electrons which strike on the storage grid to draw the picture pattern.

Advantage:

a. Refreshing of CRT is not required.
b. Very complex pictures can be displayed at a very high resolution without flicker.

Disadvantage:

a. They do not display colors.
b. Selective or part erasing of screen is not possible.

Distinguish between Raster & Random Scan Displays:

RASTER SCAN DISPLAY	RANDOM SCAN DISPLAY
a) It draws the image by scanning one row at a time	a) It draws the image by directing the electron beam directly to the part of the screen where the image is to be drawn.
b) Refresh rate is independent of picture complexity.	b) Refresh rate directly depends on picture complexity.
c) It produces jagged lines that are plotted as discrete point sets.	c) Smooth lines are displayed because CRT beam directly follows line path.
d) They are more suited to geometric area drawing applications e.g. Monitors, Tele vision	d) They are more suited to line drawing application e) g. CRO, pen plotter.

e) Raster scan displays have less resolution.	e) Random scan displays have high resolution since picture definition is stored as a set of line drawing commands not as a set of intensity values.
f) Editing is easy.	f) Editing is difficult.

Colour Displays:

There are 2 methods for producing color displays.

A. Beam Penetration technique
B. Shadow Mask technique

Beam penetration technique:

This technique is used in Random Scan Monitors. In this technique, the inside of CRT is coated with two layers of phosphor, usually red & green. The displayed color depends on how far the electron beam penetrates into the phosphor layers.

The outer layer is of red phosphor and inner layer is of green phosphor. A beam of slow electrons excites only the outer red layer. A beam of fast electrons penetrates the outer red layer and excites the inner green layer. At intermediate beam speeds, combinations of red and green light are emitted and two additional colors orange and yellow are displayed. The beam acceleration voltage controls the speed of the electrons and hence the screen color at any point on the screen.

Shadow mask technique:

The shadow mask technique produces a much wider range of colors than the beam penetrationtechnique. Hence this technique is commonly used in raster scan displays including color T.V. In shadow mask technique, the CRT screen has three phosphor color dots at each pixel position. One phosphor dot emits red light, another emits a green light and the third one emits a blue light. The CRT has three electron guns one for each dot, a shadow mask grid just behind the phosphor coated screen.

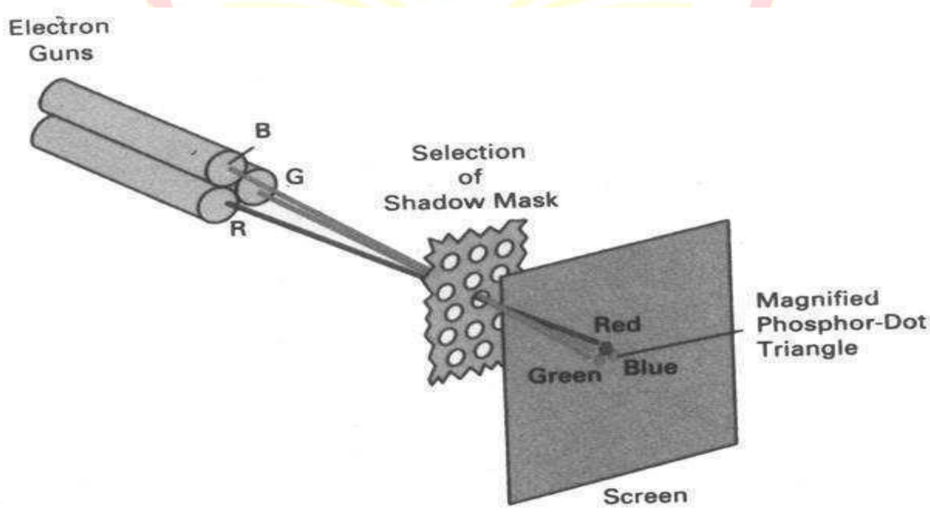


Figure 1.6 Operations of delta-delta shadow mask method

The shadow mask grid consists of series of holes aligned with the phosphor dot patterns. As shown in figure, the three electron beams are deflected and focused as a group onto the shadow mask and when they pass through a hole onto a shadow mask they excite a dot triangle. A dot triangle consists of 3 small phosphor dots of red, green and blue colour. These phosphor dots are arranged so that each electron beam can activate only its corresponding colour dot when it passes through the shadow mask. A dot triangle when activated appears as a small dot on the screen which has colour combination of three small dots in the dot triangle. By varying the intensity of the three electron beams we can obtain different colours in the shadow mask CRT.

Input Devices:

Following are the commonly used input devices

1. Keyboard
2. Mouse
3. Scanner
4. Trackball / Space ball
5. Joystick
6. Digitizer / Graphical tablet

1. Keyboard:

Keyboard is the primary input device for any graphics system. It is used for entering text and numbers. Keyboards are available in various sizes, shapes & styles. The standard keyboard consists of

a. Alphanumeric keys
b. Function keys
c. Modifier keys
d. Cursor Movement keys
e. Numeric Keypad

When we press a key on the keyboard, the keyboard controller places a code corresponding to the key pressed, in a part of its memory called keyboard buffer. This code is called the scan code. The keyboard controller informs the CPU of the computer about the key pressed with the help of interrupt signals. The CPU then reads the scan code from the Keyboard Buffer.

2. Mouse:

A mouse is a palm sized box used to position the screen cursor. It consists of a ball on the bottom connected to wheels to provide the amount and direction of movement. One, two or three buttons are usually included on the top of the mouse for signaling the execution of same operation. Now-a-days mouse consists of one more wheel on top to scroll the screen pages.

3. Track Ball & Spaceball:

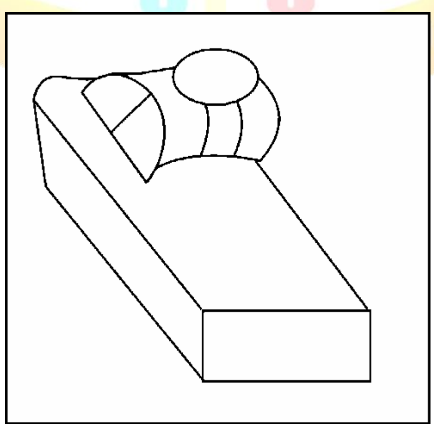


Figure 1.7 Operations of track ball

A trackball is a ball that can be rotated with the finger or palm of the hand to produce the screen cursor movement. Potentiometers attached to the ball measure the amount and direction of rotation.

While the trackball is used for 2D positioning, the space ball is used for 3D positioning & selection of operation in virtual reality system unlike the trackball the space ball does not actually move. It consists of strain gauges which measures the amount of pressure applied to the space ball to provide input for spatial positioning & orientation of the ball is pushed a pulled in various directions.

4. Scanners:

Scanner has become an important part of the home, office over the last few years. It is a device which can be used to store drawings, graphics, and photos as text available in the printed form for computer processing.

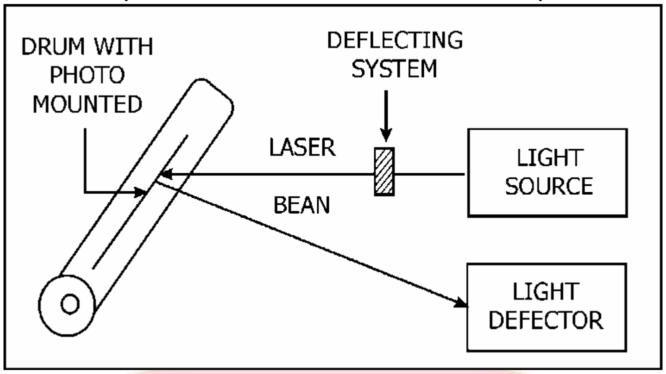


Figure 1.8 Working of Scanners

As shown in the figure the photograph is mounted on a rotating drum. A finely collimated light beam is directed at the photo, and the amount of light reflected is measured by a photocell. As the drum rotates the light source slowly moves from one end to the other, thus doing a raster scan of the inter photograph.

5. Joysticks:

A joystick has a small vertical lever mounted on the base and used to steer the screen cursor around. It consists of two potentiometers attached to a single liver. Moving the liver changes the settings on the potentiometer. The left or right movement is indicated by other one potentiometer & the forward orbackward movement is indicated by another potentiometer. Thus, with a position can be simultaneously altered by the motion of a single lever.

Some joysticks may return to this zero (centre) positions when released. Joysticks are inexpressive and quiet commonly used where only rough positioning is needed.

6. Digitizer or Graphical Tablet:

For applications such as tracing we need a device called a Digitizer or Graphical Tablet. It consists of a flat surface, ranging in size from about 6 x 6 inches up to 48 to 72 inches or more, which can direct the position of a movable style.

In this arrangement, a grid of wires is embedded in the tablet surface. Electromagnetic signals generated by electrical pulses applied in sequence to the wires in the grid induce an electrical signal in a wire cod in the styles. The strength of the signal induced by each pulse is used to determine the position of the styles.

Cartesian coordinate system:

A coordinate system provides a framework for translating geometric ideas into numerical expressions.

In a two-dimensional plane, we pick any point and single it out as a reference point called the origin. Through the origin, we construct two perpendicular number lines called axes. These are labeled the X axis and the Y axis. Any point in two dimensions in this X-Y plane can be specified by a pair of numbers, the first number is for the X axis, and the second number is for the Y axis.

Interactive graphics display working:

The modern graphics display is extremely simple in construction. It consists of three components:

- A digital memory, or frame buffer, in which the image is stored as a matrix of intensity values.
- A monitor
- A display controller, which is a simple interface that passes the contents of the frame buffer to the monitor.

Inside the frame buffer the image is stored as a pattern of binary digital numbers, which represent a rectangular array of picture elements, or pixel. The pixel is the smallest addressable screen element. In the simplest case where we wish to store only black and white images, we can represent black pixels by 0's in the frame buffer and white pixels by 1's. The display controller simply reads each successive byte of data from the frame buffer and converts each 0 and 1 to the corresponding video signal. This signal is then fed to the monitor. If we wish to change the displayed picture all we need to do is to change of modify the frame buffer contents to represent the

Scan conversion techniques:

The process of determining which pixels provide best approximation to the desired line is known as rasterization. When combined with the process of generating the picture in scan line order it is known as scan conversion.

Characteristics of a line:

- The line should be appearing as a straight line and it should start and end accurately.
- The line should be displayed with constant brightness along its length independent of its length and orientation.
- The line should be drawn rapidly.

The basic concept behind all drawing algorithms is to reduce the computations and provide the result rapidly, so most of the line drawing algorithms use incremental methods. In these methods line starts with starting point and then a fix increment is added to get the next point on the line. This is continued till the end of line.

DDA (Digital Differential Analyzer) Algorithm:

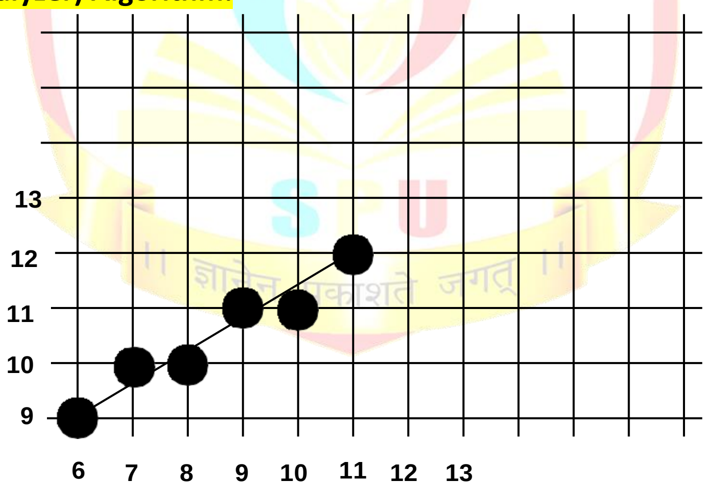


Figure 1.9 Representation of line with DDA algorithm

Procedure DDA (X1, Y1, X2, Y2: Integer);

Var Length, I: Integer;

X,Y,Xinc,Yinc:Real;

Begin

Length: = ABS(X2 - X1);

If ABS (Y2 - Y1) > Length Then

Length: = ABS (Y2-Y1);

Xinc: = (X2 - X1)/Length;

Yinc: = (Y2 - Y1)/Length;

```
        X: = X1+0.5*SIGN(Xinc);
        Y: = Y1+0.5*SIGN(Yinc);
For I: = 0 To Length Do
        Begin
                Plot (Round(X), Round(Y));
                X: = X + Xinc;
                Y: = Y + Yinc
        End {For}
End; {DDA}
```

DDA (*digital differential analyzer*) creates good lines but it is too time consuming due to the round function and long operations on real values.

Key points:

- DDA works with floating point arithmetic
- Rounding to integers necessary
- It takes lot of computation time because at each and every step it has to round off.

Bresenham's Line Algorithm:

A) Bresenham's line algorithm for m<1

F0R (X1, Y1) =1, 2 AND (X2, Y2) =8, 5

K	YK+1
0	2
1	3
2	3
3	4
4	4
5	5
6	5

input line endpoints, (x_0,y_0) and (x_n, y_n)
calculate $\Delta x = x_n - x_0$ and $\Delta y = y_n - y_0$
calculate parameter $p_0 = 2\,\Delta y - \Delta x$
set pixel at position (x_0,y_0)
repeat the following steps until (x_n, y_n) is reached:
if $p_i < 0$
set the next pixel at position $(x_i +1, y_i)$
calculate new $p_{i+1} = p_i + 2\,\Delta y$
if $p_i \geq 0$
set the next pixel at position $(x_i +1, y_i + 1)$
calculate new $p_{i+1} = p_i + 2\Delta y - 2\Delta x$

Repeat steps until dx-1 times.

B) Bresenham's line algorithm for m>1

input line endpoints, (x_0,y_0) and (x_n, y_n)

calculate $\Delta x = x_n - x_0$ and $\Delta y = y_n - y_0$

calculate parameter $p_0 = 2\,\Delta x - \Delta y$

set pixel at position (x_0,y_0)

repeat the following steps until (x_n, y_n) is reached:if $p_i < 0$

set the next pixel at position $(x_i , y_i +1)$

calculate new $p_{i+1} = p_i + 2\,\Delta x$

if $p_i \geq 0$

set the next pixel at position $(x_i +1, y_i + 1)$

calculate new $p_{i+1} = p_i + 2\Delta x - 2\Delta y$

Repeat steps until dy-1 times.

F0R (X1, Y1) =1, 3 AND (X2, Y2) =8, 12

K	PK	XK+1	YK+1
0	5	2	4
1	1	3	5
2	-3	3	6
3	11	4	7
4	7	5	8
5	3	6	9
6	-1	6	10
7	13	7	11
8	9	8	12

Key points

- Bresenham's algorithm uses integer arithmetic
- Constants need to be computed only once
- Bresenham's algorithm generally faster than DDA

Midpoint Circle Algorithm:

1. Input radius r
2. Plot a point at (0, r)
3. Calculate the initial value of the decision parameter as $p_0 = 5/4 - r \approx 1 - r$
4. At each position x_k, starting at k = 0, perform the following test:

if $p_k < 0$

plot point at (x_{k+1}, y_k)

compute new $p_{k+1} = p_k + 2x_{k+1} + 1$

else

plot point at $(x_{k+1}, y_k - 1)$

compute new $p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$

5. Determine symmetry points in the other seven octants and plot points
6. Repeat steps 4 and 5 until $x \geq y$.

For radius =8

So your first point is (0, r) = (0, 8)

K	PK	XK+1	YK+1
0	-7	1	8
1	-4	2	8

2	1	3	7
3	-6	4	7
4	3	5	6
5	2	6	5

Bresenham's Circle Algorithm:

$x_0 = 0$

$y_0 = r$

$p_0 = 3 - 2r$

if $p_i < 0$ *then*

$y_{i+1} = y_i$

$p_{i+1} = p_i + 4x_i + 6$

else if $p_i \geq 0$ *then*

$y_{i+1} = y_i - 1$

$p_{i+1} = p_i + 4(x_i - y_i) + 10$

Stop when $x_i \geq y_i$ and determine symmetry points in the other octants

For radius=6

So your first point is (0, r) = (0, 6)

K	PI	XK+1	YK+1
0	-9	1	6
1	1	2	5
2	-1	3	5
3	17	4	4

Midpoint circle algorithm

Let us suppose centre is other than origin here xc, yc is (5,5)

For radius =8

So your first point is (0,r)=(0,8)

K	PK	XK+1	YK+1	PLOTTED POINTS
0	-7	1	8	X=xc+x y=yc+y X=5+1=6,y=5+8=13
1	-4	2	8	X=xc+x y=yc+y X=5+2=7,y=5+8=13
2	1	3	7	X=xc+x y=yc+y X=5+3=8,y=5+7=12
3	-6	4	7	X=xc+x y=yc+y X=5+4=9,y=5+7=12
4	3	5	6	X=xc+x y=yc+y X=5+5=10,y=5+6=11
5	2	6	5	X=xc+x y=yc+y X=5+6=11,y=5+5=10

Remaining points you can calculate

XC+X, YC-Y

XC-X, YC+Y

XC-X, YC-Y

XC+Y, YC+X

XC+Y, YC-X

XC-Y, YC+X

,XC-Y, YC-X

Bresenham's Circle Algorithm:

For radius=6 suppose center is(xc,yc)=3

K	PI	XK+1	YK+1	PLOTTED POINTS
0	-9	1	6	X=xc+x y=yc+y X=3+1=4,y=3+6=9
1	1	2	5	X=xc+x y=yc+y X=3+2=5,y=3+5=8
2	-1	3	5	X=xc+x y=yc+y X=3+3=6,y=3+5=8
3	17	4	4	X=xc+x y=yc+y X=3+4=7,y=3+4=7

Boundary Fill Algorithm:

A boundary-fill procedure accepts as input the coordinates of an interior point (x, y), a fill color, and a boundary color. Starting from (x, y), the procedure tests neighboring positions to determine whether they are of the boundary color. If not, they are painted with the fill color, and their neighbors are tested. This process continues until all pixels up to the boundary color for the area have been tested.

```
void BoundaryFill4(int x, int y, int fill, int boundary)
{
int current;
current=GetPixel(x,y);
        if (current!=boundary) and (current!=fill)
        {
         SetPixel(x,y,fill);
        BoundaryFill4(x+1,y,fill,boundary);
        BoundaryFill4(x-1,y,fill,boundary);
        BoundaryFill4(x,y+1,fill,boundary);
        BoundaryFill4(x,y-1,fill,boundary);
        }
}
```

Flood Fill Algorithm:

Sometimes we want to fill in (or recolor) an area that is not defined within a single-color boundary. It has area bordered by several different color regions. We can paint such areas by replacing a specified interior color instead of searching for a boundary color value. This approach is called a flood-fill algorithm. We start from a specified interior point (x, y) and reassign all pixel values that are currently set to a given interior color with the desired fill color. If the area we want to paint has more than one interior color, we can first reassign pixel values so that all interior points have the same color.

```
void FloodFill4(int x, int y, int fill, int old_color)
{
        if (GetPixel(x,y)== old_color)
        {
        SetPixel(x,y,fill);
        FloodFill4(x+1,y,fill,boundary);
        FloodFill4(x-1,y,fill,boundary);
        FloodFill4(x,y+1,fill,boundary);
        FloodFill4(x,y-1,fill,boundary);
        }}
```

Unit-II

2-D Transformation

In many applications, changes in orientations, size, and shape are accomplished with geometric transformations that alter the coordinate descriptions of objects.

Basic geometric transformations are:

- Translation
- Rotation
- Scaling

Other transformations:

- Reflection
- Shear

Translation:

We translate a 2D point by adding translation distances, tx and ty, to the original coordinate position (x,y):

$$x' = x + tx$$
$$y' = y + ty$$

Alternatively, translation can also be specified by the following transformation matrix:

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$

Then we can rewrite the formula as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scaling:

We scale a 2D point by multiplying scaling factor, Sx and Sy, to the original coordinate position (x,y):

$$x' = xSx$$
$$y' = ySy$$

Alternatively, scaling can also be specified by the following transformation matrix:

$$\begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then we can rewrite the formula as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scaling is performed about the origin (0,0) not about the centre of the line or polygon.

- Scale > 1 enlarge the object and move it away from the origin
- Scale = 1 leave the object alone
- Scale< 1 shrink the object and move it towards the origin.

Uniform scaling: Sx = Sy

Differential scaling Sx != Sy -> alters proportions

Rotation:

To rotate an object about the origin (0,0), we specify the rotation angle. Positive and negative values for the rotation angle define counter clockwise and clockwise rotations respectively. The following is the computation of this rotation for a point:

$$x' = x \cos - y \sin$$
$$y' = x \sin + y \cos$$

Alternatively, this rotation can also be specified by the following transformation matrix:

$$\begin{bmatrix} \cos & -\sin & 0 \\ \sin & \cos & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

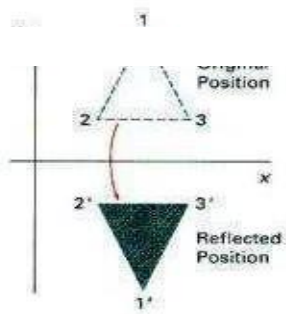
Then we can rewrite the formula as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos & -\sin & 0 \\ \sin & \cos & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Reflection:

Reflection about the x axis:

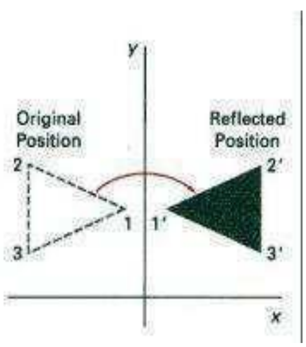
$$x' = x$$
$$y' = -y$$



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Figure 2.1 Reflection about the x axis

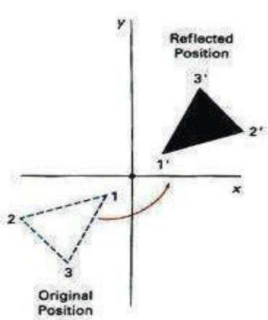
Reflection about the y axis:



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Figure 2.2 Reflection about the y axis

Flipping both x and y coordinates of a point relative to the origin:

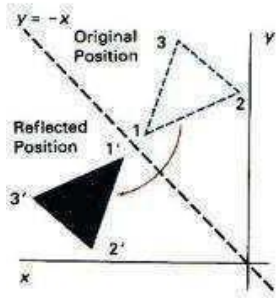


$$x' = -x$$
$$y' = -y$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Figure 2.3 Reflection about the origin

Reflection about the diagonal line y=x:

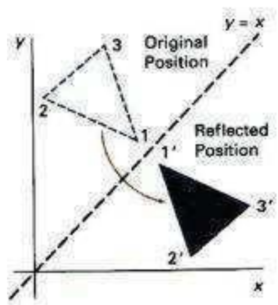


$$x' = y$$
$$y' = x$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Figure 2.4 Reflection about the line y=x

Reflection about the diagonal line y=-x:

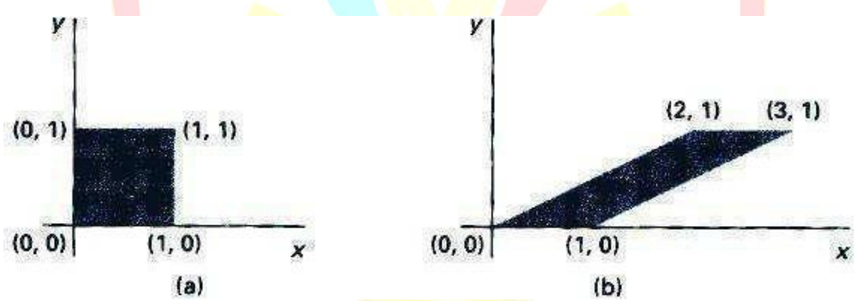


$$x' = -y$$
$$y' = -x$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Figure 2.6 Shearing about the x axis

$$x' = x + y.Shx$$



$$y' = x$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Shx & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Similarly we can find a y-direction shear, with a shearing parameter shy, relative to the y-axis.

Inverse Geometric Transformation

Each geometric transformation has an inverse, which is described by the opposite operation performed by the original transformation. Any transformation followed by its inverse transformation keeps an object unchanged in position, orientation, size and shape. We will use inverse transformation to nullify the effects of already applied transformation.

Homogeneous Coordinates in 2 Dimensions

Scaling and rotations are both handled using matrix multiplication, which can be combined as we will see above. The translations cause a difficulty, however, since they use addition instead of multiplication.

We want to be able to treat all 3 transformations (translation, scaling, and rotation) in the same way - as multiplications.

The solution is to give each point a third coordinate (X, Y, W), which will allow translations to be handled as a multiplication also.

(Note that we are not really moving into the third dimension yet. The third coordinate is being added to the mathematics solely in order to combine the addition and multiplication of 2-D coordinates.)

Two triples (X,Y,W) and (X',Y',W') represent the same point if they are multiples of each other e.g. (1,2,3) and (2,4,6).

At least one of the three coordinates must be nonzero.

If W is 0 then the point is at infinity. This situation will rarely occur in practice in computer graphics.

If W is nonzero we can divide the triple by W to get the Cartesian coordinates of X and Y which will be identical for triples representing the same point (X/W, Y/W, 1). This step can be considered as mapping the point from 3-D space onto the plane W=1.

Conversely, if the 2-D Cartesian coordinates of a point are known as (X, Y), then the homogenous coordinates can be given as (X, Y, 1).

Homogeneous co-ordinates for Translation

The homogeneous co-ordinates for translation are given as

$$T = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$

Homogeneous co-ordinates for Scaling

The homogeneous co-ordinates for Scaling are given as

$$S = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Homogeneous co-ordinates for Rotation

The homogeneous co-ordinates for Rotation are given as

$$R = \begin{bmatrix} \cos & -\sin & 0 \\ \sin & \cos & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composite Transformation

There are many situations in which the final transformation of a point is a combination of several (often many) individual transformations.

Translations

By common sense, if we translate a shape with 2 successive translation vectors: (tx1, ty1) and (tx2, ty2), it is equal to a single translation of (tx1+ tx2, ty1+ ty2).

This additive property can be demonstrated by composite transformation matrix:

$$\begin{bmatrix} 1 & 0 & tx1 \\ 0 & 1 & ty1 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & tx2 \\ 0 & 1 & ty2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx1+tx2 \\ 0 & 1 & ty1+ty2 \\ 0 & 0 & 1 \end{bmatrix}$$

This demonstrates that 2 successive translations are additive.

Scaling

By common sense, if we scale a shape with 2 successive scaling factor: (sx1, sy1) and (sx2, sy2), with respect to the origin, it is equal to a single scaling of (sx1* sx2, sy1* sy2) with respect to the origin. This multiplicative property can be demonstrated by composite transformation matrix:

$$\begin{bmatrix} Sx1 & 0 & 0 \\ 0 & Sy1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} Sx2 & 0 & 0 \\ 0 & Sy2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} Sx1.Sx2 & 0 & 0 \\ 0 & Sy1.Sy2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This demonstrates that 2 successive scaling with respect to the origin are multiplicative.

Rotations

By common sense, if we rotate a shape with 2 successive rotation angles: θ1 and θ2, about the origin, it is equal to rotating the shape once by an angle θ1+ θ2 about the origin.

Similarly, this additive property can be demonstrated by composite transformation matrix:

$$\begin{bmatrix} \cos 1 & -\sin 1 & 0 \\ \sin 1 & \cos 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos 2 & -\sin 2 & 0 \\ \sin 2 & \cos 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(1+2) & -\sin(1+2) & 0 \\ \sin(1+2) & \cos(1+2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This demonstrates that 2 successive rotations are additive.

Windowing & Clipping

The world coordinate system is used to define the position of objects in the natural world. This system does not depend on the screen coordinate system, so the interval of number can be anything (positive, negative or decimal). Sometimes the complete picture of object in the world coordinate system is too large and complicate to clearly show on the screen, and we need to show only some part of the object. The capability that show some part of object internal a specify window is called windowing and a rectangular region in a world coordinate system is called window. Before going into clipping, you should understand the differences between window and a viewport.

A *Window* is a rectangular region in the *world coordinate system*. This is the coordinate system used to locate an object in the natural world. The world coordinate system does not depend on a display device, so the units of measure can be positive, negative or decimal numbers.

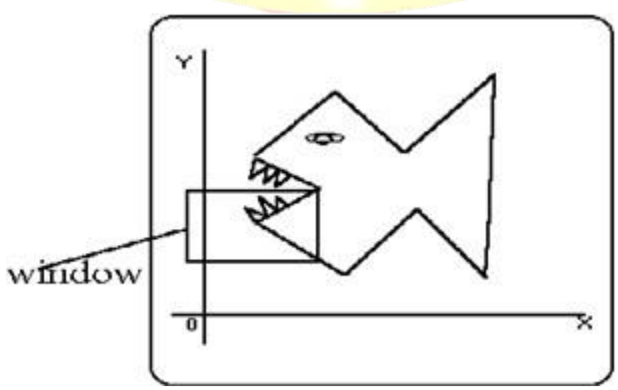


Figure 2.7 Graphical Object in World Coordinate System

A *Viewport* is the section of the screen where the images encompassed by the window on the world coordinate system will be drawn. A coordinate transformation is required to display the image, encompassed by the window, in the viewport. The viewport uses the *screen coordinate system* so this transformation is from the world coordinate system to the screen coordinate system.

When a window is "placed" on the world, only certain objects and parts of objects can be seen. Points and lines which are outside the window are "cut off" from view. This process of "cutting off" parts of the image of the world is called *Clipping*. In clipping, we examine each line to determine whether or not it is completely inside the window, completely outside the window, or crosses a window boundary. If inside the window, the line is displayed. If outside the window, the lines and points are not displayed. If a line crosses the boundary, we must determine the point of intersection and display only the part which lies inside the window.

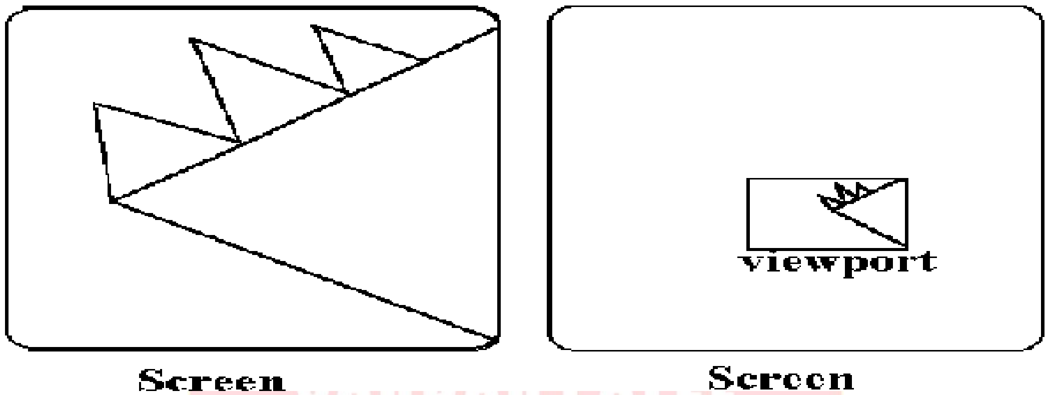


Figure 2.8 Picture in Viewing Coordinate System

Viewing Transformation

An image representing a view often becomes part of a larger image, like a photo on an album page, which models a computer monitor's display area. Since album pages vary and monitor sizes differ from one system to another, we want to introduce a device-independent tool to describe the display area. This tool is called the normalized device coordinate system (NDCS) in which a unit (1 x 1) square whose lower left corner is at the origin of the coordinate system defines the display area of a virtual display device. A rectangular viewport with its edges parallel to the axes of the NDCS is used to specify a sub-region of the display area that embodies the image.

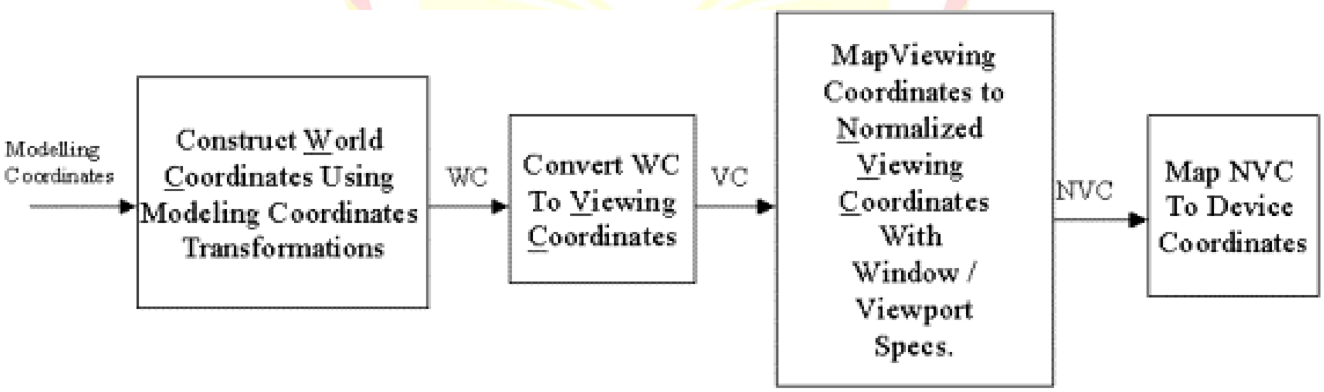


Figure 2.9 2D Viewing-Transformation Pipeline

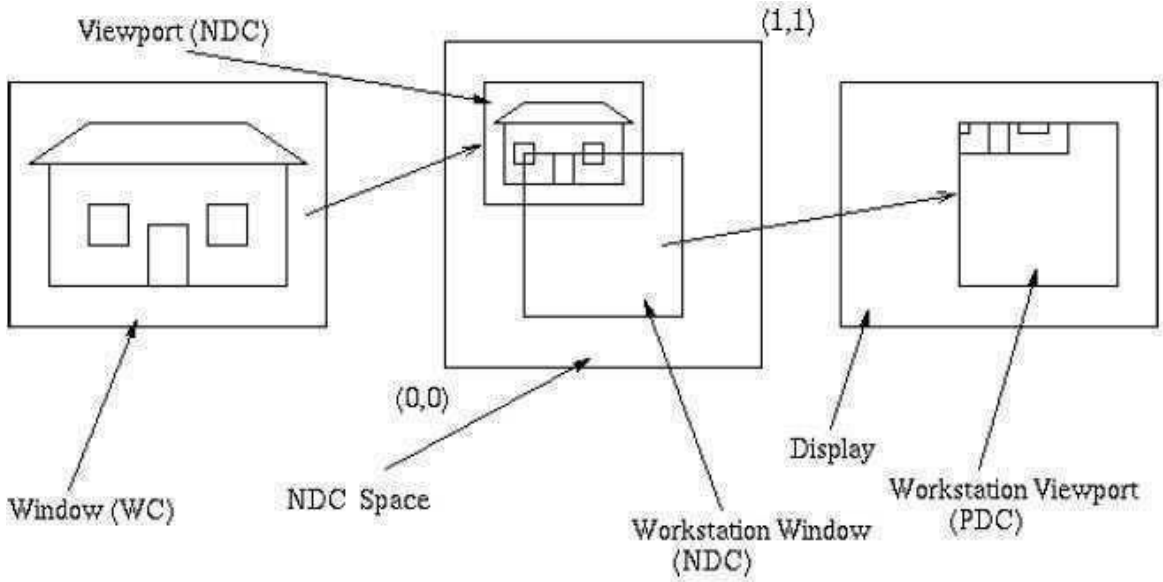


Figure 2.10 Viewing Transformation

The process that converts object coordinates in WCS to normalized device coordinate is called window–to–viewport mapping or normalization transformation. The process that maps normalized device coordinates to Physical Device Co-ordinates (PDC) / image coordinates is called work, station transformation, which is essentially a second window-to-viewport mapping, with a workstation window in the normalized device coordinate system and a workstation viewport in the device coordinate window in the normalized device coordinate system and a workstation viewport in the device coordinate system. Collectively, these two coordinate mapping operations are referred to as viewing transformation.Workstation transformation is dependent on the resolution of the display device/frame buffer. When the whole display area of the virtual device is mapped to a physical device that does not have a 1/1 aspect ratio, it may be mapped to a square sub-region so as to avoid introducing unwanted geometric distortion.

Point Clipping

Point clipping is essentially the evaluation of the following inequalities:

$$xmin \leq x \leq xmax \text{ and } ymin \leq y \leq ymax$$

Where, xmin, xmax, ymin and ymax define the clipping window. A point (x,y) is considered inside the window when the inequalities all evaluate to true.

Line Clipping

Lines that do not intersect the clipping window are either completely inside the window or completely outside the window. On the other hand, a line that intersects the clipping window is divided by the intersection points into segments that are either inside or outside the window. The following algorithms provide efficient ways to decide the spatial relationship between an arbitrary line and the clipping window and to find intersection points.

Cohen-Sutherland Line Clipping

In this algorithm we divide the line clipping process into two phases: (1) identify those lines which intersect the clipping window and so need to be clipped and (2) perform the clipping.

All lines fall into one of the following clipping categories:

1. Visible – both endpoints of the line within the window

2. Not visible – the line definitely lies outside the window. This will occur if the line from (x1,y1,) to (x2,y2) satisfies any one of the following four inequalities:

x1,x2>xmax y1,y2>ymax

x1,x2<xmin y1,y2<ymin

3. Clipping candidate – the line is in neither category 1 and 2

In fig., line l1 is in category 1 (visible); lines l2 and l3 are in category 2 (not visible); and lines l4 and l5 are in category 3 (clipping candidate).

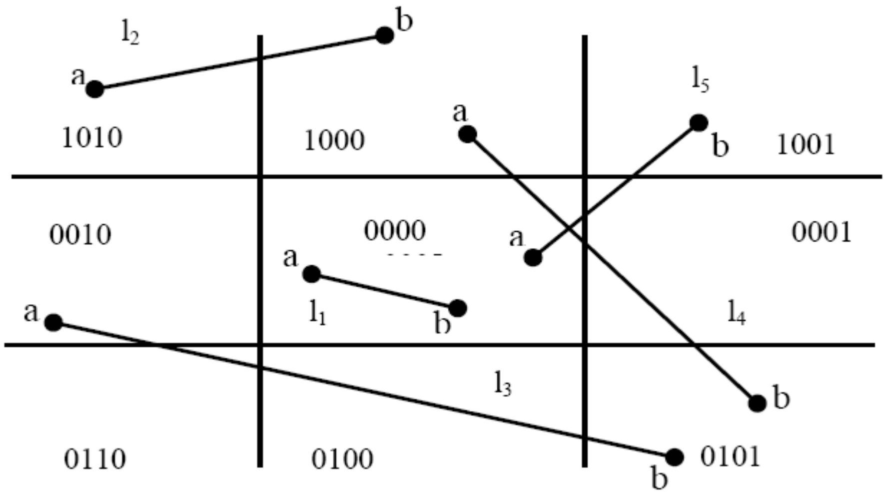


Figure 2.11 Representations of Line codes in Cohen Sutherland Algorithm

The algorithm employs an efficient procedure for finding the category of a line. It proceeds in two steps:

1. Assign a 4-bit region code to each endpoint of the line. The code is determined according to which of the following nine regions of the plane the endpoint lies in

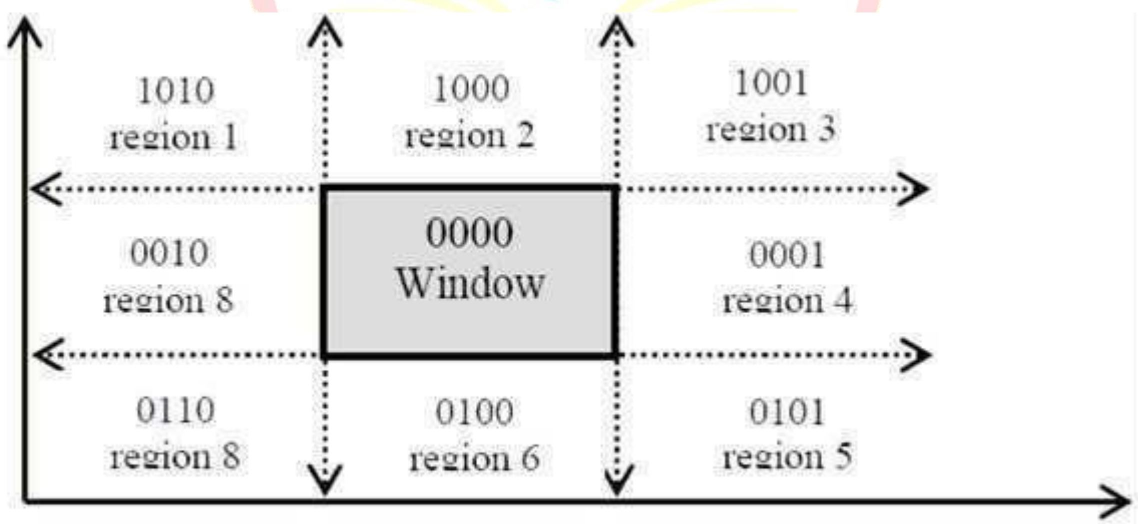


Figure 2.12 Region Code Assignments

Starting from the leftmost bit, each bit of the code is set to true (1) or false (0) according to the scheme

Bit 1: endpoint is above the window = sign (y – ymax)
Bit 2: endpoint is below the window = sign (ymin-y)
Bit 3: endpoint is to the right of the window = sign (x–xmax)
Bit 4: endpoint is to the left of the window = sign (xmin –x)

We use the convention that sign (a) =1 if a is positive, 0 otherwise. Of course, a point with code 0000 is inside the window.

2. The line is visible if both region codes are 0000, and not visible if the bitwise logical AND of the codes is not 0000, and a candidate for clipping if the bitwise logical AND of the region codes is 0000.

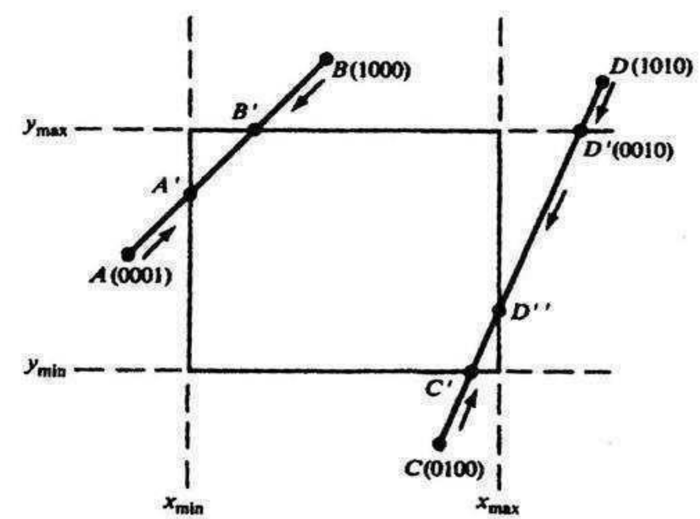


Figure 2.13 Example of Cohen Sutherland Algorithm

For a line in category 3 we proceed to find the intersection point of the line with one of the boundaries of the clipping window, or to be exact, with the infinite extension of one of the boundaries. We choose an endpoint of the line, say (x1, y1), that is outside the window, i.e., whose region code is not 0000. We then select an extended boundary line by observing that those boundary lines that are candidates for intersection are of ones for which the chosen endpoint must be "pushed across" so as to change a "1" in its code to a "0" (see Fig. above). This means:

If bit 1 is 1, intersect with line y = ymax.
If bit 2 is 1, intersect with line y = ymin.
If bit 3 is 1, intersect with line x = xmax.
If bit 4 is 1, intersect with line y = ymin.

Consider line CD is Fig. If endpoint C is chosen, then the bottom boundary line y=ymin is selected for computing intersection. On the other hand, if endpoint D is chosen, then either the top boundary line y=ymax or the right boundary line x = xmax is used. The coordinates of the intersection point are

xi=xmin or xmax // if the boundary line is vertical

yi=y1+m(xi-x1)

or

xi=x1+(yi-y1)/m // if the boundary line is horizontal

yi=ymin or ymax

Where m = (y2 – y1)/(x2 – x1) is the slope of the line.

Now we replace endpoint (x1, y1) with the intersection point (x1, yi) effectively eliminating the portion of the original line that is on the outside of the selected window boundary. The new endpoint is then assigned an updated region code and the clipped line re-categorized and handled in the same way. This iterative process terminates when we finally reach a clipped line that belongs to either category 1 (visible) or category 2 (not visible).

Liang-Barsky Line Clipping

The ideas for clipping line of Liang-Barsky and Cyrus-Beck are the same. The only difference is Liang-Barsky algorithm has been optimized for an upright rectangular clip window. Liang and Barsky have created an algorithm that uses floating-point arithmetic but finds the appropriate end points with at most four computations. This algorithm uses the parametric equations for a line and solves four inequalities to find the range of the parameter for which the line is in the viewport.

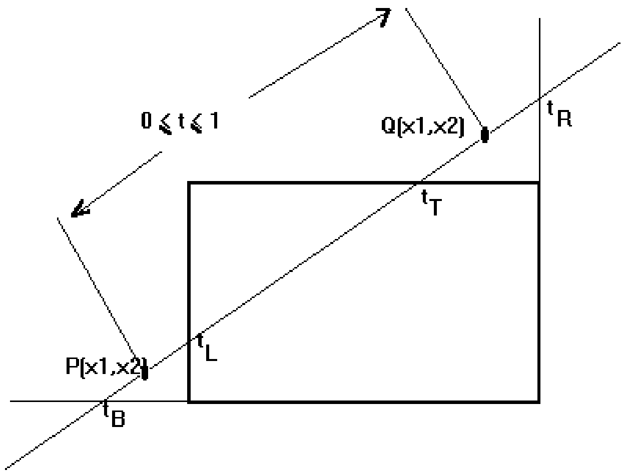


Figure 2.14 Example of Liang-Barsky Clipping

Let P(x1,y1) , Q(x2,y2)be the line which we want to study. The parametric equation of the line segment fromgives x-values and y-values for every point in terms of a parameter that ranges from 0 to 1. The equations are x=x1+(x2-x1)*t=x1+dx*t

& y=y1+(y2-y1)*t=y1+dy*t

We can see that when t = 0, the point computed is P(x1,y1); and when t = 1, the point computed is Q(x2,y2).

Algorithm

1. Set $t_{min} = 0$ and $t_{max} = 1$
2. Calculate the values of tL, tR, tT, and tB (tvalues).
   - if $t < t_{min}$ or $t > t_{max}$ ignore it and go to the next edge
   - otherwise classify the tvalue as entering or exiting value (using inner product to classify)
   - if t is entering value set $t_{min} = t$; if t is exiting value set $t_{max} = t$
3. If $t_{min} < t_{max}$. then draw a line from (x1 + dx*tmin, y1 + dy*tmin) to (x1 + dx*tmax, y1 + dy*tmax)
4. If the line crosses over the window, you will see (x1 + dx*tmin, y1 + dy*tmin) and (x1 + dx*tmax, y1 + dy*tmax) are intersection between line and edge.

Midpoint Subdivision

An alternative way to process a line in category 3 is based on binary search. The line is divided at its midpoint into two shorter line segments. The clipping categories of the two new line segments are then determined by their region codes. Each segment in category 3 is divided again into shorter segments and categorized. This bisection and categorization process continues until each line segment that spans across a window boundary (hence encompasses an intersection point) reaches a threshold for line size and all other segments are either in category 1 (visible) or in category 2 (invisible). The midpoint coordinates (xm,ym) of a line joining (x1,y1) and (x2,y2) are given by

xm=x1+x2/2 ym=y1+y2/2

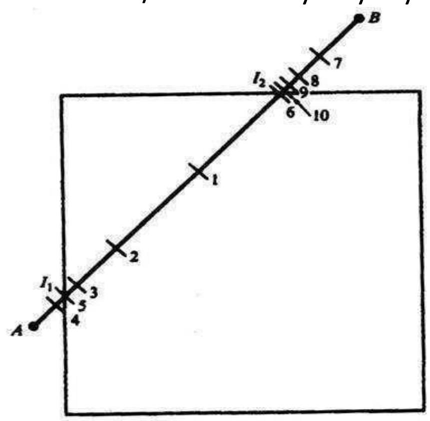


Figure 2.15 Mid-point Subdivision Algorithm

The example in above figure illustrates how midpoint subdivision is used to zoom in onto the two intersection points and with 10 bisections. The process continues until we reach two line segments that are, say, pixel – sized. i.e. mapped to one single pixel each in the image space. If the maximum number of pixels in a line is M, this method will yield a pixel-sized line segment in N subdivisions, where 2iI2IN = M or N = log2 M. For instance, when M = 1024 we sneed at most N = log2 1024 = 10 subdivisions.

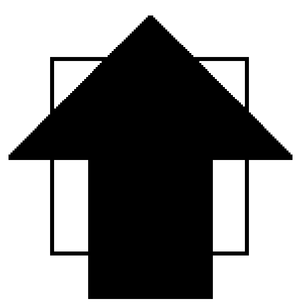
Polygon Clipping

An algorithm that clips a polygon must deal with many different cases. The case is particularly note worthy in that the concave polygon is clipped into two separate polygons. All in all, the task of clipping seems rather complex. Each edge of the polygon must be tested against each edge of the clip rectangle; new edges must be added, and existing edges must be discarded, retained, or divided. Multiple polygons may result from clipping a single polygon. We need an organized way to deal with all these cases.

Sutherland - Hodgman Polygon Clipping

Sutherland and Hodgman's polygon-clipping algorithm uses a divide-and-conquer strategy: It solves a series of simple and identical problems that, when combined, solve the overall problem. The simple problem is to clip a polygon against a single infinite clip edge. Four clip edges, each defining one boundary of the clip rectangle, successively clip a polygon against a clip rectangle.

Note the difference between this strategy for a polygon and the Cohen-Sutherland algorithm for clipping a line: The polygon clipper clips against four edges in succession, whereas the line clipper tests the out code to see which edge is crossed, and clips only when necessary.



Before clipping

Figure 2.16 Clipping Using Sutherland-Hodgeman Polygon Clipping

This figure represents a polygon (the large, solid, upward pointing arrow) before clipping has occurred.
The following figures show how this algorithm works at each edge, clipping the polygon.

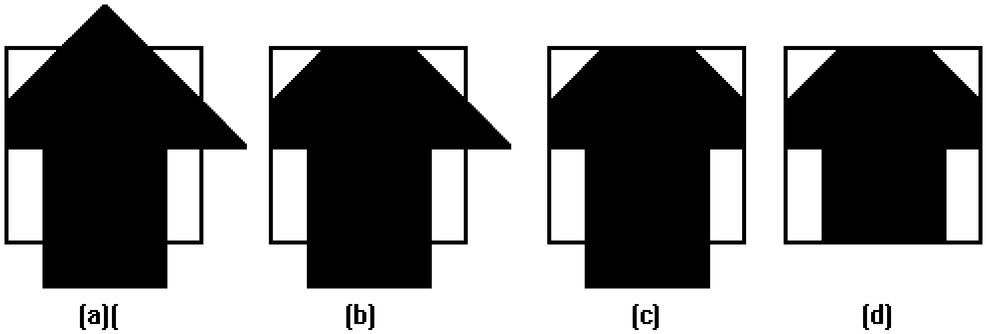


Figure 2.17 Clipping Using Sutherland-Hodgeman Polygon Clipping

a. Clipping against the left side of the clip window.
b. Clipping against the top side of the clip window.
c. Clipping against the right side of the clip window.
d. Clipping against the bottom side of the clip window.

Four Types of Edges

As the algorithm goes around the edges of the window, clipping the polygon, it encounters four types of edges. All four edge types are illustrated by the polygon in the following figure. For each edge type, zero, one, or two vertices are added to the output list of vertices that define the clipped polygon.

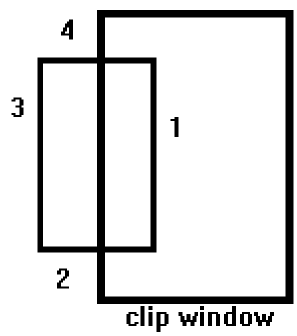


Figure 2.18 Concept of Sutherland-Hodgeman Polygon Clipping

The four types of edges are:
1. Edges that are totally inside the clip window. - add the second inside vertex point
2. Edges that are leaving the clip window. - add the intersection point as a vertex
3. Edges that are entirely outside the clip window. - add nothing to the vertex output list
4. Edges that are entering the clip window. - save the intersection and inside points as vertices

How to Calculate Intersections

Assume that we're clipping a polygon's edge with vertices at (x1,y1) and (x2,y2) against a clip window with vertices at (xmin, ymin) and (xmax,ymax).
The location (IX, IY) of the intersection of the edge with the left side of the window is:
i. IX = xmin
ii. IY = slope*(xmin-x1) + y1, where the slope = (y2-y1)/(x2-x1)

The location of the intersection of the edge with the right side of the window is:
i. IX = xmax
ii. IY = slope*(xmax-x1) + y1, where the slope = (y2-y1)/(x2-x1)

The intersection of the polygon's edge with the top side of the window is:
i. IX = x1 + (ymax - y1) / slope
ii. IY = ymax

Finally, the intersection of the edge with the bottom side of the window is:
i. IX = x1 + (ymin - y1) / slope
ii. IY = ymin

Weiler-Atherton Algorithm

• General clipping algorithm for concave polygons with holes
• Produces multiple polygons (with holes)
• Make linked list data structure
• Traverse to make new polygons

Figure 2.19 Clipping Using Weiler-Atherton Polygon Clipping

- Given polygons A and B as linked list of vertices (counter-clockwise order)
- Find all edge intersections & place in list
- Insert as "intersection" nodes
- Nodes point to A & B
- Determine in/out status of vertices

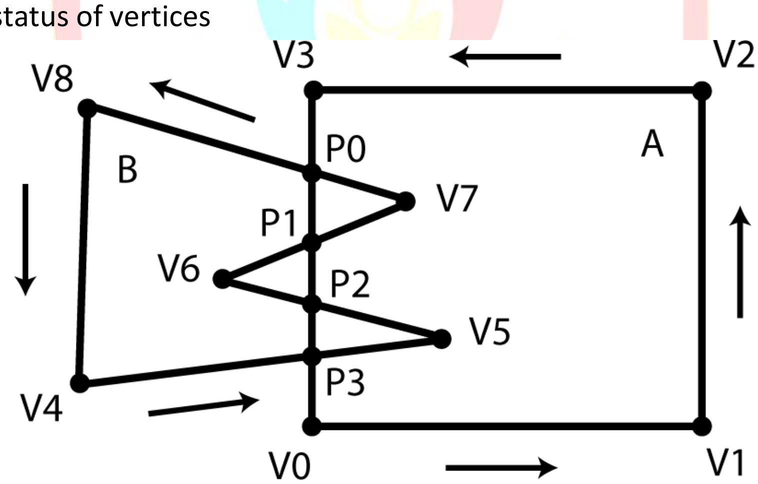


Figure 2.19 Example of Weiler-Atherton Polygon Clipping

- If "intersecting" edges are parallel, ignore
- Intersection point is a vertex

- Vertex of A lies on a vertex or edge of B
- Edge of A runs through a vertex of B
- Replace vertex with an intersection node

Weiler-Atherton Algorithm: Union

- Find a vertex of A outside of B
- Traverse linked list
- At each intersection point switch to other polygon
- Do until return to starting vertex
- All visited vertices and nodes define union'ed polygon

Weiler-Atherton Algorithm: Intersection

- Start at intersection point
  - If connected to an "inside" vertex, go there
  - Else step to an intersection point
  - If neither, stop
- Traverse linked list
- At each intersection point switch to other polygon and remove intersection point from list
- Do until return to starting intersection point
- If intersection list not empty, pick another one
- All visited vertices and nodes define and'ed polygon

If polygons don't intersect

- Union
  - If one inside the other, return polygon that surrounds the other
  - Else, return both polygons
- Intersection
  - If one inside the other, return polygon inside the other

Else, return no polygons

Point P Inside a Polygon?

- Connect P with another point P` that you know is outside polygon
- Intersect segment PP` with polygon edges
- Watch out for vertices!
- If # intersections is even (or 0) : Outside
- If odd : Insid

Unit-III

Three–Dimensional Transformations

With respect to some three- dimensional coordinate system, an object Obj is considered as a set of points.

$$Obj = \{ P (x,y,z)\}$$

If the object moved to a new position, we can regard it as a new object Obj', all of whose coordinate points P' (x', y', z') can be obtained from the original coordinate points P(x,y,z) of Obj through the application of a geometric transformation.

Translation

An object is displaced a given distance and direction from its original position. The direction and displacement of the translation is prescribed by a vector

$$V = aI + bJ + cK$$

The new coordinates of a translated point can be calculated by using the transformation

$$x' = x + a$$
$$y' = y + b$$
$$z' = z + c$$

In order to represent this transformation as a matrix transformation, we need to use homogeneous coordinate. The required homogeneous matrix transformation can then be expressed as

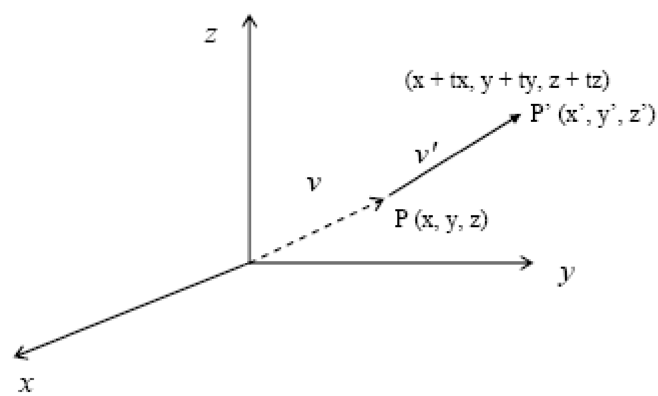


Figure 3.1 Three Dimensional Translation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling

The process of scaling changes the dimensions of an object. The scale factor s determines whether the scaling is a magnification, s>1, or a reduction, s<1.

Scaling with respect to the origin, where the origin remains fixed, is offered by the transformation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation

Rotation in three dimensions is considerably more complex then rotation in two dimensions. In 2-D, a rotation is prescribed by an angle of rotation θ and a centre of rotation, say P.

In two dimensions, a rotation is prescribed by an angle of rotations require the prescription of an angle of rotation and an axis of rotation. The canonical rotations are defined when one of the positive x, y or z coordinate axes is chosen as the axis of rotation. Then the construction of the rotation transformation proceeds just like that of a rotation in two dimensions about the origin. The Corresponding matrix transformations are

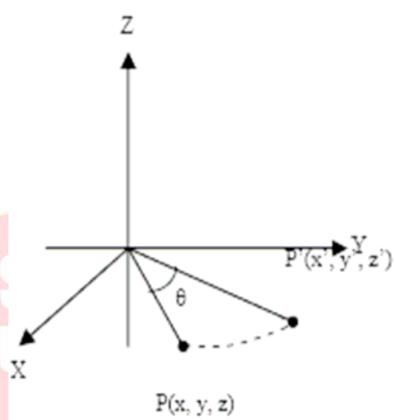


Figure 3.2 Three Dimensional Rotation

Rotation about the z axis

$$\begin{bmatrix} \cos & -\sin & 0 & 0 \\ \sin & \cos & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about the y axis

$$\begin{bmatrix} \cos & 0 & -\sin & 0 \\ 0 & 1 & 0 & 0 \\ \sin & 0 & \cos & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about the x axis

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos & \sin & 0 \\ 0 & -\sin & \cos & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that the direction of a positive angle of rotation is chosen in accordance to the right-hand rule with respect to the axis of rotation.

Parallel & Perspective Projections

Projection operations convert the viewing-coordinate description (3D) to coordinate positions on the projection plane (2D). There are 2 basic projection methods:

1. Parallel Projection transforms object positions to the view plane along parallel lines.

A parallel projection preserves relative proportions of objects. Accurate views of the various sides of an object are obtained with a parallel projection. But not a realistic representation

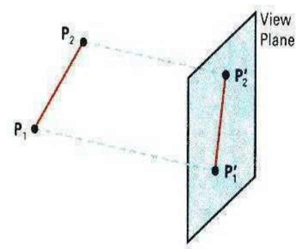


Figure 3.3 Parallel Projection

2. Perspective Projection transforms object positions to the view plane while converging to a center point of projection. Perspective projection produces realistic views but does not preserve relative proportions. Projections of distant objects are smaller than the projections of objects of the same size that are closer to the projection plane.

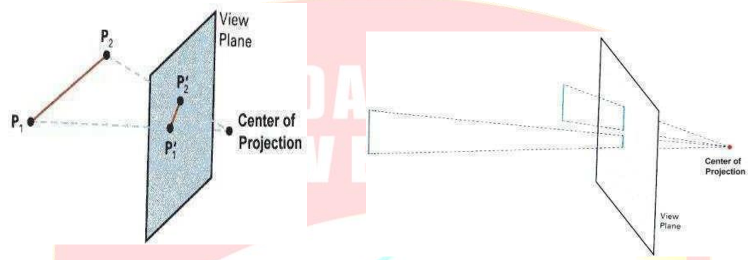


Figure 3.4 Perspective Projection

Parallel Projection Classification:

Orthographic Parallel Projection and Oblique Projection:

Figure 3.5 Orthographic Parallel Projection and Oblique Parallel Projection

Orthographic parallel projections are done by projecting points along parallel lines that are perpendicular to the projection plane.

Axonometric: The required transformation is produced in two stages, firstly the direction of projection is rotated until it aligns with one of the co-ordinate axes and then an orthographic projection along that axis is carried out. For example in the isometric case the direction of projection must be symmetric with respect to the three co-ordinate directions to allow equal foreshortening on each axis. This is obviously achieved by using a direction of projection (1,1,1).

Some special Orthographic Parallel Projections involve Plan View (Top projection), Side Elevations, and Isometric Projection.

- Front Projection
- Top Projection
- Side Projection

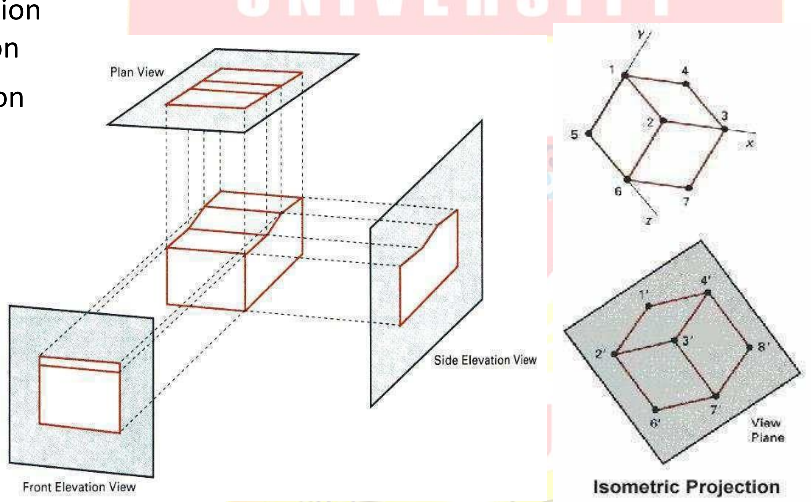


Figure 3.6 Front View, Top View, Side View and Isometric Projection

Oblique projections

Oblique projections are obtained by projecting along parallel lines that are not perpendicular to the projection plane.

The following results can be obtained from oblique projections of a cube:

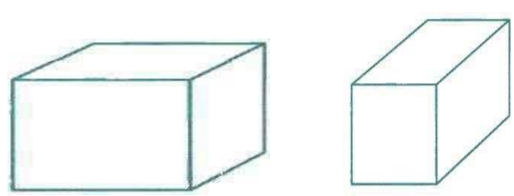


Figure 3.7 Oblique Projection

Main Types of Oblique Projections:

Cavalier: Angle between projectors and projection plane is 45^o. Perpendicular faces are projected at full scale.

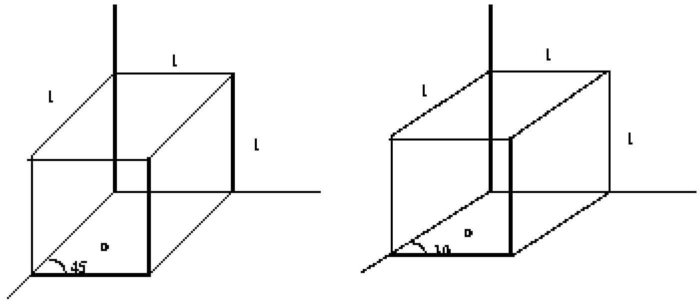


Figure 3.8 Cavalier Projection of Unit Cube

Cabinet: Angle between projectors and projection plane is arc tan(2) = 63.4^o. Perpendicular faces are projected at 50% scale.

Perspective Projection

In perspective projection, the distance from the center of projection to project plane is finite and the size of the object varies inversely with distance which looks more realistic.

The distance and angles are not preserved and parallel lines do not remain parallel. Instead, they all converge at a single point calledcenter of projection or projection reference point. There are 3 types of perspective projections which are shown in the following chart.

- One point perspective projection is simple to draw.
- Two point perspective projection gives better impression of depth.
- Three point perspective projection is most difficult to draw.

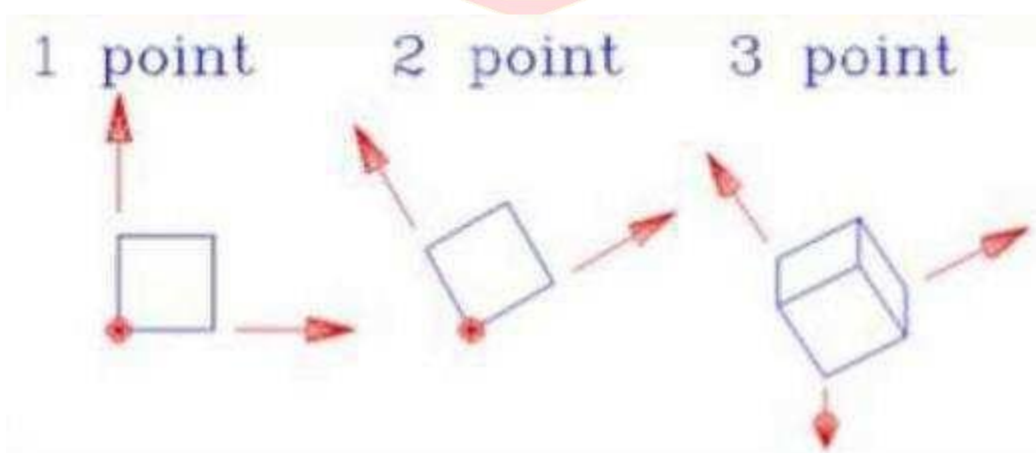


Figure 3.10 Type of Perspective Projection

Back-Face Detection

In a solid object, there are surfaces which are facing the viewer (front faces) and there are surfaces which are opposite to the viewer (back faces). These back faces contribute to approximately half of the total number of surfaces. Since we cannot see these surfaces anyway, to save processing time, we can remove them before the clipping process with a simple test. Each surface has a normal vector. If this vector is pointing in the direction of the center of projection, it is a front face and can be seen by the viewer. If it is pointing away from the center of projection, it is a back face and cannot be seen by the viewer. The test is very simple, if the z component of the normal vector is positive, then, it is a back face. If the z component of the vector is negative, it is a front face. Note that this technique only caters well for non-overlapping convex polyhedral. For other cases where there are concave polyhedra or overlapping objects, we still need to apply other methods to further determine where the obscured faces are partially or completely hidden by other objects (eg. Using Depth-Buffer Method or Depth-sort Method).

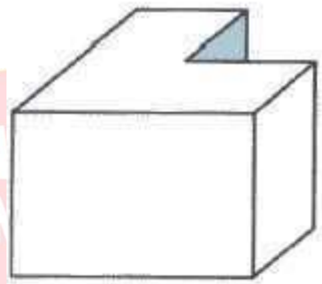


Figure 3.11 Representation of Object

The Z-buffer Algorithm

This method is developed by Cutmull. It is an image-space approach. The basic idea is to test the Z-depth of each surface to determine the closest (visible) surface.

In this method, each surface is processed separately one pixel position at a time across the surface. The depth values for a pixel are compared and the closest (smallest z) surface determines the color to be displayed in the frame buffer.

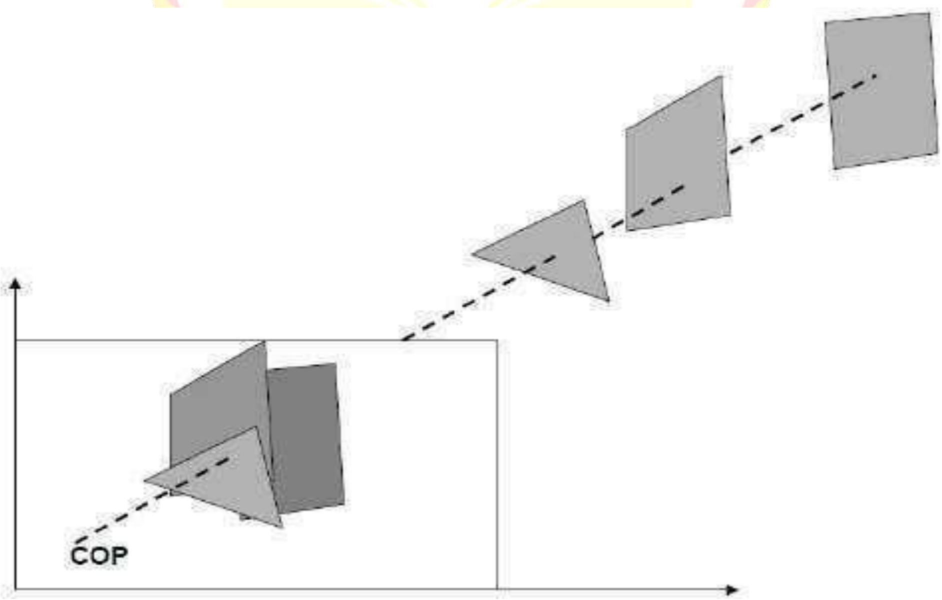


Figure 3.12 Z-Buffer Algorithm

It is applied very efficiently on surfaces of polygon. Surfaces can be processed in any order. To override the closer polygons from the far ones, two buffers named frame buffer and depth buffer, are used.

Depth buffer is used to store depth values for (x, y) position, as surfaces are processed ($0 \leq depth \leq 1$).

The frame buffer is used to store the intensity value of color value at each position (x, y).
The z-coordinates are usually normalized to the range [0, 1]. The 0 value for z-coordinate indicates back clipping pane and 1 value for z-coordinates indicates front clipping pane.

Algorithm
Step-1 − Set the buffer values −
Depthbuffer (x, y) = 0
Framebuffer (x, y) = background color
Step-2 − Process each polygon (One at a time)
For each projected (x, y) pixel position of a polygon, calculate depth z.
If Z > depthbuffer (x, y)
Compute surface color,
set depthbuffer (x, y) = z,
framebuffer (x, y) = surfacecolor (x, y)

Advantages

- It is easy to implement.
- It reduces the speed problem if implemented in hardware.
- It processes one object at a time.

Disadvantages

- It requires large memory.
- It is time consuming process.

Back-Face Detection

A fast and simple object-space method for identifying the back faces of a polyhedron is based on the "inside-outside" tests. A point (x, y, z) is "inside" a polygon surface with plane parameters A, B, C, and D if When an inside point is along the line of sight to the surface, the polygon must be a back face (we are inside that face and cannot see the front of it from our viewing position).
We can simplify this test by considering the normal vector **N** to a polygon surface, which has Cartesian components (A, B, C).
In general, if V is a vector in the viewing direction from the eye (or "camera") position, then this polygon is a back-face if

$$\mathbf{V.N > 0}$$

Furthermore, if object descriptions are converted to projection coordinates and your viewing direction is parallel to the viewing z-axis, then −

$$V = (0, 0, V_z) \text{ and } V.N = V_Z C$$

So that we only need to consider the sign of C the component of the normal vector **N**.

In a right-handed viewing system with viewing direction along the negative ZVZV axis, the polygon is a back face if C < 0. Also, we cannot see any face whose normal has z component C = 0, since your viewing direction is towards that polygon. Thus, in general, we can label any polygon as a back face if its normal vector has a z component value −

$$\mathbf{C <= 0}$$

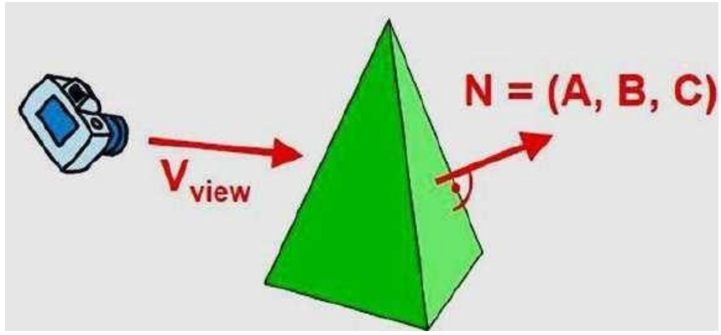


Figure 3.13 Back Face Detection

Similar methods can be used in packages that employ a left-handed viewing system. In these packages, plane parameters A, B, C and D can be calculated from polygon vertex coordinates specified in a clockwise direction (unlike the counter clockwise direction used in a right-handed system).

Also, back faces have normal vectors that point away from the viewing position and are identified by C >= 0 when the viewing direction is along the positive ZvZv axis. By examining parameter C for the different planes defining an object, we can immediately identify all the back faces.

Depth Sorting Method

Depth sorting method uses both image space and object-space operations. The depth-sorting method performs two basic functions −

- First, the surfaces are sorted in order of decreasing depth.
- Second, the surfaces are scan-converted in order, starting with the surface of greatest depth.

The scan conversion of the polygon surfaces is performed in image space. This method for solving the hidden-surface problem is often referred to as the **painter's algorithm**. The following figure shows the effect of depth sorting −

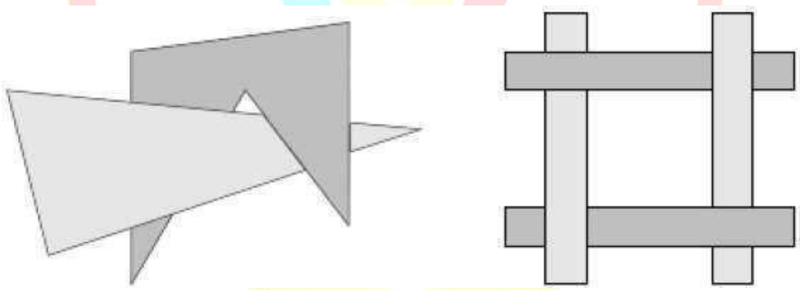


Figure 3.14 Depth Sorting Algorithm

The algorithm begins by sorting by depth. For example, the initial "depth" estimate of a polygon may be taken to be the closest z value of any vertex of the polygon.

Let us take the polygon P at the end of the list. Consider all polygons Q whose z-extents overlap P's. Before drawing P, we make the following tests. If any of the following tests is positive, then we can assume P can be drawn before Q.

- Do the x-extents not overlap?
- Do the y-extents not overlap?
- Is P entirely on the opposite side of Q's plane from the viewpoint?
- Is Q entirely on the same side of P's plane as the viewpoint?
- Do the projections of the polygons not overlap?

If all the tests fail, then we split either P or Q using the plane of the other. The new cut polygons are inserting into the depth order and the process continues. Theoretically, this partitioning could generate $O(n^2)$ individual polygons, but in practice, the number of polygons is much smaller.

Binary Space Partition (BSP) Trees

Binary space partitioning is used to calculate visibility. To build the BSP trees, one should start with polygons and label all the edges. Dealing with only one edge at a time, extend each edge so that it splits the plane in two. Place the first edge in the tree as root. Add subsequent edges based on whether they are inside or outside. Edges that span the extension of an edge that is already in the tree are split into two and both are added to the tree.

- From the above figure, first take **A** as a root.
- Make a list of all nodes in figure (a).
- Put all the nodes that are in front of root **A** to the left side of node **A** and put all those nodes that are behind the root **A** to the right side as shown in figure (b).
- Process all the front nodes first and then the nodes at the back.
- As shown in figure (c), we will first process the node **B**. As there is nothing in front of the node **B**, we have put NIL. However, we have node **C** at back of node **B**, so node **C** will go to the right side of node **B**.
- Repeat the same process for the node **D**.

The z-Buffer algorithm is one of the most commonly used routines. It is simple, easy to implement, and is often found in hardware.

The idea behind it is uncomplicated: Assign a z-value to each polygon and then display the one (pixel by pixel) that has the smallest value.

There are some advantages and disadvantages to this:

Advantages:

- Simple to use
- Can be implemented easily in object or image space
- Can be executed quickly, even with many polygons
- Disadvantages:
- Takes up a lot of memory
- Can't do transparent surfaces without additional code

Consider these two polygons (right: edge-on left: head-on).

The computer would start (arbitrarily) with Polygon 1 and put its depth value into the buffer. It would do the same for the next polygon, P2. It will then check each overlapping pixel and check to see which one is closer to the viewer, and display the appropriate color.

This is a simplistic example, but the basic ideas are valid for polygons in any orientation and permutation (this algorithm will properly display polygons piercing one another, and polygons with conflicting depths, such as:

Visible Surface Determination: Painter's Algorithm

The painter's algorithm is based on depth sorting and is a combined object and image space algorithm. It is as follows:

1. Sort all polygons according to z value (object space); Simplest to use maximum z value
2. Draw polygons from back (maximum z) to front (minimum z)

This can be used for wireframe drawings as well by the following:

1. Draw solid polygons using Poly scan (in the background color) followed by Polyline (polygon color).
2. Poly scan erases Polygons behind it then Polyline draws new Polygon.

Problems with simple Painter's algorithm

Look at cases where it doesn't work correctly. S has a greater depth than S' and so will be drawn first. But S' should be drawn first since it is obscured by S. We must somehow reorder S and S'.:

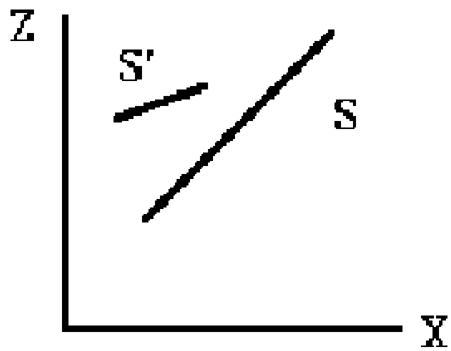


Figure 3.15 Representation of Lines

We will perform a series of tests to determine, if two polygons need to be reordered. If the polygons fail a test, then the next test must be performed. If the polygons fail all tests, then they are reordered. The initial tests are computationally cheap, but the later tests are more expensive.

So look at revised algorithm to test for possible reordering could store Zmax, Zmin for each Polygon.

- sort on Zmax
- start with polygon with greatest depth (S)
- compare S with all other polygons (P) to see if there is any depth overlap (Test 0)

If S.Zmin <= P.Zmax then have depth overlap (as in above and below figures)

Figure 3.16 Simple Painter's Algorithm

If have depth overlap (failed Test 0) we may need to reorder polygons.

Next (Test 1) check to see if polygons overlap in xy plane (use bounding rectangles)

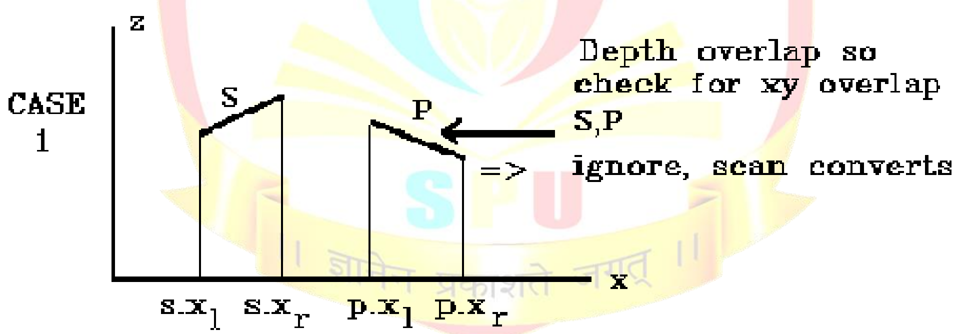


Figure 3.17 Case I

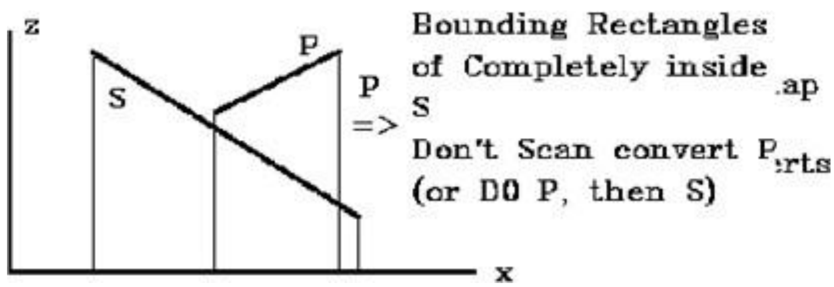


Figure 3.18 Case II

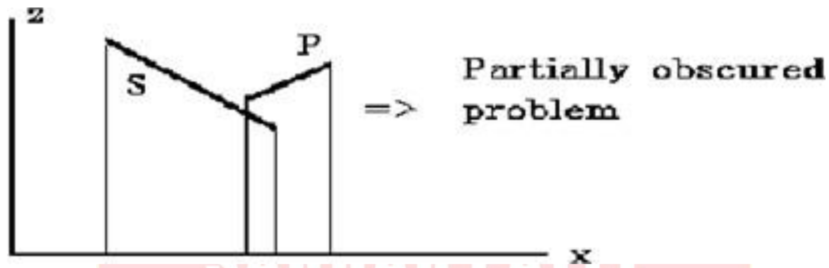


Figure 3.19 Case III

Do above tests for x and y.

If have case 1 or 2 then we are done (passed Test 1) but for case 3 we need further testing failed Test 1)
Next test (Test 2) to see if polygon S is "outside" of polygon P (relative to view plane)
Remember: a point (x, y, z) is "outside" of a plane if we put that point into the plane equation and get:

$Ax + By + Cz + D > 0$

So to test for S outside of P, put all vertices of S into the plane equation for P and check that all vertices give a result that is > 0.

i.e. $Ax' + By' + Cz' + D > 0$ x', y', z' are S vertices

A, B, C, D are from plane equation of P (choose normal away from view plane since define "outside" with respect to the view plae)

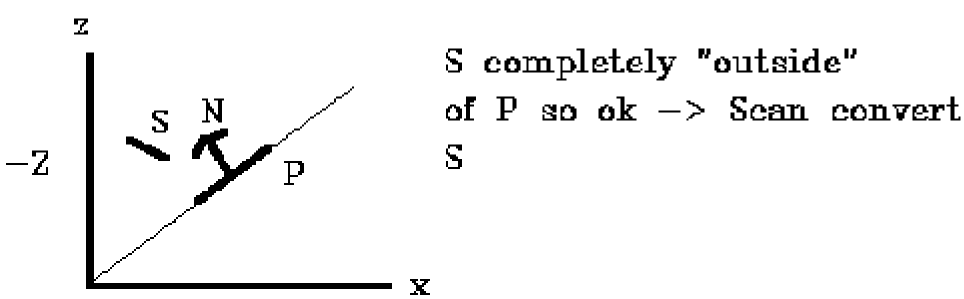


Figure 3.20 Case IV

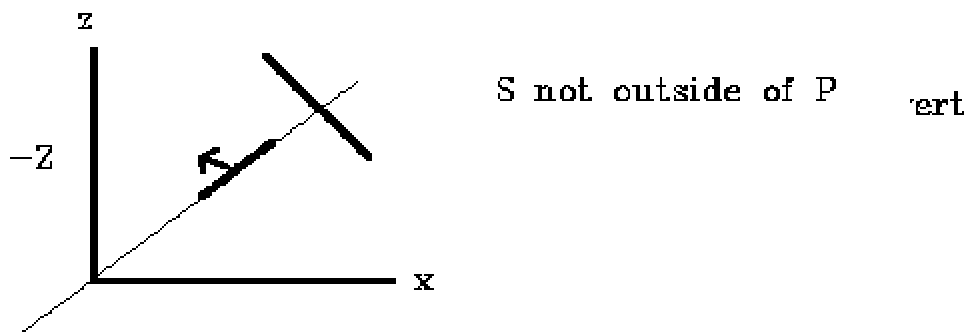


Figure 3.21 Case V

If the test of S "outside" of P fails, then test to see if P is "inside" of S (again with respect to the view plane) (Test 3).

Compute plane equation of S and put in all vertices of P, if all vertices of P inside of S then P inside.

inside test: Ax' + By' + Cz' + D &lt0 where x', y', z' are coordinates of P vertices.

Figure 3.22 Case VI

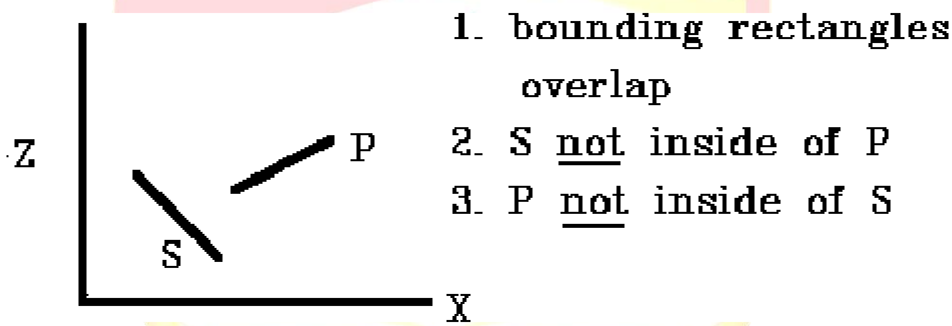
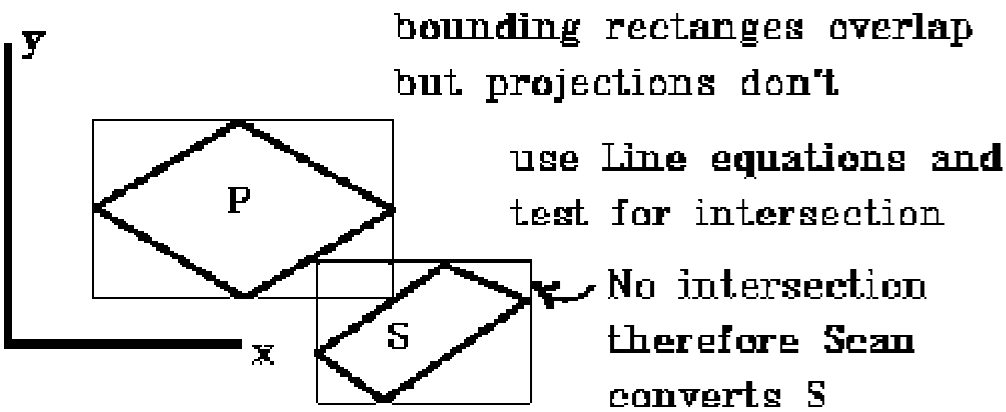


Figure 3.23 Case VII

Then we do the 4th test and check for overlap for actual projections in xy plane since may have bounding rectangles overlap but not actual overlap

For example: Look at projection of two polygons in the xy plane

Then have two possible cases.



All 4 tests have failed therefore interchange P and S and scan convert P before S. But before we scan convert P we must test P against all other polygons. Look at an example of multiple interchanges

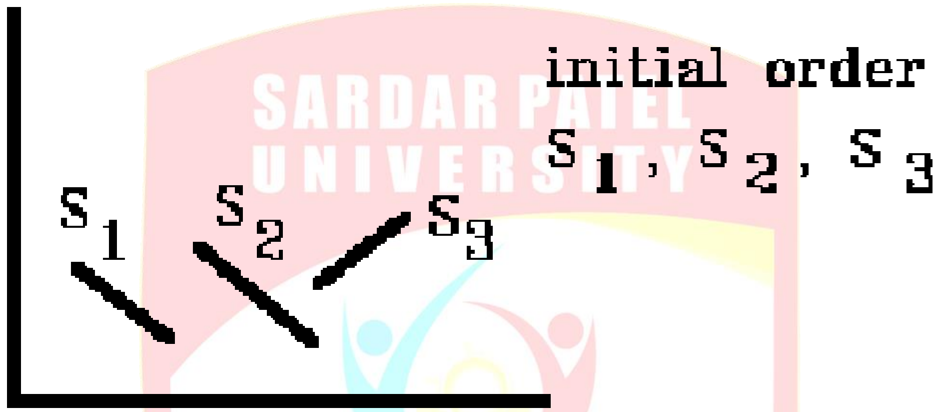


Figure 3.25 Case IX

Test S1 against S2 and it fails all tests so reorder: S2, S1, S3
Test S2 against S3 and it fails all tests so reorder: S3, S2, S1
Possible Problem: Polygons that alternately obscure one another. These three polygons will continuously reorder.
One solution might be to flag a reordered polygon and subdivide the polygon into several smaller polygons.

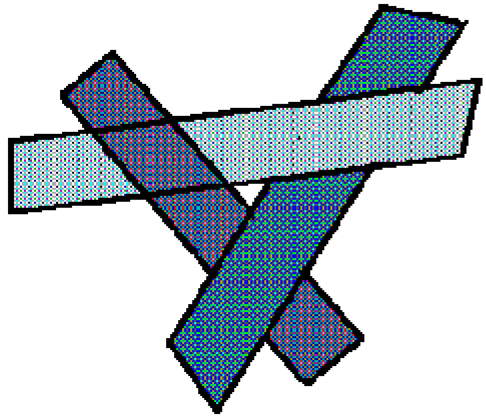


Figure 3.26 Final Object

Bezier Curves:

Bezier curve is discovered by the French engineer **Pierre Bézier**. These curves can be generated under the control of other points. Approximate tangents by using control points are used to generate curve. The Bezier curve can be represented mathematically as –

Suppose we are given n+1 control point positions : Pk = (Xk,Yk,Zk), with k = o to n. These coordinate points are blended to produce the position vector P(u), which describes the path of an approximating Bezier Polynomial Function between Po and Pn.

$$P(u) = \sum_{=0}^{n} \mathrm{Pk\ BEZk, n(u)} \qquad 0<=u<=1$$

The Bezier blending functions $BEZ_{k,n}(u)$ are the Bernstein polynomials.

$$BEZ_{k,n}(u) = C(n,k)u^{k}(1-u)^{n-k}$$

Where,

C(n,k) are the binomial coefficients.

C(n,k) = n! / k!(n-k)!

The simplest Bézier curve is the straight line from the point P0P0 to P1P1. A quadratic Bezier curve is determined by three control points. A cubic Bezier curve is determined by four control points.

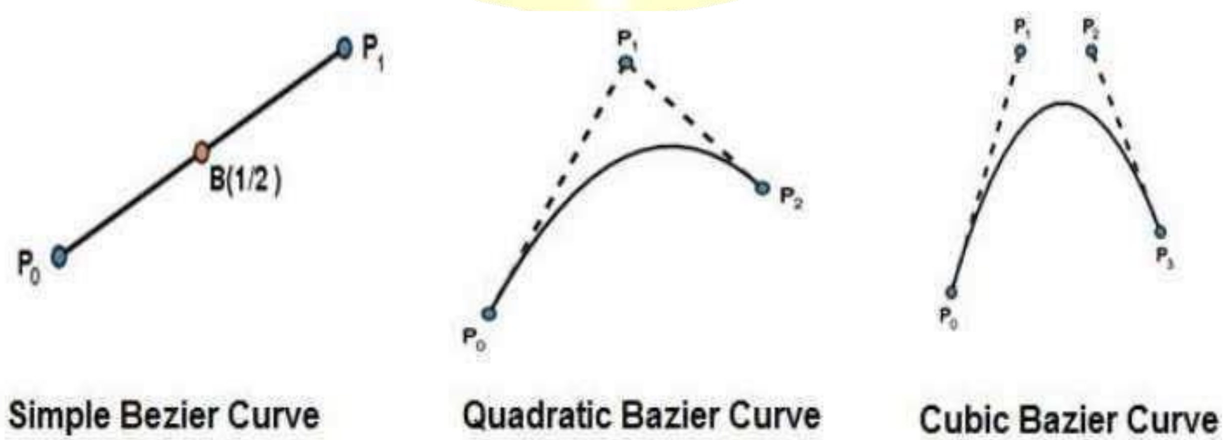


Figure 3.27 Representation of Bezier Curve

Properties of Bezier Curves

Bezier curves have the following properties −

- They generally follow the shape of the control polygon, which consists of the segments joining the control points.
- They always pass through the first and last control points.
- They are contained in the convex hull of their defining control points.
- The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- A Bezier curve generally follows the shape of the defining polygon.
- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.
- They are invariant under an affine transformation.
- Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
- A given Bezier curve can be subdivided at a point t=t0 into two Bezier segments which join together at the point corresponding to the parameter value t=t0.

B-Spline Curves

The Bezier-curve produced by the Bernstein basis function has limited flexibility.

- First, the number of specified polygon vertices fixes the order of the resulting polynomial which defines the curve.
- The second limiting characteristic is that the value of the blending function is nonzero for all parameter values over the entire curve.

The B-spline basis contains the Bernstein basis as the special case. The B-spline basis is non global.

A B-spline curve is defined as a linear combination of control points Pi and B-spline basis function Ni, k t given by

$$C(t) = \sum_{i=0}^{n} P_i N_{i,k}(t), \qquad n \geq k-1, \qquad t \in [tk-1, tn+1]$$

Where,

- {pi: i=0, 1, 2….n} are the control points
- k is the order of the polynomial segments of the B-spline curve. Order k means that the curve is made up of piecewise polynomial segments of degree k - 1,
- the Ni, k(t) are the “normalized B-spline blending functions”. They are described by the order k and by a non-decreasing sequence of real numbers normally called the “knot sequence”.

ti: i = 0, . . . n + K

The Ni, k functions are described as follows −

$$N_{i,1}(t) = \begin{cases} 1, & if\, u \in [t_i, t_{i+1}) \\ 0, & Otherwise \end{cases}$$

and if k > 1,

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1}} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

And

$$t \in [t_{k-1}, t_{n+1})$$

Properties of B-spline Curve

- B-spline curves have the following properties −
- The sum of the B-spline basis functions for any parameter value is 1.
- Each basis function is positive or zero for all parameter values.
- Each basis function has precisely one maximum value, except for k=1.
- The maximum order of the curve is equal to the number of vertices of defining polygon.
- The degree of B-spline polynomial is independent on the number of vertices of defining polygon.
- B-spline allows the local control over the curve surface because each vertex affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.
- The curve exhibits the variation diminishing property.
- The curve generally follows the shape of defining polygon.
- Any affine transformation can be applied to the curve by applying it to the vertices of defining polygon.
- The curve line within the convex hull of its defining polygon.

Basic Illumination Model:

A simple illumination model (called a lighting model in OpenGL) can be based on three components: ambient reflection, diffuse reflection and specular reflection. The model we look at is known as the Phong model. The OpenGL lighting model has the components of the Phong model, but is more elaborate.

Ambient Reflection

Imagine a uniform intensity background light which illuminates all objects in the scene equally from all directions. This is known as ambient light. The ambient light model attempts to capture the effect of light which results from many object-object reflections without detailed calculations, that is, achieve computational efficiency. Those detailed calculations are performed in some rendering techniques such as radiosity which attempt to achieve higher quality images, but not real-time.

Diffuse Reflection

Objects illuminated by only ambient light have equal intensity everywhere. If there are light sources in a scene then different objects should have different intensities based on distance and orientation with respect to the light source and the viewer.

A point on a diffuse surface appears equally bright from all viewing positions because it reflects light equally in all directions. That is, its intensity is independent of the position of the viewer.

Whilst independent of the viewer, the intensity of a point on a diffuse surface does depend on the orientation of the surface with respect to the light source and the distance to the light source.
A simple model for diffuse reflection is Lambertian reflection.

Assuming the following geometry

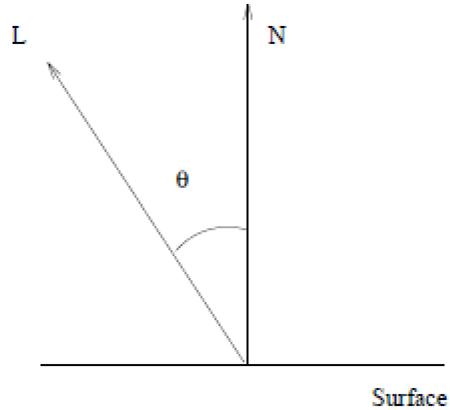


Figure 3.28 The direction of light is measured from the surface normal

Then the intensity of the diffuse reflection from a point light source is

$$I = L_d k_d \cos$$

Where Ld is the intensity of the (point) light source, kd is the diffuse reflection coefficient of the object's material, and θ is the angle between the normal to the surface and the light source direction vector.

If the normal vector to the surface N and the light source direction vector L are both normalized then the above equation can be simplified to

$$I = L_d k_d (N.L)$$

If a light source is an infinite distance from the object then L will be the same for all points on the object — the light source becomes a directional light source. In this case less computation can be performed.

Adding the ambient reflection and diffuse reflection contributions together we get

$$I = L_a k_a + L_d k_d (N.L)$$

Specular Reflection

Specular reflection occurs on hard, shiny surfaces and is seen as highlights. Specular highlights are strongly directional. The approach to specular reflection in the Phong model is that the specular reflection intensity drops off as the cosine of the angle between the normal and the specular reflection direction raised to some power n which indicates the shininess of the surface. The higher the power of n the smaller and brighter the highlight.

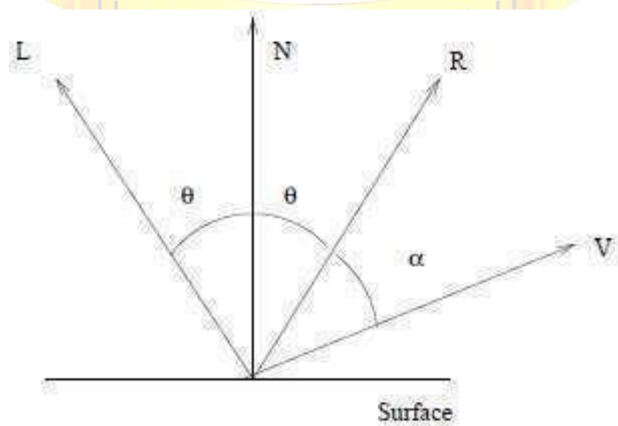


Figure 3.29 Specular Reflection

The specular component of the illumination model may thus be given as

$$I = f_{att} L_{s\lambda} k_{s\lambda} cos^n \alpha$$

If the direction of (specular) reflection R and the viewpoint direction V are normalised then the equation becomes

$$I = f_{att} L_{s\lambda} k_{s\lambda} (\mathbf{R}.\mathbf{V})^n$$

The full lighting equation becomes

$$I_\lambda = L_{a\lambda} k_{a\lambda} + f_{att}[L_{d\lambda} k_{d\lambda}(\mathbf{N}.\mathbf{L}) + L_{s\lambda} k_{s\lambda}(\mathbf{R}.\mathbf{V})^n]$$

Phong Shading Algorithm

Using highlights avoids surfaces that look dull, lifeless, boring, blah. One cool thing about highlights is that, in addition to being all bright and shiny, they change as the object moves. In this way, highlights provide useful visual information about shape and motion.

The simplest model for approximating surface highlights is the *Phong* model, originally developed by Bui-Tong Phong. In this model, we think of the interaction between light and a surface as having three distinct components:

- Ambient
- Diffuse
- Specular

The *ambient* component is usually given a dim constant value, such as 0.2. It approximates light coming from a surface due to all the non-directional ambient light that is in the environment. In general, you'll want this to be some tinted color, rather than just gray. $[r_a,g_a,b_a]$. For example, a slightly greenish object might have an ambient color of [0.1,0.3,0.1].

The *diffuse* component is that dot product **n•L** that we discussed in class. It approximates light, originally from light source **L**, reflecting from a surface which is diffuse, or non-glossy. One example of a non-glossy surface is paper. In general, you'll also want this to have a non-gray color value, so this term would in general be a color defined as: $[r_d,g_d,b_d](\mathbf{n}\bullet\mathbf{L})$.

Finally, the Phong model has a provision for a highlight, or *specular*, component, which reflects light in a shiny way. This is defined by $[r_s,g_s,b_s](\mathbf{R}\bullet\mathbf{L})^p$, where **R** is the mirror reflection direction vector we discussed in class (and also used for ray tracing), and where *p* is a specular power. The higher the value of *p*, the shinier the surface.

The complete Phong shading model for a single light source is:

$[r_a,g_a,b_a] + [r_d,g_d,b_d]\max_0(\mathbf{n}\bullet\mathbf{L}) + [r_s,g_s,b_s]\max_0(\mathbf{R}\bullet\mathbf{L})^p$

If you have multiple light sources, the effect of each light source $\mathbf{L}_i$ will geometrically depend on the normal, and therefore on the diffuse and specular components, but not on the ambient component. Also, each light might have its own [r,g,b] color. So the complete Phong model for multiple light sources is:

$[r_a,g_a,b_a] + \Sigma_i([L_r,L_g,L_b] ([r_d,g_d,b_d]\max_0(\mathbf{n}\bullet\mathbf{L_i}) + [r_s,g_s,b_s]\max_0(\mathbf{R}\bullet\mathbf{L_i})^p))$

Below you can see three different shaders for a sphere. In all cases, the sphere is lit by two light sources: a dim source from the rear left (-1,0,-.5) and a bright source from the front upper right (1,1,.5). The spheres have, respectively, no highlight, a highlight with an exponent of *p* = 4, and a highlight with an exponent of *p* = 16.

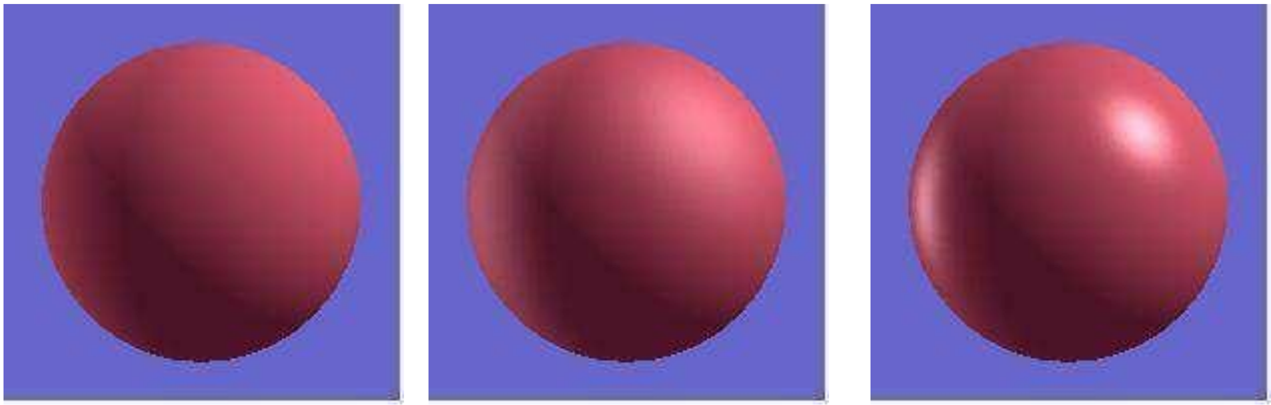


Figure 3.30 Graphical Object Representation Using Phong Shadding

Gouraud Shading Model

The second shading model, Gouraud shading, computes intensity for each vertex and then interpolates the computed intensities across the polygons. Gouraud shading performs a bi-linear interpolation of the intensities down and then across scan lines. It thus eliminates the sharp changes at polygon boundaries

The algorithm is as follows:

1. Compute a normal N for each vertex of the polygon.
2. From N compute intensity I for each vertex of the polygon.
3. From bi-linear interpolation compute intensity Ii for each pixel.
4. Paint pixel to shade corresponding to Ii.

How do we compute N for a vertex? Let N = the average of the normal of the polygons which include the vertex. Note that 4 sided polygons have 4 neighbors and triangles have 6 neighbors.

We can find the neighbors for a particular vertex by searching the polygons and including any polygons which include the vertex. Now look at performing the bi-linear intensity interpolation. This is the same as the bi-linear interpolation for the z depth of a pixel (used with the z buffer visible surface algorithm).

Advantages of Gouraud shading:

Gouraud shading gives a much better image than faceted shading (the facets no longer visible). It is not too computationally expensive: one floating point addition (for each color) for each pixel. (the mapping to actual display registers requires a floating point multiplication)

Disadvantages to Gouraud shading:

It eliminates creases that you may want to preserve, e.g. in a cube. We can modify data structure to prevent this by storing each physical vertex 3 times, i.e. a different logical vertex for each polygon. here is a data structure for a cube that will keep the edges:

Ray Tracing

Ray tracing is a technique for rendering three-dimensional graphics with very complex light interactions. This means you can create pictures full of mirrors, transparent surfaces, and shadows, with stunning results. We discuss ray tracing in this introductory graphics article because it is a very simple method to both understand and implement. It is based on the idea that you can model reflection and refraction by recursively following the path that light takes as it bounces through an environment.

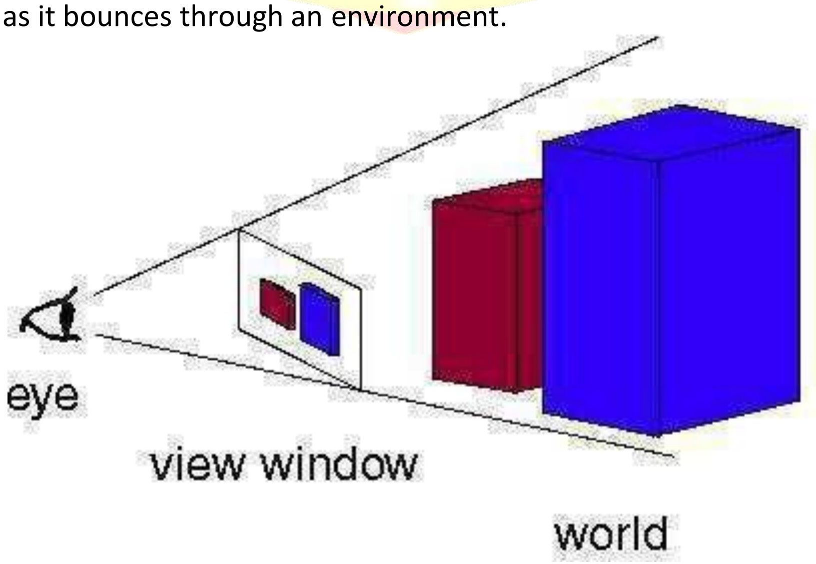


Figure 3.31 The eye, view window, and world

Ray tracing is so named because it tries to simulate the path that light rays take as they bounce around within the world - they are *traced* through the scene. The objective is to determine the color of each light ray that strikes the view window before reaching the eye. A light ray can best be thought of as a single photon (although this is not strictly accurate because light also has wave properties - but I promised there would be no theory).

The name ``ray tracing'' is a bit misleading because the natural assumption would be that rays are traced starting at their point of origin, the light source, and towards their destination, the eye. This would be an accurate way to do it, but unfortunately it tends to be very difficult due to the sheer numbers involved. Consider tracing one ray in this manner through a scene with one light and one object, such as a table. We begin at the light bulb, but first need to decide how many rays to shoot out from the bulb. Then for each ray we have to decide in what direction it is going. There is really infinity of directions in which it can travel - how do we know which to choose? Let's say we've answered these questions and are now tracing a number of photons. Some will reach the eye directly, others will bounce around some and then reach the eye and many, and many more will probably never hit the eye at all. For all the rays that never reach the eye, the effort tracing them was wasted.

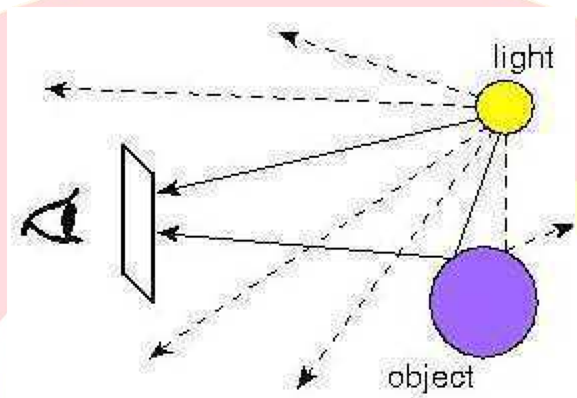


Figure 3.32 Tracing rays from the light source to the eye

In order to save ourselves this wasted effort, we trace only those rays that are *guaranteed* to hit the view window and reach the eye. It seems at first that it is impossible to know beforehand which rays reach the eye. After all, any given ray can bounce around the room many times before reaching the eye. However, if we look at the problem *backwards*, we see that it has a very simple solution. Instead of tracing the rays starting at the light source, we trace them backwards, starting at the eye.

Consider any point on the view window whose color we're trying to determine. Its color is given by the color of the light ray that passes through that point on the view window and reaches the eye. We can just as well follow the ray backwards by starting at the eye and passing through the point on its way out into the scene. The two rays will be identical, except for their direction: if the original ray came directly *from* the light source, then the backwards ray will go directly *to* the light source; if the original bounced off a table first, the backwards ray will also bounce off the table. You can see this by looking at Figure again and just reversing the directions of the arrows. So the backwards method does the same thing as the original method, except it doesn't waste any effort on rays that never reach the eye.

This, then, is how ray tracing works in computer graphics. For each pixel on the view window, we define a ray that extends from the eye to that point. We follow this ray out into the scene and as it bounces off of different objects. The final color of the ray (and therefore of the corresponding pixel) is given by the colors of the objects hit by the ray as it travels through the scene.

Just as in the light-source-to-eye method it might take a very large number of bounces before the ray ever hits the eye, in backwards method it might take many bounces before the ray every hits the light. Since we need to establish some limit on the number of bounces to follow the ray on, we make the following approximation: every time a ray hits an object, we follow a single new ray from the point of intersection *directly towards* the light source.

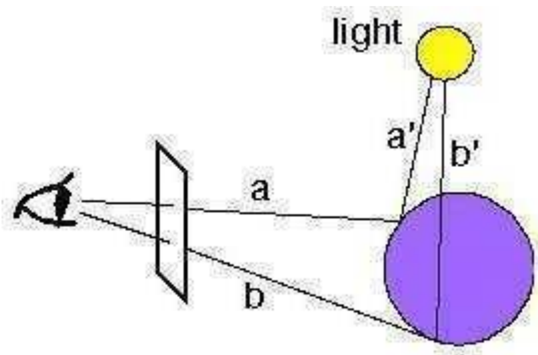


Figure 3.33 We trace a new ray from each ray-object intersection directly towards the light source

In the figure we see two rays, **a** and **b,** which intersect the purple sphere. To determine the color of **a,** we follow the new ray **a'** directly towards the light source. The color of **a** will then depend on several factors, discussed in Color and Shading below. As you can see, **b** will be shadowed because the ray **b'** towards the light source is blocked by the sphere itself. Ray **a** would have also been shadowed if another object blocked the ray **a'**.

Color Model

A color model is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components. When this model is associated with a precise description of how the components are to be interpreted (viewing conditions, etc.), the resulting set of colors is called color space.

CIE (Commission International de l'Eclairage - International Color Commision) organisation produced two models for defining color:
• 1931: Measured on 10 subjects (!) on samples subtending 2 (!) degrees of the field of view

• 1964: Measured on larger number of subjects subtending 10 degrees of field of view

• The CIE 1931 model is the most commonly used
    • It defines three primary “colors” X, Y and) that can be used to describe all visible colors, as well as a standard white, called C.

• The range of colors that can be described by combinations of other colors is called a color gamut.
- Since it is impossible to find three colors with a gamut containing all visible colors, the CIE’s three primary colors are imaginary. They cannot be seen, but they can be used to define other visible colors.

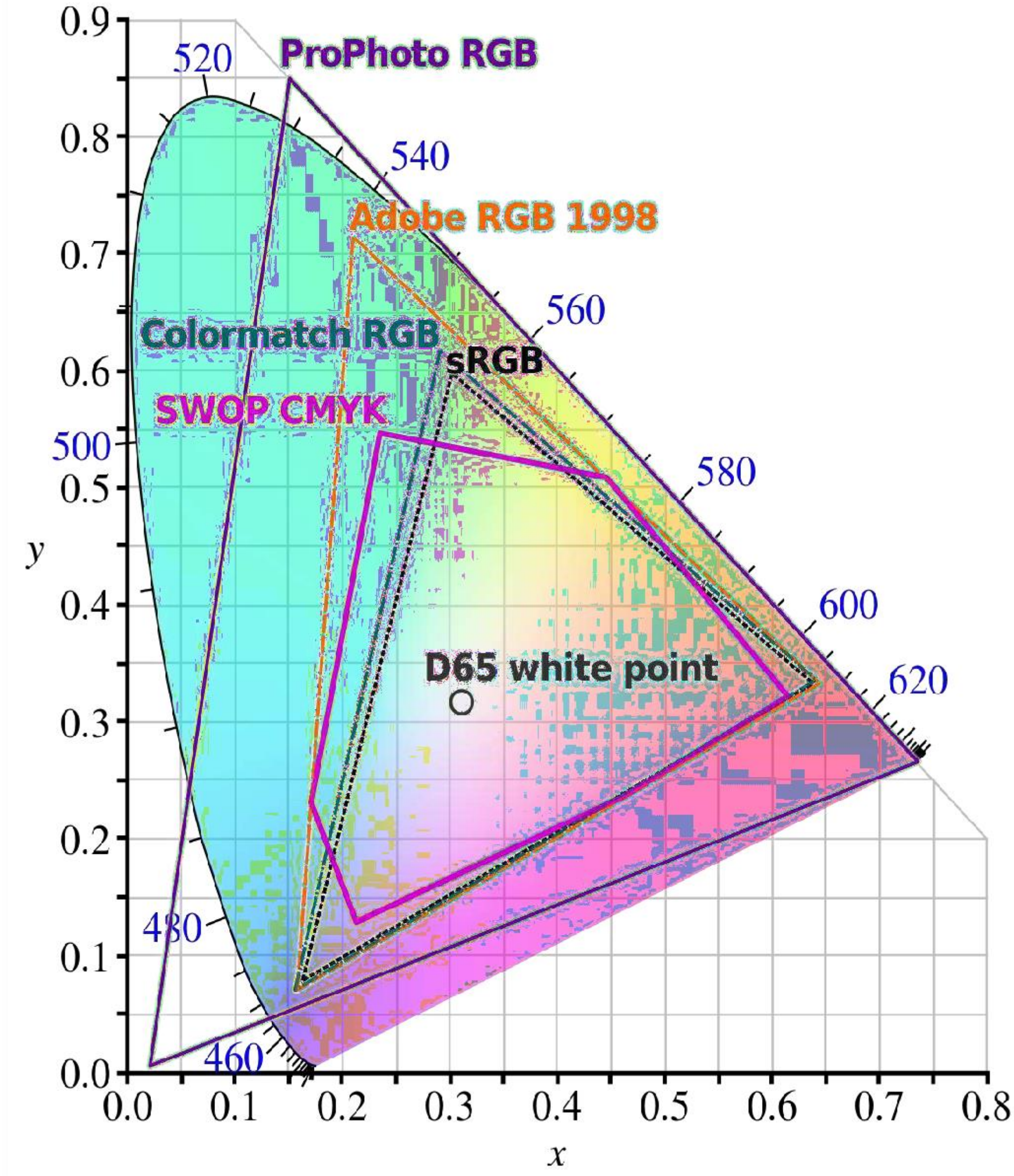


Figure 3.34 CIE Chromaticity Diagram

RGB Color Model

Media that transmit light (such as television) use additive color mixing with primary colors of red, green, and blue, each of which stimulates one of the three types of the eye's color receptors with as little stimulation as possible of the other two. This is called "RGB" color space. Mixtures of light of these primary colors cover a large part of the human color space and thus produce a large part of human color experiences. This is why color television sets or color computer monitors need only produce mixtures of red, green and blue light. See Additive color.

Other primary colors could in principle be used, but with red, green and blue the largest portion of the human color space can be captured. Unfortunately there is no exact consensus as to what loci in the chromaticity diagram the red, green, and blue colors should have, so the same RGB values can give rise to slightly different colors on different screens.

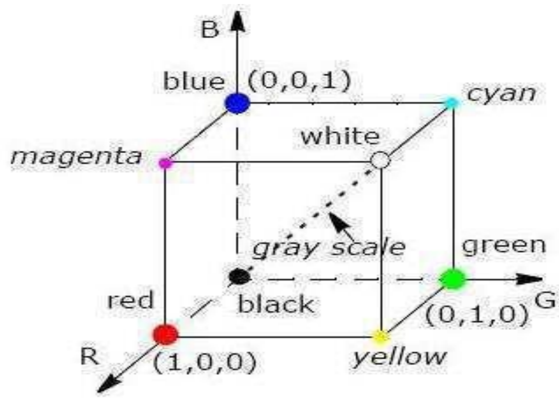


Figure 3.35 RGB Color Model

YIQ Color Model

YIQ is the color space used by the NTSC color TV system, employed mainly in North and Central America, and Japan. I stands for in-phase, while Q stands for quadrature, referring to the components used in quadrature amplitude modulation. Some forms of NTSC now use the YUV color space, which is also used by other systems such as PAL. The Y component represents the luma information, and is the only component used by black-and-white television receivers. I and Q represent the chrominance information. In YUV, the U and V components can be thought of as X and Y coordinates within the color space. I and Q can be thought of as a second pair of axes on the same graph, rotated 33°; therefore IQ and UV represent different coordinate systems on the same plane.

This model was designed to separate chrominance from luminance. This was a requirement in the early days of color television when black-and-white sets still were expected to pick up and display what were originally color pictures. The Y-channel contains luminance information (sufficient for black-and-white television sets) while the I and Q channels (in-phase and in-quadrature) carried the color information. A color television set would take these three channels, Y, I, and Q, and map the information back to R, G, and B levels for display on a screen.

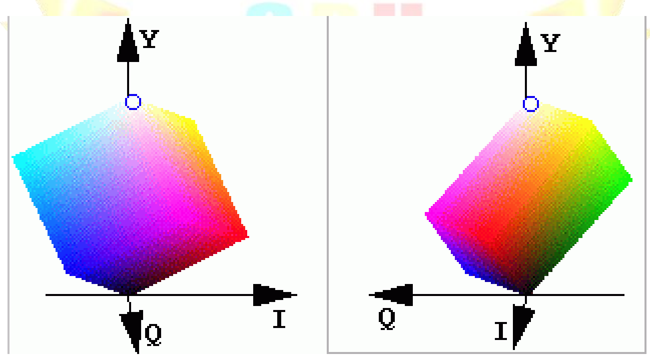


Figure 3.36 YIQ Color Model

CMYK Color Model

It is possible to achieve a large range of colors seen by humans by combining cyan, magenta, and yellow transparent dyes/inks on a white substrate. These are the subtractive primary colors. Often a fourth ink, black, is added to improve reproduction of some dark colors. This is called "CMY" or "CMYK" color space.

The cyan ink absorbs red light but transmits green and blue, the magenta ink absorbs green light but transmits red and blue, and the yellow ink absorbs blue light but transmits red and green. The white substrate reflects the transmitted light back to the viewer. Because in practice the CMY inks suitable for printing also reflect a little bit of color, making a deep and neutral black impossible, the K (black ink) component, usually printed last, is needed to compensate for their deficiencies. Use of a separate black ink is also economically driven when a lot of black content is expected, e.g. in text media, to reduce simultaneous use of the three colored inks. The dyes used in traditional color photographic prints and slides are much more perfectly transparent, so a K component is normally not needed or used in those media.

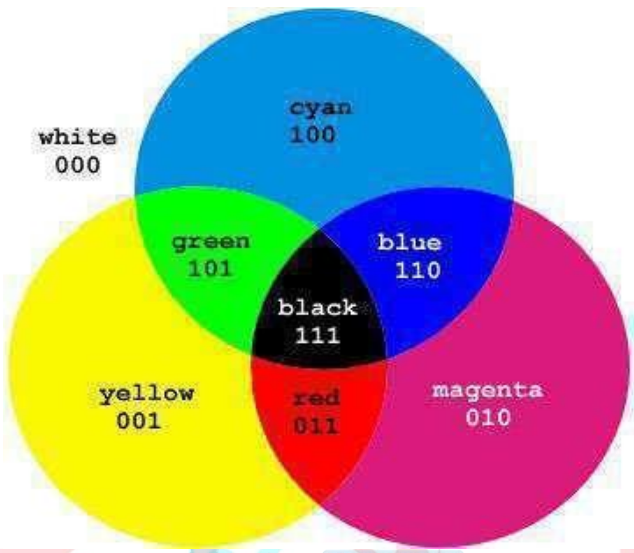


Figure 3.37 CMYK Color Model

HSV Color Model

Recognizing that the geometry of the RGB model is poorly aligned with the color-making attributes recognized by human vision, computer graphics researchers developed two alternate representations of RGB, HSV and HSL (hue, saturation, value and hue, saturation, lightness), in the late 1970s. HSV and HSL improve on the color cube representation of RGB by arranging colors of each hue in a radial slice, around a central axis of neutral colors which ranges from black at the bottom to white at the top. The fully saturated colors of each hue then lie in a circle, a color wheel.

HSV models itself on paint mixture, with its saturation and value dimensions resembling mixtures of a brightly colored paint with, respectively, white and black. HSL tries to resemble more perceptual color models such as NCS or Munsell. It places the fully saturated colors in a circle of lightness ½, so that lightness 1 always implies white, and lightness 0 always implies black.

HSV and HSL are both widely used in computer graphics, particularly as color pickers in image editing software. The mathematical transformation from RGB to HSV or HSL could be computed in real time, even on computers of the 1970s, and there is an easy-to-understand mapping between colors in either of these spaces and their manifestation on a physical RGB device

Recognizing that the geometry of the RGB model is poorly aligned with the color-making attributes recognized by human vision, computer graphics researchers developed two alternate representations of RGB, HSV and HSL (hue, saturation, value and hue, saturation, lightness), in the late 1970s. HSV and HSL improve on the color cube representation of RGB by arranging colors of each hue in a radial slice, around a central axis of neutral colors which ranges from black at the bottom to white at the top. The fully saturated colors of each hue then lie in a circle, a color wheel.

HSV models itself on paint mixture, with its saturation and value dimensions resembling mixtures of a brightly colored paint with, respectively, white and black. HSL tries to resemble more perceptual color models such as

NCS or Munsell. It places the fully saturated colors in a circle of lightness ½, so that lightness 1 always implies white, and lightness 0 always implies black.

HSV and HSL are both widely used in computer graphics, particularly as color pickers in image editing software. The mathematical transformation from RGB to HSV or HSL could be computed in real time, even on computers of the 1970s, and there is an easy-to-understand mapping between colors in either of these spaces and their manifestation on a physical RGB device

Unit-IV

Multimedia is the processing and presentation of information in a more structured and understandable manner using more than one media such as text, graphics, animation, audio and video. Thus multimedia products can be an academic presentation, game or corporate presentation, information kiosk, fashion-designing etc. Multimedia systems are those computer platforms and software tools that support the interactive uses of text, graphics, animation, audio, or motion video. If the sequence and timing of these media elements can be controlled by the user, then one can name it as *Interactive Multimedia*.

A *Multimedia Application* is an Application which uses a collection of multiple media sources e.g. text, graphics, images, sound/audio, animation and/or video.

Hypermedia can be considered as one of the multimedia applications.

Categorization of Multimedia

Linear active content progresses often without any navigational control for the viewer such as a cinema presentation.

Non-linear uses interactivity to control progress as with a video game or self-paced computer based training. Hypermedia is an example of non-linear content.

Multimedia Presentation

A Multimedia Presentation can be described as an intuitive book. Each page of the book can include pictures, animations, sounds and other controls. The pages can turn themselves or wait for their users to click on Next. Unlike the pages of a traditional book, they can talk to their readers too. No matter what your particular need may be, whether to e-mail an electronic photograph album to your relatives, allow users of your web page to download a multimedia catalog, show prospective clients samples of your work or make a dull subject come alive in class, multimedia presentation software can transform your ideas into a complete, professional application. In a live multimedia presentation, the slides are commonly projected onto large screens or printed on overhead transparencies. The slides may be distributed in printed form as handouts to accompany the live presentation. *Multimedia Presentation* has some great features like the ability to:

- Build and distribute slide show style presentations.
- Write your own clickable advertisements, to distribute alone or with other products.
- Design training materials that include pictures, sounds and interactive elements.
- Compile a distributable portfolio.
- Completed presentations are frequently published in multiple formats, which may include print, the Web, or electronic files.

Uses of Multimedia

Multimedia improves information relation. Multimedia applications include the following:

- Medicine
- Entertainment and fine arts
- Business
    - Showing what things look like, how they move and how they change.
- Education
    - Recorded or broadcast lectures.
    - Recording students' performances to enable feedback and promote reflection.
    - Bringing in an expert speaker from a distant location.
- Mathematical and Scientific Research

- Recorded or broadcast conference presentations and discussions.
- Demonstrating new techniques to colleagues.
- Publicizing and promoting research outcomes to related professionals and to the general public.
- Capturing data - such as focus groups, interviews, behavioral observations.

Text Compression

Text is a very big part of most files that digital technology users create. For example, these files could be: Word or PDF documents, emails, cell phone texts SMS format or web pages. Therefore being able to compress text for storage or transmission is extremely important. Fortunately files containing mainly text can be significantly compressed. Like image compression there are many algorithms or methods that have been devised to do this. There is one important point to note about text compression and that is it needs to use a lossless method. This means the method must not discard any data when it compresses the data. If this was so, the data when it is uncompressed would be incomplete. Some techniques used by general purpose compressors such as zip, gzip, bzip2, 7zip, etc, and some types of models of text compression are,

- Static
- Semi adaptive or Semi static
- Adaptive

A static model is a fixed model that is known by both the compressor and the decompressor and does not depend on the data that is being compressed. For example, the frequencies of symbols in English language computed from a large corpus of English texts could be used as the model

A semiadaptive or semistatic model is a fixed model that is constructed from the data to be compressed. or example, the symbol frequencies computed from the text to be compressed can be used as the model. This model has to be included as a part of the compressed data.

An adaptive model changes during the compression. At a given point in compression, the model is a function of the previously compressed part of the data. Since that part of the data is available to the decompressor at the corresponding point in decompression, there is no need to store the model. For example, we could start compressing using a uniform distribution of symbols but then adjust that distribution towards the symbol frequencies in the already processed part of the text.

- Text compression predates most work on general data compression.
- Text compression is a kind of data compression optimized for text (i.e., based on a language and a language model).
- Text compression can be faster or simpler than general data compression, because of assumptions made about the data.
- Text compression assumes a language and language model
- Text compression is effective when the assumptions are met;

Example 1:

Uncompressed text: "I am dumb and because I am dumb, I can't even tell you that I am dumb."

Compressed text: "$1 and because $1, I can't even tell you that $1. $1=[I am dumb]"

Text File Formats

ASCII

The ASCII standard allows ASCII-only text files (unlike most other file types) to be freely interchanged and readable on Unix, Macintosh, Microsoft Windows, DOS, and other systems. These differ in their preferred line ending convention and their interpretation of values outside the ASCII range (their character encoding).

UTF-8

In English context text files can be uniquely ASCII, when in some files are usually 8 bits permissive allowing storage of native texts. In that international context, a Byte Order Mark can appear in start of file to differentiate UTF-8 encoding from legacy regional encoding.

MIME

Text files usually have the MIME type "text/plain", usually with additional information indicating an encoding. Prior to the advent of Mac OS X, the Mac OS system regarded the content of a file (the data fork) to be a text file when its resource fork indicated that the type of the file was "TEXT". Under the Microsoft Windows operating system, a file is regarded as a text file if the suffix of the name of the file (the "extension") is "txt". However, many other suffixes are used for text files with specific purposes. For example, source code for computer programs is usually kept in text files that have file name suffixes indicating the programming language in which the source is written.

.TXT is a file format for files consisting of text usually containing very little formatting (e.g., no bolding or italics). The precise definition of the .txt format is not specified, but typically matches the format accepted by the system terminal or simple text editor. Files with the .txt extension can easily be read or opened by any program that reads text and, for that reason, are considered universal (or platform independent).

The ASCII character set is the most common format for English-language text files, and is generally assumed to be the default file format in many situations. For accented and other non-ASCII characters, it is necessary to choose a character encoding. In many systems, this is chosen on the basis of the default locale setting on the computer it is read on. Common character encodings include ISO 8859-1 for many European languages.

Because many encodings have only a limited repertoire of characters, they are often only usable to represent text in a limited subset of human languages.

Unicode is an attempt to create a common standard for representing all known languages, and most known character sets are subsets of the very large Unicode character set. Although there are multiple character encodings available for Unicode, the most common is UTF-8, which has the advantage of being, backwards-compatible with ASCII: that is, every ASCII text file is also a UTF-8 text file with identical meaning.

Unicode is a computing industry standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems. Developed in conjunction with the Universal Character Set standard and published in book form as The Unicode Standard, the latest version of Unicode contains a repertoire of more than 110,000 characters covering 100 scripts. The standard consists of a set of code charts for visual reference, an encoding method and set of standard character encodings, a set of reference data computer files, and a number of related items, such as character properties, rules for normalization, decomposition, collation, rendering, and bidirectional display order (for the correct display of text containing both right-to-left scripts, such as Arabic and Hebrew, and left-to-right scripts). As of September 2013, the most recent version is Unicode 6.3. The standard is maintained by the Unicode Consortium.

Unicode can be implemented by different character encodings. The most commonly used encodings are UTF-8, UTF-16 and the now-obsolete UCS-2. UTF-8 uses one byte for any ASCII characters, which have the same code values in both UTF-8 and ASCII encoding, and up to four bytes for other characters. UCS-2 uses a 16-bit code unit (two 8-bit bytes) for each character but cannot encode every character in the current Unicode standard. UTF-16 extends UCS-2, using two 16-bit units (4 × 8 bit) to handle each of the additional characters.

Unicode Transformation Format and Universal Character Set

Unicode defines two mapping methods: the Unicode Transformation Format (UTF) encodings, and the Universal Character Set (UCS) encodings. An encoding maps (possibly a subset of) the range of Unicode code points to sequences of values in some fixed-size range, termed code values. The numbers in the names of the encodings indicate the number of bits in one code value (for UTF encodings) or the number of bytes per code value (for UCS) encodings. UTF-8 and UTF-16 are probably the most commonly used encodings. UCS-2 is an obsolete subset of UTF-16; UCS-4 and UTF-32 are functionally equivalent.

UTF encodings include:

1. UTF-1 – a retired predecessor of UTF-8, maximizes compatibility with ISO 2022, no longer part of The Unicode Standard
2. UTF-7 – a 7-bit encoding sometimes used in e-mail, often considered obsolete (not part of The Unicode Standard, but only documented as an informational RFC, i.e. not on the Internet Standards Track either)
3. UTF-8 – an 8-bit variable-width encoding which maximizes compatibility with ASCII.
4. UTF-EBCDIC – an 8-bit variable-width encoding similar to UTF-8, but designed for compatibility with EBCDIC. (Not part of The Unicode Standard)
5. UTF-16 – a 16-bit, variable-width encoding
6. UTF-32 – a 32-bit, fixed-width encoding
7. There are basically four ways to encode Unicode characters in bytes:

UTF−8

128 characters are encoded using 1 byte (the ASCII characters). 1920 characters are encoded using 2 bytes (Roman, Greek, Cyrillic, Coptic, Armenian, Hebrew, Arabic characters). 63488 characters are encoded using 3 bytes (Chinese and Japanese among others). The other 2147418112 characters (not assigned yet) can be encoded using 4, 5 or 6 characters. For more info about UTF−8, do `man 7 utf−8' (manage contained in the man−pages−1.20 package).

UCS−2

Every character is represented as two bytes. This encoding can only represent the first 65536 Unicode characters.

UTF−16

This is an extension of UCS−2 which can represent 1112064 Unicode characters. The first 65536 Unicode characters are represented as two bytes, the other ones as four bytes.

UCS−4

Every character is represented as four bytes. The space requirements for encoding a text, compared to encodings currently in use (8 bit per character for European languages, more for Chinese / Japanese / Korean), is as follows. This has an influence on disk storage space and network download speed (when no form of compression is used).

UTF−8

No change for US ASCII, just a few percent more for ISO−8859−1, 50% more for Chinese / Japanese / Korean, 100% more for Greek and Cyrillic.

Text Files Format	Description / Properties	Usage and Archival Recommendations
.txt	Text file Simple plain text document Compatible across software packages Supports very little formatting	Good for extremely simple files Commonly used for introductory “‘ead me” files containing basic information on project archives

.doc	Microsoft Word document (-2003) Proprietary binary format Can be read by OpenOffice Easily converted into PDF format	Accepted for archiving because it is sowidely used However, will become obsolete
.docx	Microsoft Word document (2007) Human readable XML format Stored along with embedded content as zipped file	Good for dissemination and preservation Conversion to .doc file to open with earlierversions of MS Word
.rtf	Rich	Formatting issues when using opened indifferent software Large file sizes mean that .docx or. odt fileformats are preferred

	Text Format (Microsoft) Tagged plain text format	

.odt	Open Document Text (OpenOffice) ISO standard, human readable XMLformat	Open source format good for use,dissemination and archiving Archive files in uncompressed formCan open .doc files Might not open correctly in other wordprocessing programs
.pdf	Portable Document Format (Adobe) Proprietary binary format Aims to retain document formatting Can store embedded data: raster and vector images (e.g. Adobe Illustrator files)	Highly suitable for dissemination PDF creators and readers freely and widely available Retain original text document and embedded objects (e.g. images, tabulardata, etc)
PDF/A	Portable Document Format / Archive (Adobe) Open ISO standard format for long-	Widely accepted as viable format for long-term archiving Retain original text file and embedded objects separately (e.g. images, tabulardata, etc)

	term archiving Formatting data self-contained in file	

Audio Components of an audio system

A sound system is always composed of four main parts:

Part	Function	Examples
Sound source	Provide the music	Mp3 player, computer, microphone
Mixer	Joins two or more sources together	Mixing desk
Amplifier	Boosts the signal from the mixer	Amplifier, powered desk
Speakers	Provide the sound for the audience	Passive or active speakers

Digital Audio

Digital audio is a technology that can be used for sound recording and reproduction using audio signals that have been encoded in digital form.

- Audio comes from different sources:
  - Speech.
  - Sounds of instruments, Music.
  - Sounds of all other kinds (the sound of wind, train and ocean).
- Audio needs new methods for coding and processing.
- Audio processing is a key task in multimedia systems
  - Audio coding (MPEG audio, mp3, AAC and others)
  - Authoring and representation (composition)
  - Analysis and searching (retrieval and database)
  - 3D sound, etc.

In a digital audio system, a microphone converts sound to an analog electrical signal, then an analog-to-digital converter (ADC)—typically using pulse-code modulation—converts the analog signal into a digital signal. This digital signal can then be recorded, edited and modified using digital audio tools. When the sound engineer wishes to listen to the recording on headphones or loudspeakers (or when a consumer wishes to listen to a digital sound file of a song), a digital-to-analog converter (DAC) performs the reverse process, converting a digital signal back into an analog signal, through an audio power amplifier and send to a loudspeaker.

Digital audio systems may include compression, storage, processing and transmission components. Conversion to a digital format allows convenient manipulation, storage, transmission and retrieval of an audio signal. Unlike analog audio, in which making copies of recording results in generation loss, a degradation of the signal quality, when using digital audio, an infinite number of copies can be made without any degradation of signal quality.

Digital audio processing

Audio signal processing or audio processing is the intentional alteration of audio signals often through an audio effect or effects unit. As audio signals may be electronically represented in either digital or analogformat, signal processing may occur in either domain. Analog processors operate directly on the electrical signal, while digital processors operate mathematically on the digital representation of that signal.

The five general steps in digital audio signal processing are as follows:

1. Signal generation
2. Signal amplification and processing
3. Signal transmission
4. Signal reception
5. Signal storage.

Step 1 Signal Generation: Signal generation is the conversion of the sound waves from the source into electrical energy which corresponds to the frequency of the original source. The audio signal may be generated mechanically using a microphone or turntable to create an analog of the original sound signal such as a phonograph record or audio cassette. Microphones are used to transduce the physical energy of music and voice into electrical energy. The audio signal may be generated electromagnetically using tape recorders. The audio signal may also be generated digitally by using laser optics to create a binary or digital equivalent of the original sound. Television signal generation requires electronic line-by-line scanning of an image using an electron beam to scan each element of the picture. The image is subsequently retraced by the television receiver.

Step 2 Signal Amplification and Processing: Audio signals are amplified and mixed using audio consoles. After the audio signal has been converted from a physical sound wave into an electrical or digital facsimile, the audio signal must be amplified to boost the signal and processed including the mixing, combining and routing for broadcast transmission and/or recording. Sound sources are combined at the mixing board. The amplified sound may be fine tuned using equalizers and special effects. The switcher is used to mix TV signals and put the desired picture on the air. A special effect generator is used to add transitions, split screen and keying.

Digital video editing and effects can also be produced using computer software such as Adobe Premiere Pro and After Effects.

Step 3 Signal Transmission: The electronic signal is superimposed by a modulation process on a carrier wave generated or propagated by the radio station on its assigned frequency. The generated sound wave may travel by ground, sky and direct waves. Radio waves occupy a segment of the electromagnetic spectrum. AM radio channel frequencies are divided into three main types: clear channels, regional channels and local channels. FM channel frequencies are classified by antenna height and power. Stereo broadcasting and other non broadcast services are accomplished with the wide bandwidth of the FM channel. Digital radio is satellite-based or in-band on channel. Television signal transmission includes over-the-air broadcasting using the electromagnetic radiation on the VHF and UHF portions of the spectrum or by wire through a cable system using coaxial cable that can carry programming on more than 100 channels. New transmission technologies used for transmission and distribution include satellite and fiber optics for digital signals.

Step 4 Signal Reception: After the radio signal has been transduced, modulated and transmitted, the radio waves are picked up on a radio receiver where they are transduced or converted by the speaker system back into sound waves. The characteristics of the electromagnetic spectrum and modulation method used in transmission determine the type of radio receiver needed to convert the signal back into sound waves. There are several types of radio receivers including AM, AM stereo, FM, shortwave, and multiband. These receivers can be equipped with either analog tuners or a digital system. For moving images both large and small-screen TVs are now receiving high-definition television vision signals.

Step 5 Signal Storage: Audio technology is used in the storage and retrieval of sounds. Audio signals are transduced or converted for storage and eventual playback or rebroadcast. The storage medium have included glass discs, wire, vinyl, magnetic type, compact disc, video tapes, digital storage media such as digital versatile discs (DVDs) and computer hard drives including high-capacity disc drives.

Sound cards

A sound card (also known as an audio card) is an internal expansion card that provides input and output of audio signals to and from a computer under control of computer programs. The term sound card is also applied to external audio interfaces used for professional audio applications. Typical uses of sound cards include providing the audio component for multimedia applications such as music composition, editing video or audio, presentation, education and entertainment (games) and video projection.

Sound functionality can also be integrated onto the motherboard, using components similar to those found on plug-in cards. The integrated sound system is often still referred to as a sound card. Sound processing hardware is also present on modern video cards with HDMI to output sound along with the video using that connector; previously they used a SPDIF connection to the motherboard or sound card.

Image Creation and Capture Image	Format options	Archive Recommendations
Digital Cameras	Dependent on model of camera	1. Raw DNG (or TIFF) file if possible 2. Original JPEG: save archive copy on download and for presentation images always work on a copy of the file
Scanners	Wide range once scanned	Save uncompressed/lossless format (TIFF) as archive copy regardless of intended format
Graphics Images	Wide choice of formats under 'Save As…' command	Alongside software package files (e.g Photoshop [.psd], Corel Draw [.cpt]), save draft images in uncompressed TIFF format if possible, and replace with archive TIFF of end product image

Audio file formats

Audio files come in all types and sizes. We are familiar with MP3, but what about AAC, FLAC, OGG, or WMA? Why do so many standards exist? Which ones should you care about and which ones can you ignore?

It's actually quite simple once you realize that all audio formats fall into three major categories. Once you know which category you want, all you have to do is pick the format within that category that best suits your needs.

File extension	Format	Codec	Note
.mp3	MPEG	MPEG1 Audio Layer 3	
.m4a .mpa .aac	MPEG 4	AAC	
.flac	FLAC	FLAC	Supports up to 2 channels
.ogg	OGG	Vorbis	Supports up to 2 channels
.wma	WMA	WMA	Supports up to 10 Pro 5.1 channels. WMA lossless audio is not supported. Supports up to the M2 profile.
.wav	Wav	wav	
.mid .midi	Midi	midi	Supports type 0 and type 1. Seek is not supported. Supports USB device only.
.ape	Ape	ape	
.aif .aiff	AIFF	AIFF	
.m4a	ALAC	ALAC	

Uncompressed Audio Formats

Uncompressed audio is exactly what it sounds like: real sound waves that have been captured and converted to digital format without any further processing. As a result, uncompressed audio files tend to be the most accurate but take up a LOT of disk space — about 34 MB per minute for 24-bit 96 KHz stereo.

PCM

PCM stands for Pulse-Code Modulation, a digital representation of raw analog audio signals. Analog sounds exist as waveforms, and in order to convert a waveform into digital bits, the sound must be sampled and recorded at certain intervals (or pulses).

As such, this digital audio format has a "sampling rate" (how often a sample is made) and a "bit depth" (how many bits are used to represent each sample). There is no compression involved. The digital recording is a close-to-exact representation of the analog sound.

PCM is the most common audio format used in CDs and DVDs. There is a subtype of PCM called Linear Pulse-Code Modulation, where samples are taken at linear intervals. LPCM is the most common form of PCM, which is why the two terms are almost interchangeable at this point.

WAV

WAV stands for Waveform Audio File Format (also called Audio for Windows at some point but not anymore). It's a standard that was developed by Microsoft and IBM back in 1991.

A lot of people assume that all WAV files are uncompressed audio files, but that's not exactly true. WAV is actually just a Windows container for audio formats. This means that a WAV file can contain compressed audio, but it's rarely used for that.

Most WAV files contain uncompressed audio in PCM format. The WAV file is just a wrapper for the PCM encoding, making it more suitable for use on Windows systems. However, Mac systems can usually open WAV files without any issues.

AIFF

AIFF stands for Audio Interchange File Format. Similar to how Microsoft and IBM developed WAV for Windows, AIFF is a format that was developed by Apple for Mac systems back in 1988.

Also similar to WAV files, AIFF files can contain multiple kinds of audio. For example, there is a compressed version called AIFF-C and another version called Apple Loops which is used by GarageBand and Logic Audio — and they all use the same AIFF extension.

Most AIFF files contain uncompressed audio in PCM format. The AIFF file is just a wrapper for the PCM encoding, making it more suitable for use on Mac systems. However, Windows systems can usually open AIFF files without any issues.

Lossy Compressed Audio Formats

Lossy compression is a form of compression that loses data during the compression process. In the context of audio, that means sacrificing quality and fidelity for file size. The good news is that, in most cases, you won't be able to hear the difference. However, if the audio gets compressed too much or too often, you'll start hearing artifacts and other weirdness's that become more and more noticeable.

MP3

MP3 stands for MPEG-1 Audio Layer 3. It was released back in 1993 and quickly exploded in popularity, eventually becoming the most popular audio format in the world for music files. There's a reason why we have

"MP3 players" but not "OGG players"…

The main pursuit of MP3 is to cut out all of the sound data that exists beyond the hearing range of most normal people and to reduce the quality of sounds that aren't as easy to hear, and then to compress all other audio data as efficiently as possible.

Nearly every digital device in the world with audio playback can read and play MP3 files, whether we're talking about PCs, Macs, Androids, iPhones, Smart TVs, or whatever else. When you need universal, MP3 will never let you down.

AAC

AAC stands for Advanced Audio Coding. It was developed in 1997 as the successor to MP3, and while it did catch on as a popular format to use, it never really overtook MP3 as the most popular for everyday music and recording.

The compression algorithm used by AAC is much more advanced and technical than MP3, so when you compare a particular recording in MP3 and AAC formats at the same bitrate, the AAC one will generally have better sound quality.

Again, even though MP3 is more of a household format, AAC is widely used today. In fact, it's the standard audio compression method used by YouTube, Android, iOS, iTunes, later Nintendo portables, and later PlayStations.

OGG (Vorbis)

OGG doesn't stand for anything. Actually, it's not even a compression format. OGG is a multimedia container that can hold all kinds of compression formats, but is most commonly used to hold Vorbis files — hence why these audio files are called Ogg Vorbis files.

Vorbis was first released in 2000 and grew in popularity due to two reasons: first, it adheres to the principles of open source software, and second, it performs significantly better than most other lossy compression formats (i.e. produces a smaller file size for equivalent audio quality).
MP3 and AAC have such strong footholds that OGG has had a hard time breaking into the spotlight — not many devices support it natively— but it's getting better with time. For now, it's mostly used by hardcore proponents of open software.

WMA

WMA stands for Windows Media Audio. It was first released in 1999 and has gone through several evolutions since then, all while keeping the same WMA name and extension. As you might expect, it's a proprietary format created by Microsoft.
Not unlike AAC and OGG, WMA was meant to address some of the flaws in the MP3 compression method — and as such, WMA's approach to compression is pretty similar to AAC and OGG. In other words, in terms of objective quality, WMA is better than MP3.
But since WMA is proprietary, not many devices and platforms support it. It also doesn't offer any real benefits over AAC or OGG, so in most cases when MP3 isn't good enough, it's simply more practical to go with one of those two instead.

Lossless Compressed Audio Formats

On the other side of the coin is lossless compression, which is a method that reduces file size without any loss in quality between the original source file and the resulting file. The downside is that lossless compression isn't as efficient as lossy compression, meaning equivalent files can be 2x to 5x larger.This is obviously much harder to do well, but there are a few good formats for this. And don't confuse lossless compression with high-resolution audio (which is most likely a scam anyway).

FLAC

FLAC stands for Free Lossless Audio Codec. A bit on the nose maybe, but it has quickly become one of the most popular lossless formats available since its introduction in 2001.
What's nice is that FLAC can compress an original source file by up to 60% without losing a single bit of data. What's even nicer is that FLAC is an open source and royalty-free format rather than a proprietary one, so it doesn't impose any intellectual property constraints.
FLAC is supported by most major programs and devices and is the main alternative to MP3 for CD audio. With it, you basically get the full quality of raw uncompressed audio in half the file size — what's not to love about it?

ALAC

ALAC stands for Apple Lossless Audio Codec. It was developed and launched in 2004 as a proprietary format but eventually became open source and royalty-free in 2011. ALAC is sometimes referred to as Apple Lossless.
While ALAC is good, it's slightly less efficient than FLAC when it comes to compression. However, Apple users don't really have a choice between the two because iTunes and iOS both provide native support for ALAC and no support at all for FLAC.

WMA

WMA stands for Windows Media Audio. We covered it above in the lossy compression section, but we mention it here because there's a lossless alternative called WMA Lossless that uses the same extension. Confusing, I know.

Compared to FLAC and ALAC, WMA Lossless is the worst in terms of compression efficiency but only slightly. It's a proprietary format so it's no good for fans of open source software, but it is supported natively on both Windows and Mac systems.

The biggest issue with WMA Lossless is the limited hardware support. If you want lossless audio across multiple devices, you should stick with FLAC unless all of your devices are of the Windows variety.

So Which Format Should You Use?

Video and Audio Files Format	Notes
.wav	Waveform Audio (Microsoft) Uncompressed file Recommended for long term preservation

For most people, the decision is actually pretty easy: If you're capturing and editing raw audio, use an uncompressed format. This way you're working with thetruest quality of audio possible. When you're done, you can export to a compressed format.

If you're listening to music and want faithful audio representation, use lossless audio compression. This is why audiophiles always scramble for FLAC albums over MP3 albums. Note that you'll need more storage space for these.

If you're okay with "good enough" music quality, if your audio file doesn't have any music, or if you need to conserve disk space, use lossy audio compression. Most people actually can't hear the difference between lossy and lossless compression.

For those who want utmost quality in their music playback, note that high-quality audio files won't matter if your playback device can't faithfully recreate those sounds. Meaning, you need to have good quality speakers or good quality headphones!

.aif	Audio Interchange File (Apple) Uncompressed file Recommended for long term preservation
.mp3	MPEG1, 2 Audio Layer 3 (Moving Picture Expert Group: Audio group) Patented ISO standard compressed format
.rm / .ram	Read Audio file format used for streaming radio over the internet
.wma	Windows Media Audio Compressed file used by Windows
.ogg	Open standard format for compressed audio files
.avi	Audio Video Interleave (Microsoft)
.wmv	Windows Media Video (Microsoft) Proprietary compression format for hard media delivery (DVD, Blu-Ray)
.mov	QuickTime File Format (Apple)
.mp4	MPEG4 – Digital Video File Format ISO standard Recommended by DSpace for long term storage

Audio Processing software

Software to record and edit audio is often referred to as a digital audio editor. High-end systems forprofessionals are also called digital audio workstations.

Audio can consist of only spoken text, such as the narration of this video, or include music and other sound effects. A digital audio editor typically contains functions for the following tasks:

- Record audio from a variety of different input sources
- Mix multiple tracks into one output track
- Apply effects, such as noise reduction and equalization, to improve the quality
- Edit sound clips by cutting out certain parts, switching the order of clips, etc.
- Convert between a variety of different formats

Some of the most widely used audio software applications include Audacity by the Audacity Team, Audition by Adobe, Garage Band by Apple, ProTools by Avid, WavePad, FL Studio 12.1.2, Ardour, Wavosaur, Sound Forge Audio Studio 10, Steinberg Cubase, Linux MultiMedia Studio (LMMS), PreSonus Studio One, - but there are many others.

Description of some audio processing software's:

Audacity

First released in 2000, Audacity has gone on to become one of the most popular pieces of free software around, and for good reason. The open-source program gives users a wide range of options for recording and editing audio, all tied together with a simple interface. Perhaps most importantly, unlike many other free programs, Audacity is not "free for a limited time" or "free with many features locked." Everything Audacity has to offer is free of charge.

Ardour

Ardour is an open-source DAW designed for Linux and available for Mac OS and Windows. Its developer, Paul Davis, also invented the JACK sound server for Linux and worked previously as one of the original programmers at Amazon. Ardour features highly versatile multi-track recording features that include the ability to import video for film scoring purposes, to record and edit either non-destructively or destructively, and to prepare any combination of individual tracks for recording. Its use of JACK makes it compatible with a number of outside applications, as well. However, unlike other DAWs, Ardour does not come with any built-in effects or instruments and relies instead on the installation of third-party software.

Cubase 8.5

Cubase has been around in some form since 1989. Along the way, their parent company Steinberg has been responsible for some DAW innovations and concepts, like VST plugin architecture, that have been widely adopted in the industry. Cubase 8.5, accordingly, is a bit like Logic Pro X — doing a bit of everything quite well — but with its own feel and flavor. Cubase is the MVP on your production team. It can handle pro-level audio and mixing work in a similar fashion to Pro Tools, but has also invested significant effort over the past few releases to balance that with a pretty sweet creative workflow. The MIDI and plugin capabilities of Cubase are second to none, making it a really good choice for musos who want to explore both their compositional and engineering sides. There's even some pretty strong notation capabilities that make it a solid candidate for orchestration, transcription, and film composition. One of the most interesting features of Cubase 8.5 is their new cloud collaboration service, built specifically to handle long-distance collaborative projects, including the sharing of plugins via a service called VST Transit.

Garage Band

Arguably the most popular DAW ever created, Garage Band comes free with every Macintosh computer and boasts a simple, clean interface. The last couple of versions have seen Garage Band come to resemble a diet version of Apple's flagship DAW, Logic (discussed below). It's both loved for its simplicity and loathed for its

limitations. Case in point: Claire Boucher, also known as Grimes, recorded her breakthrough album Visions using nothing but Garage Band. That is, before later renouncing the platform as far too limited.
Simplicity is the major selling point here. Garage Band takes the complex world of music production and streamlines it into a basic, welcoming package with a gentle learning curve. There's enough in the way of beats, drum samplers, software instruments, and effects to keep one occupied for a while. The simplicity of the software also means work happens quickly. You can hear basic results immediately, enjoy the vibe, and keep working.
Garage Band can get you where you need to be quickly. Where a fast, simple, and no-frills solution is called for (and maybe one that also works on your iPhone), Garage Band is going to deliver the goods. It did for Rihanna with "Umbrella," and for the demos that got Cloud Nothings signed to labels.

Pro Tools
The undeniable industry standard, Pro Tools is the lumbering DAW behemoth by which everything else is judged. It's been designed from the ground up to be the standard for large commercial recording studios and media production houses, and with a vintage of 27 years, it's definitely had time to establish its majority market share. Many swear by it, but in past years it's come under criticism for perceived issues with creative workflow and the strict licensing and commercial practices of its parent company, Avid. Pro Tools was designed from the ground up to work with rack-mounted studio gear, large mix consoles, and complex patch routings. If your aim is to work with these kinds of things, there's a lot of capability here to help you. As Pro Tools pioneered many innovations in non-destructive editing, take-comping, and bus-routing, there's serious depth below the surface. From a pure audio recording and manipulation standpoint, it's hard to beat. There's also the question of hardware. Avid has created a whole line of audio hardware — from consumer-grade interfaces to the highest pro-quality desks — designed to work specifically with Pro Tools. If you love the program and can afford the hefty prices, this could be seen as a long-term investment in your career.

Video
With latest technology it is possible to include video impact on clips of any type into any multimedia creation, be it corporate presentation, fashion design, entertainment games, etc.
The video clips may contain some dialogues or sound effects and moving pictures. These video clips can be combined with the audio, text and graphics for multimedia presentation. Incorporation of video in a multimedia package is more important and complicated than other media elements. One can procure video clips from various sources such as existing video films or even can go for an outdoor video shooting.
The entire video available are in analog format. To make it usable by computer, the video clips are needed to be converted into computer understandable format, i.e., digital format. Both combinations of software and hardware make it possible to convert the analog video clips into digital format. This alone does not help, as the digitised video clips take lots of hard disk space to store, depending on the frame rate used for digitisation.
The computer reads a particular video clip as a series of still pictures called *frames*. Thus video clip is made of a series of separate frames where each frame is slightly different from the previous one. The computer reads each frame as a bitmap image. Generally there are 15 to 25 frames per second so that the movement is smooth. If we take less frames than this, the movement of the images will not be smooth.
To cut down the space there are several modern technologies in windows environment. Essentially these technologies compress the video image so that lesser space is required.
However, latest video compression software makes it possible to compress the digitised video clips to its maximum. In the process, it takes lesser storage space. One more advantage of using digital video is, the quality of video will not deteriorate from copy to copy as the digital video signal is made up of digital code and not electrical signal. Caution should be taken while digitizing the video from analog source to avoid frame droppings and distortion. A good quality video source should be used for digitization.

Video Color Spaces

A standard that defines a limited set of colors that a particular technology is able to describe, with maximum Red, Green and Blue extents that are mapped to sit inside the full CIE XYZ space. The space within the full CIE XYZ space that a particular color space covers is called its Gamut.

While the full XYZ color space is used for DCP (Digital Cinema Package) encoding, you are far more likely to come across these most common color spaces.

Commonly used color spaces

Most photographers will have to deal with a handful of different color spaces in their work. In general, these can be divided into three groups: Working, Device and Output spaces. Let's take a look at each group and list some of the options you'll see in each.

Working spaces

Working color spaces are color models that are well suited to image editing tasks such as color and tone adjustments. Ideally, these are large color spaces, offering the photographer the ability to choose between a wide gamut of colors. There's no single ideal color space but there are a few very good choices.

- Adobe RGB (1998) - As the name implies, this color space was created by Adobe in the late 90s, when Photoshop implemented full color management. Although Adobe never intended this color space to be the universal standard, it is widely supported. Most DSLR cameras offer this as a color space choice for JPEG creation. While Adobe RGB does not contain as many colors as ProPhotoRGB, it's easier to use and a very good choice for both 8-bit and 16-bit image editing.
- ProPhoto RGB - This color space was designed as a universal standard for high-bit image editing, and includes all the colors that the human eye can see. ProPhoto RGB is a very popular color space for experienced Photoshop users. Because the space is so wide, it's not appropriate for 8-bit images.
- ProPhoto RGB is not always the best choice for a working space. Because 15% of the color space is beyond the range of human vision, color mapping is happening in ways that are impossible to see. Some images will suffer when they are converted from ProPhoto RGB to a CMYK space because these out-of-gamut colors map to the destination space in unwanted ways.
- CIELAB - This color space is used internally by Photoshop during color space conversions. Some people have found interesting ways to use CIELAB space to manipulate images, since the luminance is totally separate from the color information. However, CIELAB is not an easy space to use since it's not intuitive for most people. Images that are in CIELAB space also don't have full support in Photoshop for adjustment layers and other non-destructive imaging tools.

Device-dependent spaces

Photographers will most commonly run into device-dependent color spaces when profiling monitors, desktop printers or sending images out for CMYK printing.

- Monitor RGB - Modern monitors include a factory-created profile that is loaded into the monitor's firmware and is communicated to your computer via the monitor connection cable. If you want your monitor to do the best possible job reproducing your images, you should create a custom profile for it.
- Desktop printer profile - Your printer comes with profiles in the driver software. You can create a custom profile for your printer if you want to maximize its color fidelity.

Delivery spaces

When images are sent from one person to another, it can be appropriate to consider converting the working space. The choice of a delivery space is often dependent on what the photographer knows about the recipient of the image. In some cases, much is known about the recipient, or the exact delivery space is precisely specified. Other times, images get sent out with little indication of how they will be treated on the other end.

- sRGB - This color space is a small color space – it's often thought of as a lowest common denominator. sRGB is very similar to older monitor spaces and, in fact, it's common for unmanaged computers to assume that an image is in the monitor color space. This makes sRGB a good choice to send to unknown users. At the moment, sRGB is the only appropriate choice for images uploaded to the web since most web browsers don't support any color management. Additionally, sRGB is a very good choice for images sent to minilabs, especially if there is no custom profile available. Because sRGB is not a wide color space, it's not appropriate as a working space.

- Adobe RGB - This is probably the most often-requested color space for delivery, if a color space is specified. It offers a good gamut and very wide support. Note that Adobe RGB images that are uploaded to websites without conversion to sRGB will generally appear dark and muted.
- CMYK profile - When you send an image off for CMYK printing, you can either use a generic CMYK profile, or, if the printer can supply one, you can convert the image to a custom color space.

Digital Video

Digital video is a representation of moving visual images in the form of encoded digital data. This is in contrast to analog video, which represents moving visual images with analog signals. Digital video comprises a series of digital images displayed in rapid succession.

Digital Video Processing

Video processing is a particular case of signal processing, which often employs video filters and where the input and output signals are video files or video streams. Video processing techniques are used in television sets, VCRs, DVDs, video codecs, video players, video scalers and other devices. For example—commonly only design and video processing is different in TV sets of different manufactures.

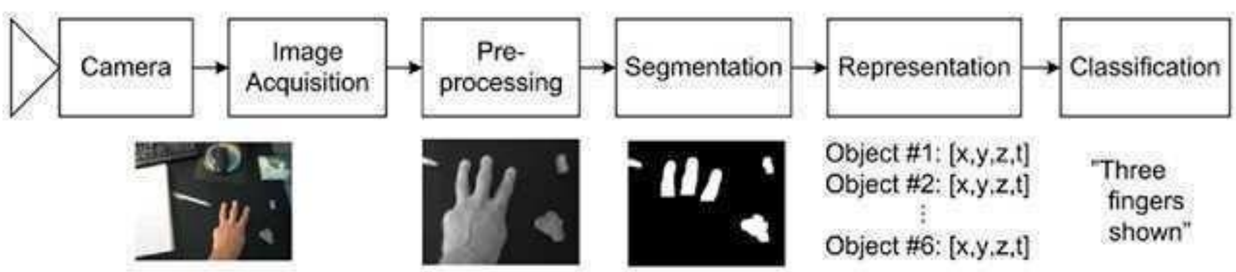


Figure 4.1 The block diagram provides a general framework for video and images

Image Acquisition In this step everything to do with the camera and setup of your system is covered, e.g., camera type, camera settings, optics, and light sources.

Pre-processing This step does something to your image / video before the actual processing commences, e.g., convert the image from color to gray-scale or crop the most interesting part of the image (as seen in Fig. above).

Segmentation This is where the information of interest is extracted from the image or video data. Often this block is the "heart" of a system. In the example in the figure the information is the fingers. The image below the segmentation block shows that the fingers (together with some noise) have been segmented (indicated by white objects).

Representation In this block the objects extracted in the segmentation block are represented in a concise manner, e.g., using a few representative numbers as illustrated in the figure.

Classification Finally this block examines the information produced by the previous block and classifies each object as being an object of interest or not. In the example in the figure this block determines that three finger objects are present and hence output this.

It should be noted that the different blocks might not be as clear-cut defined in reality as the figure suggests. One designer might place a particular method in one block while another designer will place the same method in the previous or following block. Nevertheless the framework is an excellent starting point for any video and image processing system. The last two blocks are sometimes replaced by one block called BLOB Analysis. This is especially done when the output of the segmentation block is a black and white image as is the case in the figure. In this topic we follow this idea and have therefore merged the descriptions of these two blocks into one—BLOB Analysis.

Video File Formats

The MPEG standards consist of different Parts. Each part covers a certain aspect of the whole specification. The standard also specifies Profiles and Levels. Profiles are intended to define a set of tools that are available, and Levels define the range of appropriate values for the properties associated with them.

MPEG-4 (.mp4)	Also MPEG-4 part 14 or mp4, a digital multimedia format used to store video and audio, but also store subtitles and still images. It will be the best video format for IMovie, in case you want a smaller video file size.
Quick Time Movie (.mov)	This file format specifies a multimedia container file containing one or more tracks, each of which stores a particular type of data: audio, video, effects, or text.
MPEG-2	aka H.222/H.262 as defined by the ITU. It is the core of most digital television and DVD formats.
AVCHD	Short for Advanced Video Coding High Definition, it is a file based format for the digital recording and playback of high definition video.
DV & HDV	The format for storing digital video recorded by camcorder. If you want video for better quality, believe it will be the best video format for IMovie as it is uncompressed.

MPEG has standardized the following compression formats and ancillary standards:

MPEG-1: is the first compression standard for audio and video. It was basically designed to allow moving pictures and sound to be encoded into the bitrate of a Compact Disc. To meet the low bit requirement, MPEG-1 downsamples the images, as well as using picture rates of only 24-30 Hz, resulting in a moderate quality. It includes the popular Layer 3 (MP3) audio compression format.

MPEG-2: Transport, video and audio standards for broadcast-quality television. MPEG-2 standard was considerably broader in scope and of wider appeal--supporting interlacing and high definition. MPEG-2 is considered important because it has been chosen as the compression scheme for over-theair digital television ATSC, DVB and ISDB, digital satellite TV services like Dish Network, digital cable television signals, SVCD, and DVD.

MPEG-3: Developments in standardizing scalable and multi-resolution compression which would have become MPEG-3 were ready by the time MPEG-2 was to be standardized; hence, these were incorporated into MPEG-2 and as a result there is no MPEG-3 standard. MPEG-3 is not to be confused with MP3, which is MPEG-1 Audio Layer 3.

MPEG-4: MPEG-4 uses further coding tools with additional complexity to achieve higher compression factors than MPEG-2. In addition to more efficient coding of video, MPEG-4 moves closer to computer graphics applications. In more complex profiles, the MPEG-4 decoder effectively becomes a rendering processor and the compressed bitstream describes three-dimensional shapes and surface texture. MPEG-4 also provides Intellectual Property Management and Protection (IPMP) which provides the facility to use proprietary technologies to manage and protect content like digital rights management.

Several new higher-efficiency video standards (newer than MPEG-2 Video) are included (an alternative to MPEG-2 Video), notably:

- MPEG-4 Part 2 (or Advanced Simple Profile) and
- MPEG-4 Part 10 (or Advanced Video Coding or H.264). MPEG-4 Part 10 may be used on HD DVD and Blu-ray discs, along with VC-1 and MPEG-2.

In addition, the following standards, while not sequential advances to the video encoding standard as with MPEG-1 through MPEG-4, are referred to by similar notation:

- MPEG-7: A multimedia content description standard.
- MPEG-21: MPEG describes this standard as a multimedia framework.

Unit-V

Animation

Animation means giving life to any object in computer graphics. It has the power of injecting energy and emotions into the most seemingly inanimate objects. Computer-assisted animation and computer-generated animation are two categories of computer animation. It can be presented via film or video.

The basic idea behind animation is to play back the recorded images at the rates fast enough to fool the human eye into interpreting them as continuous motion. Animation can make a series of dead images come alive. Animation can be used in many areas like entertainment, computer aided-design, scientific visualization, training, education, e-commerce, and computer art.

Animation Techniques

Animators have invented and used a variety of different animation techniques. Basically there are six animation techniques which we would discuss one by one in this section.

Traditional Animation (frame by frame)

Traditionally most of the animation was done by hand. All the frames in an animation had to be drawn by hand. Since each second of animation requires 24 frames (film), the amount of efforts required to create even the shortest of movies can be tremendous.

Key framing

In this technique, a storyboard is laid out and then the artists draw the major frames of the animation. Major frames are the ones in which prominent changes take place. They are the key points of animation. Key framing requires that the animator specifies critical or key positions for the objects. The computer then automatically fills in the missing frames by smoothly interpolating between those positions.

Procedural

In a procedural animation, the objects are animated by a procedure − a set of rules − not by key framing. The animator specifies rules and initial conditions and runs simulation. Rules are often based on physical rules of the real world expressed by mathematical equations.

Behavioral

In behavioural animation, an autonomous character determines its own actions, at least to a certain extent. This gives the character some ability to improvise, and frees the animator from the need to specify each detail of every character's motion.

Performance Based (Motion Capture)

Another technique is Motion Capture, in which magnetic or vision-based sensors record the actions of a human or animal object in three dimensions. A computer then uses these data to animate the object.

This technology has enabled a number of famous athletes to supply the actions for characters in sports video games. Motion capture is pretty popular with the animators mainly because some of the commonplace human actions can be captured with relative ease. However, there can be serious discrepancies between the shapes or dimensions of the subject and the graphical character and this may lead to problems of exact execution.

Physically Based (Dynamics)

Unlike key framing and motion picture, simulation uses the laws of physics to generate motion of pictures and other objects. Simulations can be easily used to produce slightly different sequences while maintaining physical realism. Secondly, real-time simulations allow a higher degree of interactivity where the real person can manoeuvre the actions of the simulated character.

In contrast, the applications based on key-framing and motion select and modify motions form a pre-computed library of motions. One drawback that simulation suffers from is the expertise and time required to handcraft the appropriate controls systems.

Key Framing

A key frame is a frame where we define changes in animation. Every frame is a key frame when we create frame by frame animation. When someone creates a 3D animation on a computer, they usually don't specify the exact position of any given object on every single frame. They create key frames.

Key frames are important frames during which an object changes its size, direction, shape or other properties. The computer then figures out all the in-between frames and saves an extreme amount of time for the animator. The following illustrations depict the frames drawn by user and the frames generated by computer.

Uses of Animation

- **Cartoons** - One of the most exciting applications of multimedia is games. Nowadays the live internet is used to play gaming with multiple players has become popular. In fact, the first application of multimedia system was in the field of entertainment and that too in the video game industry. The integrated audio and video effects make various types of games more entertaining.
- **Simulations** - Computer simulation and animation are well known for their uses in visualizing and explaining complex and dynamic events. They are also useful in the analysis and understanding of these same types of events. That is why they are becoming increasingly used in litigation. While simulation and animation are different, they both involve the application of 3D computer graphics and are presented in that form with motion on a video screen. Simulation produces motion, which is consistent with the laws of physics and relies on the inputs by the user to be consistent with the events portrayed. The motion in an animation can be derived from a reconstruction of the event or can be taken from a simulation. Currently available animation software is more advanced in the ability to build objects and scenes to achieve photos-realism. Current simulation software is able to produce a dynamic visualization in a fraction of the time required by an animation.
- **Scientific Visualisation** - Multimedia had a wide application in the field of science and technology. It is capable of transferring audio, sending message and formatted multimedia documents. At the same time it also helps in live interaction through audio messages and it is only possible with the multimedia. It reduces the time and cost can be arranged at any moment even in emergencies. At the same time, the multimedia is enough useful services based on images. Similarly, it is useful for surgeons as they can use images created from imaging scans of human body to practice complicated procedures such as brain removal and reconstructive surgery. The plans can be made in a better way to reduce the costs and complications.
- **Analysis and Understanding** - Another successful use of the animated information graphic is for explaining a process or procedure. Although technical animations have been around for a long time, modern versions use the visual language of videos, such as overlaying windows, novel transitions between segments, a popping soundtrack, lively pace and surprising sound effects. Often the concepts and ideas are pared down to the basics, as they should be when intended for the general public.
- **Teaching and Communicating** - In the area of education also, the multimedia has a great importance. Talking particularly about the schools, the usage of multimedia is very important for children also. It is extensively used in the field of education and training. Even in traditional method we used audio for imparting education, where charts, models etc. were used. Nowadays the classroom need is not limited to that traditional method rather it needs audio and visual media. The multimedia integrates all of them in one system. As an education aid the PC contains a high-quality display with mic option. This all has promoted the development of a wide range of computer based training.
- **Architecture Visualization** - Architectural Animation is a short digital architectural movie which includes the concerned project or construction, the site, animated people and vehicles, all of which are digitally generated through 2D or 3D animation techniques. Unlike an architectural rendering, which is a single image from a single point of view, an architectural animation is a series of such still images. When this

series of images are put together in a sequence and played, they produce the effect of a movie, much like a real movie except, all images in an architectural animation are digitally created by computer. It is appropriate to add a computer-generated digital landscape around the central construction to enhance its visual effect and to better convey its relationship to the surrounding area. Architectural animation is thus an effective and attractive way to provide designers and stakeholders with a realistic view of what the project will look like on completion.

3D architectural animation is highly user-friendly for the viewers since it provides an accurate realistic visual of the construction. Its gives a clear idea about the building from all angles along with a visual on core construction activities. Concepts of computer graphics and 3D animation help in creating highly realistic 3D architectural animation of any construction, and it gives a completely authentic idea of the finished product or building to the client. Designers or architects emboss their designs or plans on paper sheets and make them comprehensible for the clients through labelling. These days, various 3D animation software have been introduced in the market, which facilitate designers to present their plans in a more simplified and legible manner. Moving at par with industry standards, expert animators also use 3D animation techniques to prepare 3D project models, 3D house plans, 3D building plans, and 3D construction plans to provide a step by step analysis of the whole construction process.

Principles of Animation

Disney's Twelve Basic Principles of Animation were introduced by the Disney animators Ollie Johnston and Frank Thomas in their 1981 book The Illusion of Life: Disney Animation. Johnston and Thomas in turn based their book on the work of the leading Disney animators from the 1930s onwards, and their effort to produce more realistic animations. The main purpose of the principles was to produce an illusion of characters adhering to the basic laws of physics, but they also dealt with more abstract issues, such as emotional timing and character appeal.

Squash and Stretch

The most important principle is "squash and stretch" the purpose of which is to give a sense of weight and flexibility to drawn objects. It can be applied to simple objects, like a bouncing ball, or more complex constructions, like the musculature of a human face. Taken to an extreme point, a figure stretched or squashed to an exaggerated degree can have a comical effect in realistic animation, however, the most important aspect of this principle is the fact that an object's volume does not change when squashed or stretched. If the length of a ball is stretched vertically, its width (in three dimensions, also its depth) needs to contract correspondingly horizontally

Anticipation

Anticipation is used to prepare the audience for an action, and to make the action appear more realistic. A dancer jumping off the floor has to bend his knees first; a golfer making a swing has to swing the club back first. The technique can also be used for less physical actions, such as a character looking off-screen to anticipate someone's arrival, or attention focusing on an object that a character is about to pick up.

Staging

This principle is akin to staging in theatre, as it is known in theatre and film. Its purpose is to direct the audience's attention, and make it clear what is of greatest importance in a scene; Johnston and Thomas defined it as "the presentation of any idea so that it is completely and unmistakably clear", whether that idea is an action, a personality, an expression, or a mood. This can be done by various means, such as the placement of a character in the frame, the use of light and shadow, or the angle and position of the camera. The essence of this principle is keeping focus on what is relevant, and avoiding unnecessary detail.

Straight Ahead Action and Pose to Pose

These are two different approaches to the actual drawing process. "Straight ahead action" means drawing out a scene frame by frame from beginning to end, while "pose to pose" involves starting with drawing a few key frames, and then filling in the intervals later. "Straight ahead action" creates a more fluid, dynamic illusion of movement, and is better for producing realistic action sequences. On the other hand, it is hard to maintain proportions, and to create exact, convincing poses along the way. "Pose to pose" works better for dramatic or emotional scenes, where composition and relation to the surroundings are of greater importance. A combination of the two techniques is often used

Computer animation removes the problems of proportion related to "straight ahead action" drawing; however, "pose to pose" is still used for computer animation, because of the advantages it brings in composition. The use of computers facilitates this method, and can fill in the missing sequences in between poses automatically. It is, however, still important to oversee this process and apply the other principles discussed.

Follow Through and Overlapping Action

Follow through and overlapping action is a general heading for two closely related techniques which help to render movement more realistically, and help to give the impression that characters follow the laws of physics, including the principle of inertia. "Follow through" means that loosely tied parts of a body should continue moving after the character has stopped and the parts should keep moving beyond the point where the character stopped to be "pulled back" only subsequently towards the center of mass and/or exhibitingvarious degrees of oscillation damping. "Overlapping action" is the tendency for parts of the body to move at different rates (an arm will move on different timing of the head and so on). A third, related technique is "drag", where a character starts to move and parts of him take a few frames to catch up. These parts can be inanimate objects like clothing or the antenna on a car, or parts of the body, such as arms or hair. On the human body, the torso is the core, with arms, legs, head and hair appendices that normally follow the torso's movement. Body parts with much tissue, such as large stomachs and breasts, or the loose skin on a dog, are more prone to independent movement than bonier body parts. Again, exaggerated use of the technique can produce a comical effect, while more realistic animation must time the actions exactly, to produce a convincing result. The "moving hold" animates between similar key frames, even characters sitting still can display some sort of movement, such as the torso moving in and out with breathing.

Slow In and Slow Out

The movement of the human body, and most other objects, needs time to accelerate and slow down. For this reason, animation looks more realistic if it has more drawings near the beginning and end of an action, emphasizing the extreme poses, and fewer in the middle. This principle goes for characters moving between two extreme poses, such as sitting down and standing up, but also for inanimate, moving objects, like the bouncing ball in the above illustration.

Arc

Most natural action tends to follow an arched trajectory, and animation should adhere to this principle by following implied "arcs" for greater realism. This technique can be applied to a moving limb by rotating a joint, or a thrown object moving along a parabolic trajectory. The exception is mechanical movement, which typically moves in straight lines.

As an object's speed or momentum increases, arcs tend to flatten out in moving ahead and broaden in turns.

In baseball, a fastball would tend to move in a straighter line than other pitches; while a figure skater moving

at top speed would be unable to turn as sharply as a slower skater, and would need to cover more ground to complete the turn.
An object in motion that moves out of its natural arc for no apparent reason will appear erratic rather than fluid. For example, when animating a pointing finger, the animator should be certain that in all drawings in between the two extreme poses, the fingertip follows a logical arc from one extreme to the next. Traditional animators tend to draw the arc in lightly on the paper for reference, to be erased later.

Secondary Action

Adding secondary actions to the main action gives a scene more life, and can help to support the main action. A person walking can simultaneously swing his arms or keep them in his pockets, speak or whistle, or express emotions through facial expressions. The important thing about secondary actions is that they emphasize, rather than take attention away from the main action. If the latter is the case, those actions are better left out. For example, during a dramatic movement, facial expressions will often go unnoticed. In these cases it is better to include them at the beginning and the end of the movement, rather than during.

Timing

Timing refers to the number of drawings or frames for a given action, which translates to the speed of the action on film. On a purely physical level, correct timing makes objects appear to obey the laws of physics; for instance, an object's weight determines how it reacts to an impetus, like a push. Timing is critical for establishing a character's mood, emotion, and reaction. It can also be a device to communicate aspects of a character's personality.

Exaggeration

Exaggeration is an effect especially useful for animation, as perfect imitation of reality can look static and dull in cartoons the level of exaggeration depends on whether one seeks realism or a particular style, like a caricature or the style of a specific artist. The classical definition of exaggeration, employed by Disney, was to remain true to reality, just presenting it in a wilder, more extreme form. Other forms of exaggeration can involve the supernatural or surreal, alterations in the physical features of a character; or elements in the storyline itself. It is important to employ a certain level of restraint when using exaggeration. If a scene contains several elements, there should be a balance in how those elements are exaggerated in relation to each other, to avoid confusing or overawing the viewer.

Solid drawing

The principle of solid drawing means taking into account forms in three-dimensional space, or giving them volume and weight. The animator needs to be a skilled artist and has to understand the basics of three-dimensional shapes, anatomy, weight, balance, light and shadow, etc. For the classical animator, this involved taking art classes and doing sketches from life. One thing in particular that Johnston and Thomas warned against was creating "twins": characters whose left and right sides mirrored each other, and looked lifeless. Modern-day computer animators draw less because of the facilities computers give them, yet their work benefits greatly from a basic understanding of animation principles and their additions to basic computer animation

Appeal

Appeal in a cartoon character corresponds to what would be called charisma in an actor. A character who is appealing is not necessarily sympathetic – villains or monsters can also be appealing – the important thing is that the viewer feels the character is real and interesting. There are several tricks for making a character connect better with the audience; for likable characters a symmetrical or particularly baby-like face tends to be effective. A complicated or hard to read face will lack appeal, it may more accurately be described as 'captivation' in the composition of the pose, or the character design.

Computer Based Animation

Computer animation is the art of creating moving images via the use of computers. It is a subfield of computer graphics and animation. Increasingly it is created by means of 3D computer graphics, though 2D computer graphics are still widely used for low bandwidth and faster real-time rendering needs. It is also referred to as CGI (Computer-generated imagery or computer-generated imaging), especially when used in films. To create the illusion of movement, an image is displayed on the computer screen then quickly replaced by a new image that is similar to the previous image, but shifted slightly. This technique is identical to how the illusion of movement is achieved with television and motion pictures.

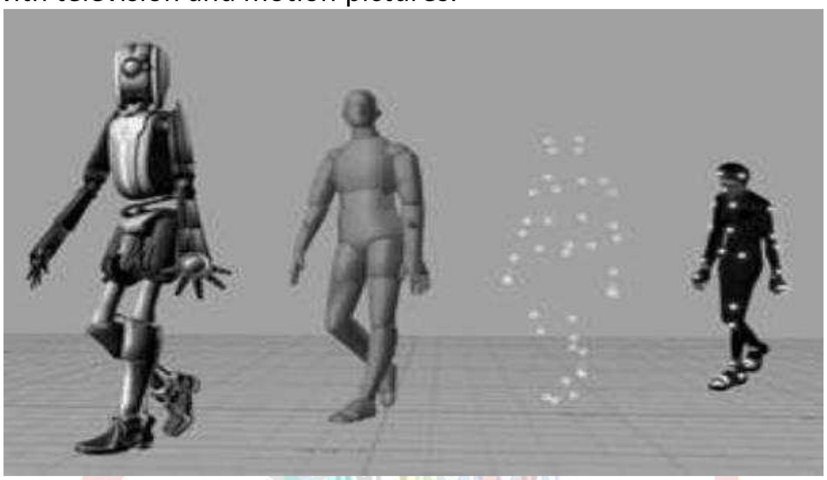


Figure 5.1 An example of computer animation which is produced in the "motion capture" technique

Computer animation is essentially a digital successor to the art of stop motion animation of 3D models and frame-by-frame animation of 2D illustrations. For 3D animations, objects (models) are built on the computer monitor (modeled) and 3D figures are rigged with a virtual skeleton. For 2D figure animations, separate objects (illustrations) and separate transparent layers are used, with or without a virtual skeleton. Then the limbs, eyes, mouth, clothes, etc. of the figure are moved by the animator on key frames. The differences in appearance between key frames are automatically calculated by the computer in a process known as tweening or morphing. Finally, the animation is rendered.

For 3D animations, all frames must be rendered after modeling is complete. For 2D vector animations, the rendering process is the key frame illustration process, while tweened frames are rendered as needed. For pre-recorded presentations, the rendered frames are transferred to a different format or medium such as film or digital video. The frames may also be rendered in real time as they are presented to the end-user audience. Low bandwidth animations transmitted via the internet (e.g. 2D Flash, X3D) often use software on the end-users computer to render in real time as an alternative to streaming or pre-loaded high bandwidth animations.

In most 3D computer animation systems, an animator creates a simplified representation of a character's anatomy, which is analogous to a skeleton or stick figure. The position of each segment of the skeletal model is defined by animation variables, or Avers for short. In human and animal characters, many parts of the skeletal model correspond to the actual bones, but skeletal animation is also used to animate other things, with facial features (though other methods for facial animation exist). The character "Woody" in Toy Story, for example, uses 700 Avers (100 in the face alone). The computer doesn't usually render the skeletal model directly (it is invisible), but it does use the skeletal model to compute the exact position and orientation of that certain character, which is eventually rendered into an image. Thus, by changing the values of Avers over time, the animator creates motion by making the character move from frame to frame.

There are several methods for generating the Avar values to obtain realistic motion. Traditionally, animators manipulate the Avers directly. Rather than set Avers for every frame, they usually set Avers at strategic points (frames) in time and let the computer interpolate or tween between them in a process called key framing. Key framing puts control in the hands of the animator and has roots in hand-drawn traditional animation.

In contrast, a newer method called motion capture makes use of live action footage. When computer animation is driven by motion capture, a real performer acts out the scene as if they were the character to be animated. His/her motion is recorded to a computer using video cameras and markers and that performance is then applied to the animated character.

Each method has its advantages and as of 2007, games and films are using either or both of these methods in productions. Key frame animation can produce motions that would be difficult or impossible to act out, while motion capture can reproduce the subtleties of a particular actor. Motion capture is appropriate in situations where believable, realistic behavior and action is required, but the types of characters required exceed what can be done throughout the conventional costuming.

Modeling

3D computer animation combines 3D models of objects and programmed or hand "key framed" movement. These models are constructed out of geometrical vertices, faces, and edges in a 3D coordinate system. Objects are sculpted much like real clay or plaster, working from general forms to specific details with various sculpting tools. Unless a 3D model is intended to be a solid color, it must be painted with "textures" for realism. A bone/joint animation system is set up to deform the CGI model (e.g., to make a humanoid model walk). In a process known as rigging, the virtual marionette is given various controllers and handles for controlling movement. Animation data can be created using motion capture or key framing by a human animator, or a combination of the two.

3D models rigged for animation may contain thousands of control points — for example, "Woody" from Toy Story uses 700 specialized animation controllers. Rhythm and Hues Studios laboured for two years to create Aslan in the movie The Chronicles of Narnia: The Lion, the Witch and the Wardrobe, which had about 1,851 controllers (742 in the face alone). In the 2004 film The Day After Tomorrow, designers had to design forces of extreme weather with the help of video references and accurate meteorological facts. For the 2005 remake of King Kong, actor Andy Serkis was used to help designers pinpoint the gorilla's prime location in the shots and used his expressions to model "human" characteristics onto the creature. Serkis had earlier provided the voice and performance for Gollum in J. R. R. Tolkien's The Lord of the Rings trilogy.

Figure 5.2 A ray-traced 3-D model of a jack inside a cube, and the jack alone below

Computer animation can be created with a computer and animation software. Some impressive animation can be achieved even with basic programs; however, the rendering can take a lot of time on an ordinary home computer. Professional animators of movies, television and video games could make photorealistic animation with high detail. This level of quality for movie animation would take hundreds of years to create on a home computer. Instead, many powerful workstation computers are used. Graphics workstation computers use two to four processors, and they are a lot more powerful than an actual home computer and are specialized for rendering. A large number of workstations (known as a "render farm") are networked together to effectively act as a giant computer.

The result is a computer-animated movie that can be completed in about one to five years (however, this process is not composed solely of rendering). A workstation typically costs $2,000-16,000 with the more expensive stations being able to render much faster due to the more technologically-advanced hardware that they contain. Professionals also use digital movie cameras, motion/performance capture, bluescreens, and film editing software, props, and other tools used for movie animation.

Facial animation

The realistic modelling of human facial features is both one of the most challenging and sought-after elements in computer-generated imagery. Computer facial animation is a highly complex field where models typically include a very large number of animation variables. Historically speaking, the first SIGGRAPH tutorials on State of the art in Facial Animation in 1989 and 1990 proved to be a turning point in the field by bringing together and consolidating multiple research elements and sparked interest among a number of researchers

The Facial Action Coding System (with 46 "action units", "lip bite" or "squint"), which had been developed in 1976, became a popular basis for many systems. As early as 2001, MPEG-4 included 68 Face Animation Parameters (FAPs) for lips, jaws, etc., and the field has made significant progress since then and the use of facial micro expression has increased.

In some cases, an affective space, the PAD emotional state model, can be used to assign specific emotions to the faces of avatars. In this approach, the PAD model is used as a high level emotional space and the lower level space is the MPEG-4 Facial Animation Parameters (FAP). A mid-level Partial Expression Parameters (PEP) space is then used to in a two-level structure – the PAD-PEP mapping and the PEP-FAP translation model.

Realism in computer animation can mean making each frame look photorealistic, in the sense that the scene is rendered to resemble a photograph or make the characters' animation believable and lifelike. Computer animation can also be realistic with or without the photorealistic rendering.

One of the greatest challenges in computer animation has been creating human characters that look and move with the highest degree of realism. Part of the difficulty in making pleasing, realistic human characters is the uncanny valley, the concept where the human audience (up to a point) tends to have an increasingly negative, emotional response as a human replica looks and acts more and more human.

The goal of computer animation is not always to emulate live action as closely as possible; so many animated films instead feature characters that are anthropomorphic animals, fantasy creatures and characters, superheroes, or otherwise have non-realistic, cartoon-like proportions. Computer animation can also be tailored to mimic or substitute for other kinds of animation, traditional stop-motion animation (as shown in Flushed Away or The Lego Movie). Some of the long-standing basic principles of animation, like squash & stretch, call for movement that is not strictly realistic, and such principles still see widespread application in computer animation.

Animation file formats

There are a number of different types of animation file formats. Each type stores graphics data in a different way. Bitmap, vector, and metafile formats are by far the most commonly used formats, and we focus on these. However, there are other types of formats as well - scene, animation, multimedia, hybrid, hypertext, hypermedia, 3D, virtual modeling reality language (VRML), audio, font, and page description language (PDL). The increasing popularity of the World Wide Web has made some of these formats more popular.

File Name	Description
JPEG/JPG (Joint Photographers Expert Group)	Most popular lossy image format. Allows users to specify what level of compression they desire.
PNG (Portable Network Graphic)	Best of lossless image formats. Widely supported across web. Allows you to include an alpha channel within file.
BMP (BitMaP)	Would avoid if possible. They offer little to no compression which results in unnecessarily large files.
TIFF/TIF (Tagged Image File Format)	Offers both compressed and uncompressed versions. Compressed are similar to PNG and uncompressed are similar to BMP.
PDF (Portable Document Format)	Most widely used document format. Great vector image format. Created by Adobe.
EPS (Encapsulated PostScript)	Most common vector image format. Standard format for print industry.
GIF (Graphic Interchange Format)	Lossless format that supports both animated and static images. Great for webpage banner ads.

Animation software's

Computer graphics animation software and tools are continuing to advance at an amazing pace to create today's digital animation movies. New generations of animated movie lovers are getting used to seeing CG visuals that were not possible just five to 10 years ago. If you want to be a 3D artist, but you don't know which software to choose? Whether you're a newcomer to art programs or incredibly experienced, here is the list of 3D modeling programs you should check out and consider using:

- LightWave 3D (NewTek) - LightWave 3D combines a state-of-the-art renderer with powerful, intuitive modeling, and animation tools. Tools that may cost extra in other professional 3D applications are part of the product package, including 999 free cross-platform render nodes, support for Windows and Mac UB 64 and 32-bit operating systems, free technical support and more. LightWave is enjoyed worldwide, as a complete 3D production solution for feature film and television visual effects, broadcast design, print graphics, visualization, game development, and Web. LightWave is responsible for more artists winning Emmy Awards than any other 3D application.
- Blender (The Blender Foundation) - Blender is a professional, free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender's features include 3D modeling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, camera tracking, rendering, motion graphics, video editing and compositing. It also features an integrated game engine.
- 3ds Max (Autodesk) - Autodesk 3ds Max, formerly 3D Studio, then 3D Studio Max is a professional 3D computer graphics program for making 3D animations, models, games and images. It is developed and produced by Autodesk Media and Entertainment. It has modeling capabilities and flexible plugin architecture and can be used on the Microsoft Windows platform. It is frequently used by video game developers, many TV commercial studios and architectural visualization studios. It is also used for movie

effects and movie pre-visualization. For its modeling and animation tools, the latest version of 3ds Max also features shaders (such as ambient occlusion and subsurface scattering), dynamic simulation, particle systems, radiosity, normal map creation and rendering, global illumination, a customizable user interface, new icons, and its own scripting language.

- Maya (Autodesk) - Autodesk Maya commonly shortened to Maya, is a 3D computer graphics software that runs on Windows, macOS and Linux, originally developed by Alias Systems Corporation (formerly Alias|Wavefront) and currently owned and developed by Autodesk, Inc. It is used to create interactive 3D applications, including video games, animated film, TV series, or visual effects.
- Cinema 4D (Maxon) - Cinema 4D is the professional 3D package for your needs. If you want to create advanced 3D graphics but need a helping hand to ensure you create jaw-dropping graphics quickly and easily, then Cinema 4D is the choice for you. Despite being designed for advanced 3D, the extra tools found in Cinema 4D Studios are still designed to be user-friendly and intuitive. Generating advanced 3D effects such as hair is surprisingly easy and fast, with Cinema 4D doing much of the work for you. Easy to learn and extremely powerful: Cinema 4D is the perfect package for all 3D artists who want to achieve breathtaking results fast and hassle-free. Beginners and seasoned professionals alike can take advantage of Cinema 4D's wide range of tools and features to quickly achieve stunning results. Cinema 4D's legendary reliability also makes it the perfect application for demanding, fast-paced 3D production, and a range of attractively priced software packages is available to fit any artist's needs.
- Softimage (Autodesk) - Autodesk Softimage, or simply Softimage is a discontinued 3D computer graphics application, for producing 3D computer graphics, 3D modeling, and computer animation. Now owned by Autodesk and formerly titled Softimage|XSI, the software has been predominantly used in the film, video game, and advertising industries for creating computer generated characters, objects, and environments.

- ZBrush (Pixologic) - ZBrush is the 3D industry's standard digital sculpting application. Use customizable brushes to shape, texture, and paint virtual clay, while getting instant feedback. Work with the same tools used by film studios, game developers and artists the world over. ZBrush is a digital sculpting tool that combines 3D/2.5D modeling, texturing and painting. It uses a proprietary "pixol" technology which stores lighting, color, material, and depth information for all objects on the screen. The main difference between ZBrush and more traditional modeling packages is that it is more akin to sculpting.
- Mudbox (Autodesk) - Mudbox is a proprietary computer-based 3D sculpting and painting tool. Currently developed by Autodesk, Mudbox was created by Skymatter, founded by Tibor Madjar, David Cardwell and Andrew Camenisch, former artists of Weta Digital, where it was first used to produce the 2005 Peter Jackson remake of King Kong. Mudbox's primary application is high-resolution digital sculpting, texture painting, and displacement and normal map creation, although it is also used as a design tool. The Mudbox user interface is a 3D environment that allows the creation of movable cameras that can be bookmarked. Models created within the program typically start as a polygon mesh that can be manipulated with a variety of different tools. A model can be subdivided to increase its resolution and the number of polygons available to sculpt with. 3D layers allow the user to store different detail passes, blending them with multiplier sliders and layer masks. Using layers the user is able to sculpt and mould their 3D model without making permanent changes.
- Modo (Luxology) - Modo's powerful and flexible 3D modeling, texturing and rendering toolset empowers artists to explore and develop ideas without jumping through technical hoops. Modo is your starting point for creative exploration. Modo is a polygon and subdivision surface modeling, sculpting, 3D painting, animation and rendering package developed by Luxology, LLC, which is now merged with and known as Foundry. The program incorporates features such as n-gons and edge weighting, and runs on Microsoft Windows, Linux and macOS platforms.

Compression Techniques

There are two categories of compression techniques used with digital graphics, lossy and lossless.

Lossless and lossy compression are terms that describe whether or not, in the compression of a file, all original data can be recovered when the file is uncompressed. With lossless compression, every single bit of data that was originally in the file remains after the file is uncompressed. All of the information is completely restored. This is generally the technique of choice for text or spreadsheet files, where losing words or financial data could pose a problem. The Graphics Interchange File (GIF) is an image format used on the Web that provides lossless compression.

On the other hand, lossy compression reduces a file by permanently eliminating certain information, especially redundant information. When the file is uncompressed, only a part of the original information is still there (although the user may not notice it). Lossy compression is generally used for video and sound, where a certain amount of information loss will not be detected by most users. The JPEG image file, commonly used for photographs and other complex still images on the Web, is an image that has lossy compression. Using JPEG compression, the creator can decide how much loss to introduce and make a trade-off between file size and image quality.

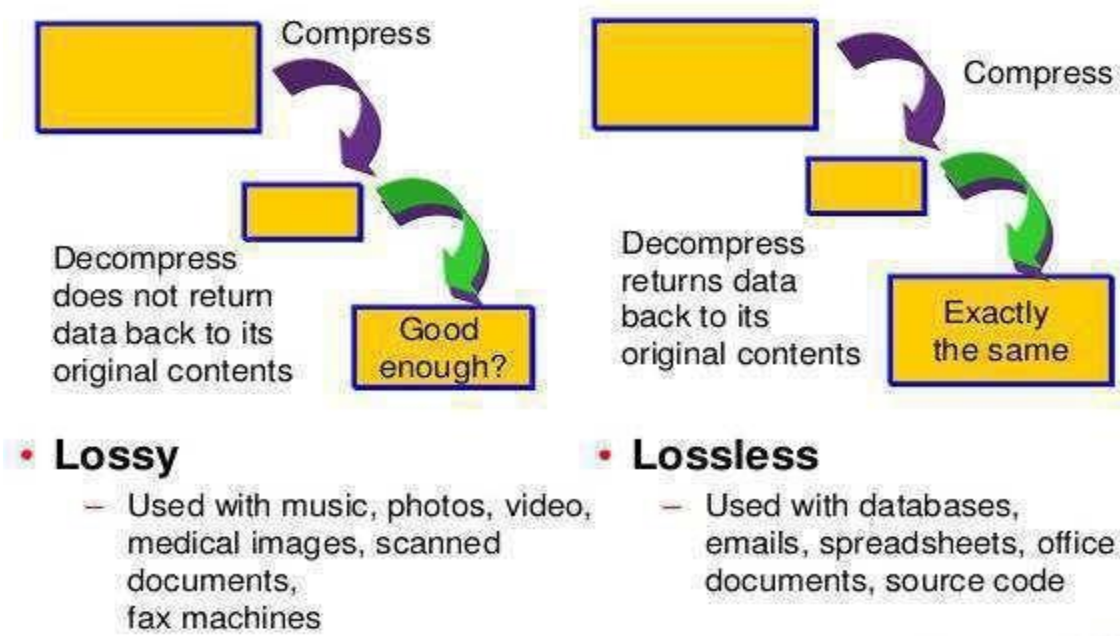


Figure 5.3 Difference between Lossless VS. Lossy Compression Technique

Image Compression

Image compression is a type of data compression applied to digital images, to reduce their cost for storage or transmission. Algorithms may take advantage of visual perception and the statistical properties of image data to provide superior results compared with generic compression methods.

Methods for lossless image compression are:

- Run-length encoding – used in default method in PCX and as one of possible in BMP, TGA, TIFF
- Area image compression
- DPCM and Predictive Coding
- Entropy encoding
- Adaptive dictionary algorithms such as LZW – used in GIF and TIFF
- Deflation – used in PNG, MNG, and TIFF
- Chain codes

Methods for lossy compression:

- Reducing the color space to the most common colors in the image. The selected colors are specified in the color palette in the header of the compressed image. Each pixel just references the index of a color in the colour palette; this method can be combined with dithering to avoid posterization.
- Chroma subsampling. This takes advantage of the fact that the human eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information in the image.
- Transform coding. This is the most commonly used method. In particular, a Fourier-related transform such as the Discrete Cosine Transform (DCT) is widely used: N. Ahmed, T. Natarajan and K.R. Rao, "Discrete Cosine Transform," IEEE Trans. Computers, 90-93, Jan. 1974. The DCT is sometimes referred to as "DCT-II" in the context of a family of discrete cosine transforms; e.g., see discrete cosine transform. The more recently developed wavelet transform is also used extensively, followed by quantization and entropy coding.
- Fractal compression.

Figure 5.4 Lossless VS. Lossy Compression

Audio Compression

Audio compression (data), a type of lossy or lossless compression in which the amount of data in a recorded waveform is reduced to differing extents for transmission respectively with or without some loss of quality, used in CD and MP3 encoding, Internet radio. Dynamic range compression, also called audio level compression, in which the dynamic range, the difference between loud and quiet, of an audio waveform is reduced.

Audio Compression Methods

Traditional lossless compression methods (Huffman, LZW, etc.) usually don't work well on audio compression (the same reason as in image compression).

The following are some of the Lossy methods applied to audio compression:

- Silence Compression - detect the "silence", similar to run-length coding
- Adaptive Differential Pulse Code Modulation (ADPCM)
  e.g., in CCITT G.721 - 16 or 32 Kbits/sec

a. encodes the difference between two consecutive signals,
b. adapts at quantization so fewer bits are used when the value is smaller.
    - It is necessary to predict where the waveform is headed -> difficult
    - Apple has proprietary scheme called ACE/MACE. Lossy scheme that tries to predict where wave will go in next sample. About 2:1 compression.

- Linear Predictive Coding (LPC) fits signal to speech model and then transmits parameters of model. Sounds like a computer talking, 2.4 kbits/sec
- Code Excited Linear Predictor (CELP) does LPC, but also transmits error term - audio conferencing quality at 4.8 kbits/sec

Video Compression

A video consists of a time - ordered sequence of frames — images. An obvious solution to video compression would be predictive coding based on previous frames.

A simple calculation shows that an uncompressed video produces an enormous amount of data: a resolution of 720x576 pixels (PAL), with a refresh rate of 25 fps and 8-bit colour depth, would require the following bandwidth:

$$720 \times 576 \times 25 \times 8 + 2 \times (360 \times 576 \times 25 \times 8) = 1.66 \text{ Mb/s (luminance + chrominance)}$$

For High Definition Television (HDTV):

$$1920 \times 1080 \times 60 \times 8 + 2 \times (960 \times 1080 \times 60 \times 8) = 1.99 \text{ Gb/s}$$

Even with powerful computer systems (storage, processor power, network bandwidth), such data amount cause extreme high computational demands for managing the data. Fortunately, digital video contains a great deal of redundancy. Thus it is suitable for compression, which can reduce these problems significantly.

Especially lossy compression techniques deliver high compression ratios for video data. However, one must keep in mind that there is always a trade-off between data size (therefore computational time) and quality. The higher the compression ratio, the lower the size and the lower the quality. The encoding and decoding process itself also needs computational resources, which have to be taken into consideration. It makes no sense, for example for a real-time application with low bandwidth requirements, to compress the video with a computational expensive algorithm which takes too long to encode and decode the data.

Video Compression Standards

The following compression standards are the most known nowadays. Each of them is suited for specific applications. Top entry is the lowest and last row is the most recent standard. The MPEG standards are the most widely used ones, which will be explained in more details in the following sections.

Standard	Application
JPEG	Still image compression
H.261	Video conferencing over ISDN
MPEG-1	Video on digital storage media (CD-ROM)
MPEG-2	Digital Television
H.263	Video telephony over PSTN
MPEG-4	Object-based coding, synthetic content, interactivity
JPEG-2000	Improved still image compression
H.264/ MPEG-4 AVC	Improved video compression

MPEG Standards

The MPEG standards are an evolving set of standards for video and audio compression and for multimedia delivery developed by the Moving Picture Experts Group (MPEG).

MPEG-G

A suite of standards to provide new effective and interoperable solutions for genomic information processing applications

MPEG-CICP
A suite of standards to specify code points for non-standard specific media formats

MPEG-I
A collection of standards to digitally represent immersive media

MPEG-DASH
DASH is a suite of standards providing a solution for the efficient and easy streaming of multimedia using existing available HTTP infrastructure (particularly servers and CDNs, but also proxies, caches, etc.).

MPEG-H
Suite of standards for heterogeneous environment delivery of audio-visual information compressed with high efficiency

MPEG-U
MPEG-U provides a general purpose technology with innovative functionality that enable its use in heterogeneous scenarios such as broadcast, mobile, home network and web domains:

MPEG-M
MPEG-M is a suite of standards to enable the easy design and implementation of media-handling value chains whose devices interoperate because they are all based on the same set of technologies, especially MPEG technologies accessible from the middleware and multimedia services

MPEG-V
MPEG-V outlines an architecture and specifies associated information representations to enable interoperability between virtual worlds (e.g., digital content provider of a virtual world, gaming, simulation), and between real and virtual worlds (e.g., sensors, actuators, vision and rendering, robotics).

MPEG-E
A standard for an Application Programming Interface (API) of Multimedia Middleware (M3W) that can be used to provide a uniform view to an interoperable multimedia middleware platform

MPEG-D
A suite of standards for Audio technologies that do not fall in other MPEG standards

MPEG-C
A suite of video standards that do not fall in other well-established MPEG Video standards

MPEG-B
A suite of standards for systems technologies that do not fall in other well-established MPEG standards

MPEG-A
A suite of standards specifying application formats that involve multiple MPEG and, where required, non MPEG standards

MPEG-21
A suite of standard that define a normative open framework for end-to-end multimedia creation, delivery and consumption that provides content creators, producers, distributors and service providers with equal opportunities in the MPEG-21 enabled open market, and also be to the benefit of the content consumers providing them access to a large variety of content in an interoperable manner.

MPEG-MAR
A Mixed and Augmented Reality Reference Model developed jointly with SC 24/WG 9

MPEG-7
A suite of standards for description and search of audio, visual and multimedia content

MPEG-4
A suite of standards for multimedia for the fixed and mobile web

MPEG-2
A suite for standards for digital television

MPEG-1
A suite of standards for audio-video and systems particularly designed for digital storage media

Multimedia Systems Architecture

The architecture of multimedia system may be described as a four-level hierarchy. In line with concepts developed in conventional layered systems such as the OSI and Internet each layer performs a specific function and supports the function performed in the layer above. The four-layers (lowest (bottom) layer first) of the architecture, known as the RT architecture (Real-time information handling), are:

Network Subsystem (Layer 1)

This layer takes care of the functionalities up layer 3 in the OSI model. Network specific functions depend on the technology used in this layer. Essentially this level provides a possible connection through a network with a specified bandwidth and error probability as supported by the underlying technology.

End-to-End QoS Control (Layer 2)

This layer maintains the connection between the source and destination and can be conceptually viewed as a single connection -- even though there may be physically many more. Each connection is managed to ensure that a given Quality of Service (Qos) is maintained.

Media Management (layer 3)

This layer provides generic services to applications so far as media management is concerned. A primary functions is synchronization across the media.

Application (Layer 4)

This layer is in the direct interface with the user. The application will also interface with the operating system, if required -- for example calls to storage media or specific library functions (subroutines).

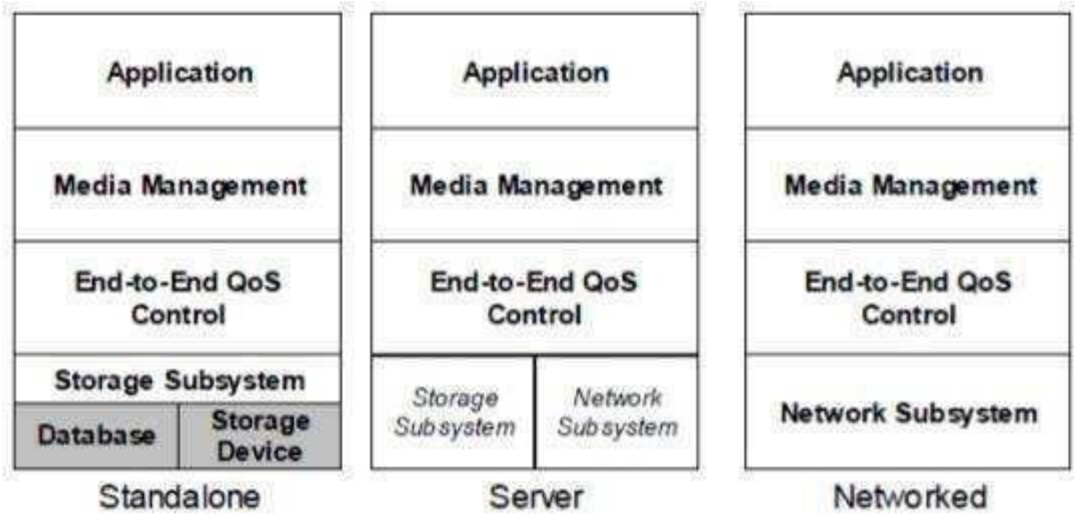


Figure 5.5 Real time multimedia architecture

Multimedia databases

Multimedia data typically means digital images, audio, video, animation and graphics together with text data. The acquisition, generation, storage and processing of multimedia data in computers and transmission over networks have grown tremendously in the recent past.

Multimedia data are blessed with a number of exciting features. They can provide more effective

dissemination of information in science, engineering, medicine, modern biology, and social sciences. It also facilitates the development of new paradigms in distance learning, and interactive personal and group entertainment.

The huge amount of data in different multimedia-related applications warranted to have databases as databases provide consistency, concurrency, integrity, security and availability of data. From an user perspective, databases provide functionalities for the easy manipulation, query and retrieval of highly relevant information from huge collections of stored data.

MultiMedia Databases (MMDBs) have to cope up with the increased usage of a large volume of multimedia data being used in various software applications. The applications include digital libraries, manufacturing and retailing, art and entertainment, journalism and so on. Some inherent qualities of multimedia data have both direct and indirect influence on the design and development of a multimedia database. MMDBs are supposed to provide almost all the functionalities, a traditional database provides. Apart from those, a MMDB has to provide some new and enhanced functionalities and features. MMDBs are required to provide unified frameworks for storing, processing, retrieving, transmitting and presenting a variety of media data types in a wide variety of formats. At the same time, they must adhere to numerical constraints that are normally not found in traditional databases.

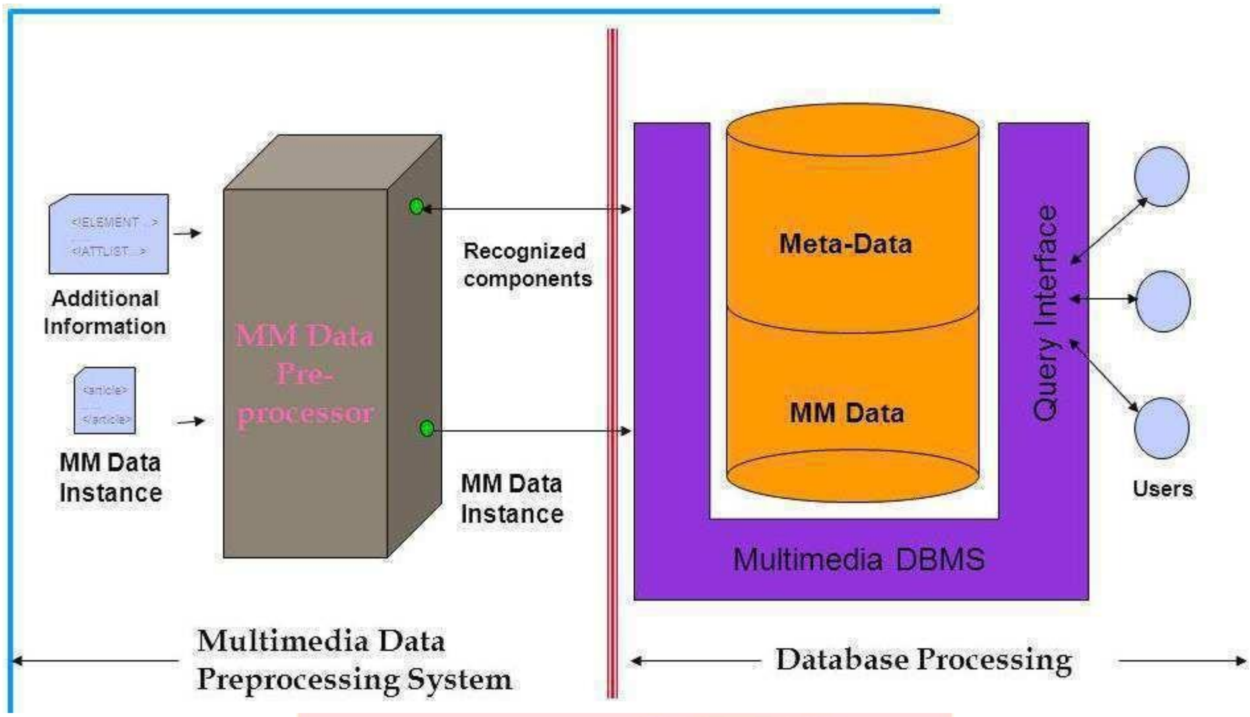


Figure 5.6 Multimedia Database Architecture

Contents of MMDB

An MMDB needs to manage several different types of information pertaining to the actual multimedia data. They are:

- **Media data** - This is the actual data representing images, audio, video that is captured, digitized, processes, compressed and stored.

- **Media format data** - This contains information pertaining to the format of the media data after it goes through the acquisition, processing, and encoding phases. For instance, this consists of information such as the sampling rate, resolution, frame rate, encoding scheme etc.

- **Media keyword data** - This contains the keyword descriptions, usually relating to the generation of the media data. For example, for a video, this might include the date, time, and place of recording, the person who recorded, the scene that is recorded, etc This is also called as content descriptive data.

- **Media feature data** - This contains the features derived from the media data. A feature characterizes the media contents. For example, this could contain information about the distribution of colors, the kinds of textures and the different shapes present in an image. This is also referred to as content dependent data.

The last three types are called meta-data as they describe several different aspects of the media data. The media keyword data and media feature data are used as indices for searching purpose. The media format data is used to present the retrieved informatio