

Subject Name : Graphics and Multimedia (63A)
Department: COMPUTER SCIENCE & APPLICATIONS
Class: III BCA / CS
Semester: VI

Unit I:

Output Primitives: Points and Lines – Line-Drawing algorithms – Loading frame Buffer – Line function – Circle-Generating algorithms – Ellipse-generating algorithms. Attributes of Output Primitives: Line Attributes – Curve attributes – Color and Grayscale Levels – Area-fill attributes – Character Attributes.

1. POINTS AND LINES Introduction

The basic elements constituting a graphic are called output primitives. Each output primitive has an associated set of attributes, such as line width and line color for lines. The programming technique is to set values for the output primitives and then call a basic function that will draw the desired primitive using the current settings for the attributes. Various graphics systems have different graphics primitives. For example GKS defines five output primitives namely, polyline (for drawing contiguous line segments), polymarker (for marking coordinate positions with various symmetric text symbols), text (for plotting text at various angles and sizes), fill area (for plotting polygonal areas with solid or hatch fill), cell array (for plotting portable raster images). At the same time graph has the output primitives namely Polyline, Polymarker, Text, Tone and have other secondary primitives besides these namely, Line and Arrow.

Points and Lines

In a CRT monitor, the electron beam is turned on to illuminate the screen phosphor at the selected location. Depending on the display technology, the positioning of the electron beam changes. In a random-scan (vector) system point plotting instructions are stored in a display list and the coordinate values in these instructions are converted to deflection voltages the position the electron beam at the screen locations.

Low-level procedure for plotting a point on the screen at (x,y) with intensity “I” can be given as **setPixel(x,y)**.

A line is drawn by calculating the intermediate positions between the two end points and displaying the pixels at those positions.

Rasterization

Rasterization is the process of converting a vertex representation to a pixel representation; rasterization is also called **scan conversion**. Included in this definition are geometric objects such as circles where you are given a center and radius.

1. The digital differential analyzer (DDA) which introduces the basic concepts for rasterization.
2. Bresenham's algorithm which improves on the DDA.

Scan conversion algorithms use **incremental** methods that exploit **coherence**. An incremental method computes a new value quickly from an old value, rather than computing the new value from scratch, which can often be slow. Coherence in space or time is the term used to denote that nearby objects (e.g., pixels) has qualities similar to the current object.

2. LINE DRAWING ALGORITHM

Digital Differential Analyzer (DDA) Algorithm

In this algorithm, the line is sampled at unit intervals in one coordinate and find the corresponding values nearest to the path for the other coordinate. For a line with positive slope less than one, $\Delta x > \Delta y$ (where $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$). Hence we sample at unit x intervals and compute each successive y values as $y_{k+1} = y_k + m$.

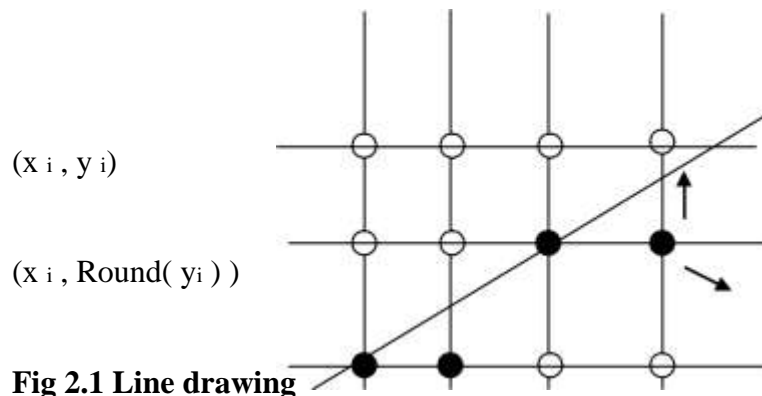


Fig 2.1 Line drawing

For lines with positive slope greater than one, $\Delta y > \Delta x$. Hence we sample at unit y intervals and compute each successive x values as

$$X_{k+1} = X_k + 1/m$$

Since the slope, m , can be any real number, the calculated value must be rounded to the nearest integer.

For a line with negative slope, if the absolute value of the slope is less than one, we make unit increment in the x direction and calculate y values as

$$y_{k+1} = y_k + m$$

For a line with negative slope, if the absolute value of the slope is greater than one, we make unit decrement in the y direction and calculate x values as

$$X_{k+1} = X_k - 1/m$$

[Note: - for all the above four cases it is assumed that the first point is on the left and second point is in the right.]

DDA Line Algorithm

```
void myLine(int x1, int y1, int x2, int y2)
```

```
{
```

```
int length,i;
```

```
double x,y;
```

```
double xincrement;  
double yincrement;  
length = abs(x2 - x1);  
if (abs(y2 - y1) > length) length = abs(y2 - y1);  
xincrement = (double)(x2 - x1)/(double)length;  
yincrement = (double)(y2 - y1)/(double)length;  
x = x1 + 0.5;  
y = y1 + 0.5;  
for (i = 1; i <= length; ++i)  
{  
    myPixel((int)x, (int)y);  
    x = x + xincrement;  
    y = y + yincrement;  
}}
```

Bresenham's Algorithm

In this method, developed by Jack Bresenham, we look at just the center of the pixels. We determine $d1$ and $d2$ which is the "error", i.e., the difference from the "true line."

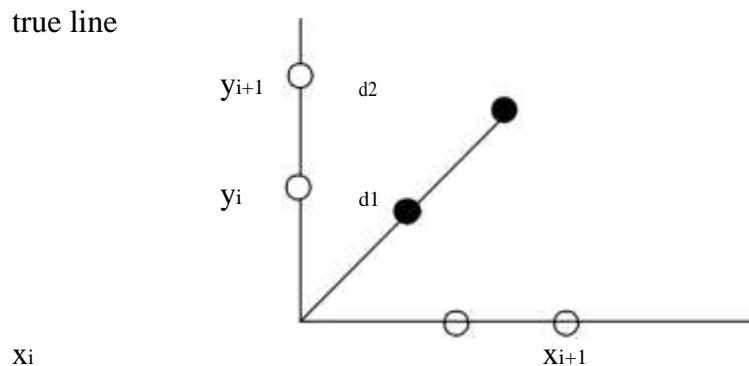


Fig 2.2 Bresenham's drawing

Steps in the Bresenham algorithm:

1. Determine the error terms
2. Define a relative error term such that the sign of this term tells us which pixel to choose
3. Derive equation to compute successive error terms from first
4. Compute first error term

Now the y coordinate on the mathematical line at pixel position x_{i+1} is calculated as $y = m(x_{i+1}) + b$

And the distances are calculated as

$$d_1 = y - y_i = m(x_i + 1) + b - y_i$$

$$d_2 = (y_i + 1) - y = y_i + 1 - m(x_i + 1) - b$$

Then

$$d_1 - d_2 = 2m(x_i + 1) - 2y + 2b - 1$$

Now define $p_i = dx(d_1 - d_2)$ = relative error of the two pixels.

Note: $p_i < 0$ if y_i pixel is closer, $p_i \geq 0$ if y_{i+1} pixel is closer. Therefore we only need to know the sign of p_i . With $m = dy/dx$ and substituting in for $(d_1 - d_2)$ we get

$$p_i = 2 * dy * x_i - 2 * dx * y_i + 2 * dy + dx * (2 * b - 1) \text{ -----} \rightarrow (1)$$

$$\text{Let } C = 2 * dy + dx * (2 * b - 1)$$

Now look at the relation of p's for successive x terms.

$$p_{i+1} = 2dy * x_{i+1} - 2 * dx * y_{i+1} + C$$

$$p_{i+1} - p_i = 2 * dy * (x_{i+1} - x_i) - 2 * dx * (y_{i+1} - y_i)$$

With $x_{i+1} = x_i + 1$ and $y_{i+1} = y_i + 1$ or y_i

$$p_{i+1} = p_i + 2 * dy - 2 * dx(y_{i+1} - y_i)$$

Now compute $p_1(x_1, y_1)$ from (1), where $b = y - dy / dx * x$

$$p_1 = 2dy * x_1 - 2dx * y_1 + 2dy + dx(2y_1 - 2dy / dx * x_1 - 1)$$

$$= 2dy * x_1 - 2dx * y_1 + 2dy + 2dx * y_1 - 2dyx_1 - dx$$

$$= 2dy - dx$$

If $p_i < 0$, plot the pixel (x_{i+1}, y_i) and next decision parameter is $p_{i+1} = p_i + 2dy$

Else and plot the pixel (x_i+1, y_i+1) and next decision parameter is

$$p_{i+1} = p_i + 2dy - 2dx$$

Bresenham Algorithm for 1st octant:

1. Enter endpoints (x_1, y_1) and (x_2, y_2) .
2. Display x_1, y_1 .
3. Compute $dx = x_2 - x_1$; $dy = y_2 - y_1$; $p_1 = 2dy - dx$.
4. If $p_1 < 0.0$, display $(x_1 + 1, y_1)$, else display $(x_1+1, y_1 + 1)$
5. if $p_1 < 0.0$, $p_2 = p_1 + 2dy$, else $p_2 = p_1 + 2dy - 2dx$
6. Repeat steps 4, 5 until reach x_2, y_2 .

Note: Only integer Addition and Multiplication by 2. Notice we always increment x by 1.

For a generalized Bresenham Algorithm must look at behavior in different octants.

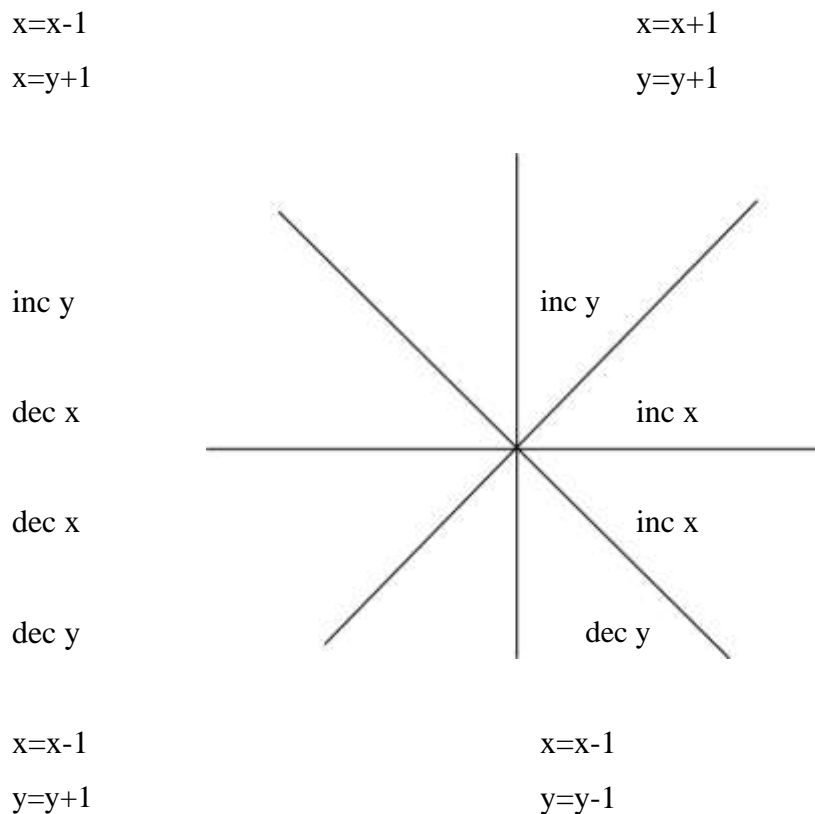


Fig 2.3 Bresenham's drawing

3. LODING THE FRAME BUFFER

To display a line or other objects in a raster system, frame buffer positions must be calculated. *setPixel* procedure stores intensity values for the pixels at corresponding address with in the frame-buffer array.

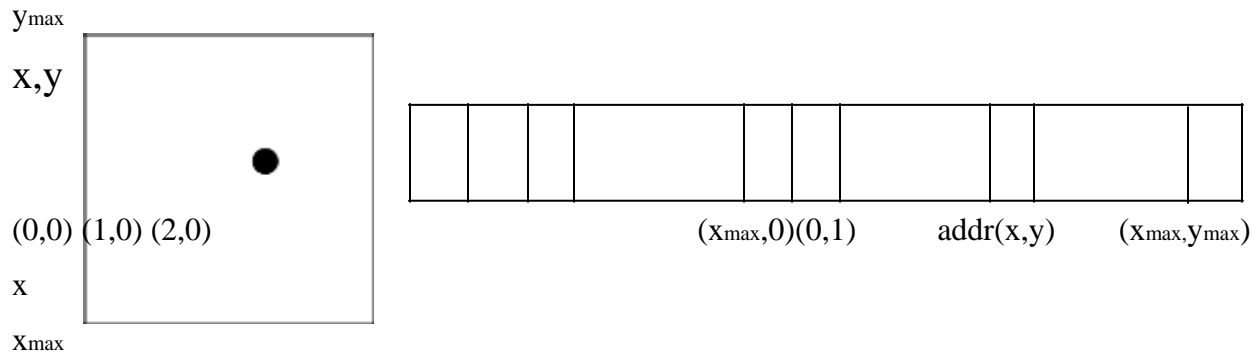


Fig 3.1 pixel positions stored linearly in row-major order within frame buffer

Example

Suppose the frame-buffer is addressed in row major order and that pixel positions vary from (0, 0) at the lower left screen corner to (x_{\max} , y_{\max}) at the top right corner.

For a bi-level system, the frame buffer bit address for pixel positions (x,y) is calculated as

$$\text{addr}(x,y) = \text{addr}(0,0) + y(x_{\max} + 1) + x$$

We can calculate the frame-buffer for the pixel at (x+1,y) as the following offset from the address for position (x,y):

$$\text{addr}(x+1,y) = \text{addr}(x,y) + 1$$

Stepping diagonally up to the next scan line from (x,y), we get to the frame-buffer address of (x+1,y+1) with the calculation

$$\text{addr}(x+1,y+1) = \text{addr}(x,y) + x_{\max} + 2$$

Where the constant $x_{\max} + 2$ is precomputed once for all line segments. Methods implementing the setPixel procedure to store pixel intensity values depend on the capabilities of a particular system and the design requirements of the software package.

5. CIRCLE GENERATING ALGORITHM

Properties of Circles

The set of points in the circumference of the circle are all at equal distance r from the centre (x_c, y_c) and its relation is given by Pythagorean theorem as

$$(x-x_c)^2 + (y-y_c)^2 = r^2$$

The points in the circumference of the circle can be calculated by unit increments in the x direction from $x_c - r$ to $x_c + r$ and the corresponding y values can be obtained as

$$Y = y_c \pm \sqrt{r^2 - (x-x_c)^2}$$

The major problem here is that the spacing between the points will not be same. It can be adjusted by interchanging x and y whenever the absolute value of the slope of the circle is greater than 1. The unequal spacing can be eliminated by using polar coordinates and is given by

$$X = x_c + r \cos \theta$$

$$Y = y_c + r \sin \theta$$

The major problem in the above two methods is the computational time. The computational time can be reduced by considering the symmetry of circles. The shape of the circle is similar in each quadrant. Thinking one step further shows that there is symmetry between octants too.

Midpoint circle algorithm

To simplify the function evaluation that takes place on each iteration of our circle drawing algorithm, we can use Midpoint circle algorithm. The equation of the circle can be expressed as a function as given below

$$f(x, y) = x^2 + y^2 - r^2$$

If the point is inside the circle then $f(x, y) < 0$ and if it is outside then $f(x, y) > 0$ and if the point is in the circumference of the circle then $f(x, y) = 0$.

Thus the circle function is the decision parameter in the midpoint algorithm.

Assume that we have just plotted (x_k, y_k) , we have to decide whether to point (x_{k+1}, y_k) or (x_{k+1}, y_{k-1}) nearer to the circle. Now we consider the midpoint between the points and define the decision parameter as

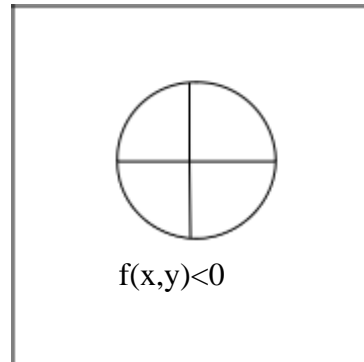


Fig 5.1 Midpoint circle

$$\begin{aligned}
 P_k &= f(x_{k+1}, y_k - 1/2) \\
 &= (x_{k+1})^2 + (y_k - 1/2)^2 - r^2 \text{ Similarly} \\
 P_{k+1} &= f(x_{k+1} + 1, y_{k+1} - 1/2) \\
 &= ((x_{k+1} + 1))^2 + (y_{k+1} - 1/2)^2 - r^2
 \end{aligned}$$

Now by subtracting the above two equations we get

$$P_{k+1} - P_k = 2(x_{k+1} + 1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) - 1$$

Where y_{k+1} is either y_k or y_{k+1} depending on the sign of p_k .

The initial decision parameter is obtained by evaluating the circle function at the starting position

$$(x_0, y_0) = (0, r)$$

$$P_0 = f(1, r - 1/2)$$

$$= 1 + (r - 1/2)^2 - r^2 \text{ (or) } P_0 = 5/4 - r$$

Hence the algorithm for the first octant is as given below

1. Calculate p_0

2. $k=0$

3. While

a) if $p_k < 0$ then plot pixel (x_{k+1}, y_{k-1}) and find the next decision parameter as

$$P_{k+1} = P_k + 2x_{k+1} + 1$$

b) Else plot pixel (x_{k+1}, y_{k-1}) and find the next decision parameter as $P_{k+1} = P_k +$

$$2x_{k+1} + 1 - 2y_{k+1}$$

c) $k=k+1$

Where $2x_{k+1}=2x_{k+2}$ and $2y_{k+1}=2y_{k-2}$

6. ELLIPSE - GENERATING ALGORITHM (Properties of ellipse)

An ellipse is defined as the set of points such that the sum of the distances from two fixed positions (foci) is the same for all points. If the distances to the two foci from any point $P = (x, y)$ on the ellipse are labeled d_1 and d_2 then, the general equation of an ellipse can be stated as

$$d_1 + d_2 = \text{constant}$$

Let the focal coordinates be $F_1 = (x_1, y_1)$ and $F_2 = (x_2, y_2)$. Then by substituting the value of d_1 and d_2 we will get

$$\sqrt{(x - x_1)^2 + (y - y_1)^2} + \sqrt{(x - x_2)^2 + (y - y_2)^2} = \text{Constant}$$

The general equation of the ellipse can be written as

$$Ax^2 + By^2 + Cxy + Dx + Ey + F = \text{Constant}$$

Where the coefficients A, B, C, D, E, and F are evaluated in terms of the focal coordinates and the dimensions of the major and minor axis of the ellipse.

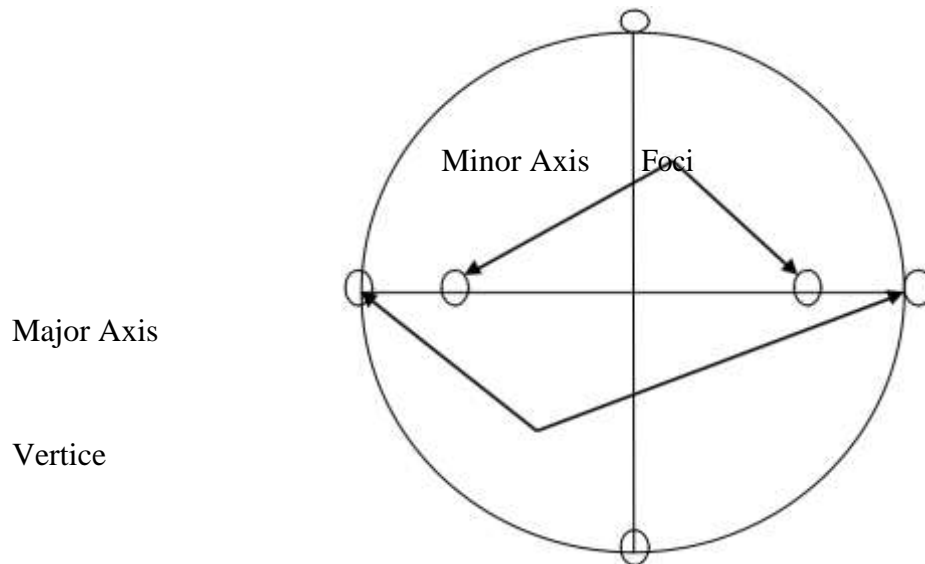


Fig. Dimension of Ellipse

If the major and the minor axis are aligned in the directions of x-axis and y-axis, then the equation of ellipse can be given by

$$\left(\frac{x - x_c}{r_x} \right)^2 + \left(\frac{y - y_c}{r_y} \right)^2 = 1$$

Where r_x and r_y are the semi-major and semi-minor axis respectively. The polar equation of the ellipse can be given by

$$X = x_c + r_x \cos \theta$$

$$Y = y_c + r_y \sin \theta$$

Midpoint Ellipse Algorithm

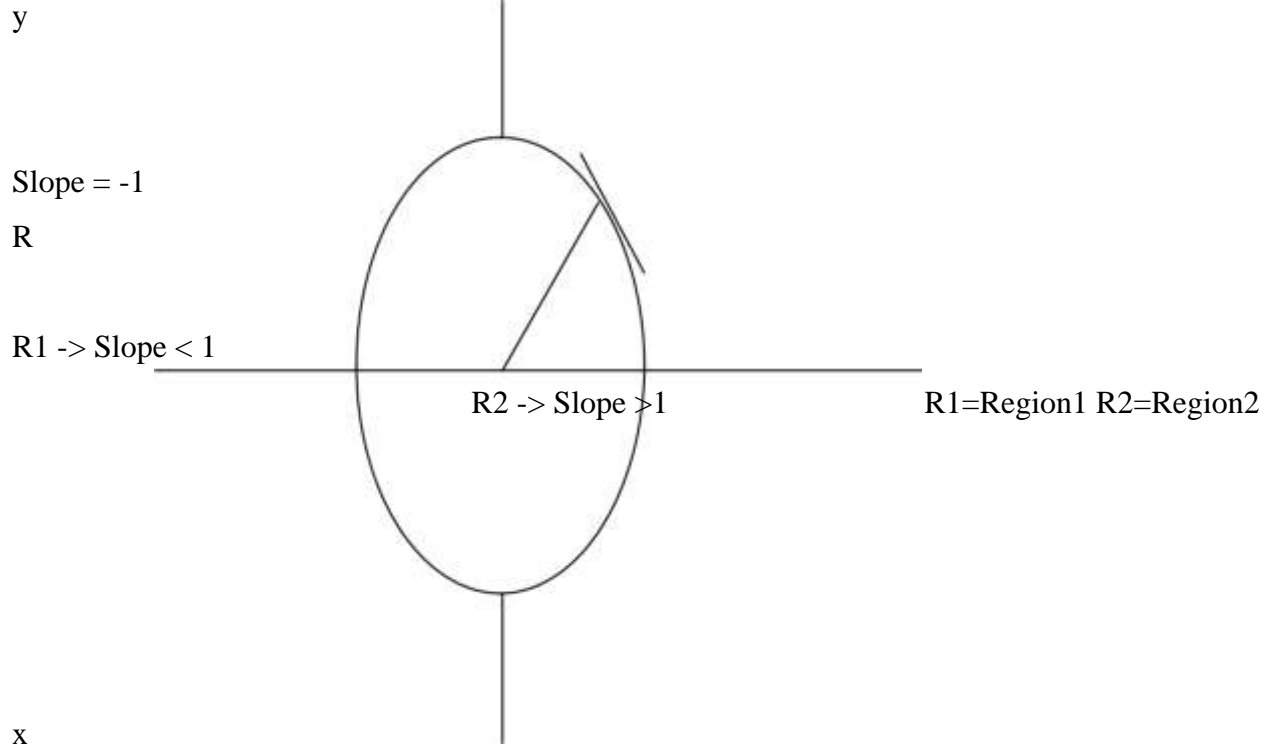


Fig. Ellipse processing regions

To calculate pixel positions for 2 regions, we start the position $(0, r_y)$ and select the points in clockwise, shifting from unit step in x to unit step in y when slope is < -1 , alternatively start at position $(r_x, 0)$ and select points in counter clockwise, shifting from unit step in y to unit step in x when slope is > 1

Function of ellipse is

$$f_{\text{ellipse}}(x,y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2 \quad (1)$$

Which following properties

$f_{\text{ellipse}}(x,y) < 0$, if (x, y) is inside the ellipse boundary.

$= 0$, if (x, y) is on the ellipse boundary.

> 0 , if (x, y) is outside the ellipse boundary.

Starting at $(0, r_y)$, take unit step in x direction. Then the slope is calculated from equation 1 as

$$dy/dx = -2r_y^2 x / 2r_x^2 y \quad \text{-----}(2)$$

At the boundary between R1 and R2, $dy/dx = -1$

$$2r_y^2 x = 2r_x^2 y$$

We can move region1 when ever

$$2r_y^2 x \geq 2r_x^2 y \quad \text{-----}(3)$$

Assume position (x_k, y_k) , determine next position by evaluating decision parameter at this midpoint from equation 1

$$P1_k = f_{\text{ellipse}}(x_k + 1, y_k - 1/2)$$

$$= r_y^2 (x_k + 1)^2 + r_x^2 (y_k - 1/2)^2 - r_x^2 r_y^2 \quad \text{-----}(4)$$

If $p1_k < 0$, mid point is inside the ellipse and pixel on scan line y_k is closer to the ellipse boundary.

Otherwise midpoint position is outside or on the ellipse boundary and pixel on scan line y_k -

1. The next decision parameter for region1 is

$$P1_{k+1} = f_{\text{ellipse}}(x_{k+1} + 1, y_{k+1} - 1/2)$$

$$= r_y^2 [(x_k + 1) + 1]^2 + r_x^2 (y_{k+1} - 1/2)^2 - r_x^2 r_y^2$$

$$P1_{k+1} = p1_k + 2r_y^2 (x_k + 1) + r_y^2 + r_x^2 [(y_{k+1} - 1/2)^2 - (y_k - 1/2)^2] \quad \text{--5}$$

Where y_{k+1} are either y_k or $y_k - 1$ depending on the sign of $p1_k$. Decision parameter are incremented by the following amounts.

$$2r_y^2 x_{k+1} + r_y^2, \text{ if } p1_k < 0$$

$$2r_y^2 x_{k+1} + r_y^2 - 2r_x^2 y_{k+1}, \text{ if } p1_k \geq 0$$

$$2r_y^2 x = 0$$

$$2r_x^2 y = 2r_x^2 r_y$$

In region1 the starting position of the ellipse is $(x_0, y_0) = (0, r_y)$

$$P1_0 = f_{\text{ellipse}}(1, r_y - 1/2)$$

$$= r_y^2 + r_x^2 (r_y - 1/2) - r_x^2 r_y^2 \quad P1_0 = r_y^2 - r_x^2 r_y + 1/4 r_x^2$$

For R2, take unit steps in the negative y direction. The decision parameter for R2 is evaluated as

$$P2_k = f_{\text{ellipse}}(x_k + 1/2, y_k - 1)$$

$$= r_y^2 (x_k + 1/2)^2 + r_x^2 (y_k - 1)^2 - r_x^2 r_y^2$$

If $p2_k > 0$, the midpoint position is outside the ellipse boundary, select pixel at x_k . If $p2_k \leq 0$, the midpoint is inside or on the ellipse boundary, select pixel at x_{k+1} . To determine the next decision parameter

$$P2_{k+1} = f_{\text{ellipse}}(x_{k+1} + 1/2, y_{k+1} - 1)$$

Where $y_{k+1} - 1 = y_k - 2$

$$= r_y^2 (x_{k+1} + 1/2)^2 + r_x^2 [(y_k - 1) - 1]^2 - r_x^2 r_y^2$$

$$P2_{k+1} = p2_k - 2 r_x^2 (y_k - 1) + r_x^2 + r_y^2 [(x_{k+1} + 1/2)^2 - (x_k + 1/2)^2] \quad \text{With } x_{k+1} \text{ set either to } x_k$$

or x_{k+1} depending on the sign of $p2_k$. Initial position in counter clockwise order starting at $(r_x, 0)$ then continue up to R1.

ATTRIBUTES OF OUTPUT PRIMITIVES

7. LINE ATTRIBUTES

Basic attributes of straight line segment are

- Line type
- Line width
- Line color

In some graphics packages, lines can also be displayed using selected pen or brush options.

Line type

Line types are

Solid lines

- Dashed lines
- Dotted lines

Solid lines are to be generated by setting the length and spacing along the line path.

Dashed line could be displayed by generating an inter dash spacing that is equal to the length of the solid sections. Length of the dashes and the inter dashes spacing are specified as user options.

A **dotted line** can be displayed by generating very short dashes with the spacing equal to or greater than the dash size.

To set line type attribute in a program, a user invokes the function setLinetype (lt)

Where parameter 'lt' is assigned a positive integer value of 1, 2, 3 or 4 to generate a lines that is solid, dashed, dotted or dash dotted.

Raster line algorithm display line type attributes by plotting pixel spans. Pixel counts for the span length and inter span spacing can be specified in a pixel mask.

Pixel mask is a string containing the digits '1' and '0' to indicate which positions to plot along the line path. The mask 1111000 for instance could be used to display a dashed line with a dash length of 4 pixels and an inter dash spacing of 3 pixels.

Plotting dashes with a fixed number of pixel results in unequal length dashes for different line orientations.



(a)

(b)

Fig. Unequal length dashes displayed with the same number of pixels.

From fig 7.1 both dashes plotted with 4 pixels but the diagonal dash is longer by factor of $\sqrt{2}$. For drawings dash length should remain approximately constant for any line orientation.

To accomplish this, we can adjust the pixel counts for the solid spans and inter span spacing according to line slope. We can display approximately equal length dashes by reducing the diagonal dash to three pixels.

Line width

Implementation of line width options depends on the capabilities of the output device. A heavy line on a video monitor could be displayed as adjacent parallel lines, while a pen plotter might require pen changes.

In PHIGS attributes, a line width command is used to set the current line width values in the attribute list. This value is then used by line drawing algorithm to connect the thickness of lines generated with subsequent output primitive command. We set the line width attribute with the command

`setLinewidthScaleFactor (1w)`

Line width parameter '1w' is assigned a positive number to indicate the relative width of the line to be displayed. A value of '1' specifies a standard width line.

On a pen plotter, for instance, a user could set '1w' to value of '0.5' to plot a line whose width is half that of the standard line. Values greater than '1' produce lines thicker than the standard.

For raster implements, a standard width line single pixels at each sample position. Other width lines are displayed as positive integer multiples of standard line by plotting additional pixel along adjacent parallel line paths.

For lines with slope magnitude less than '1' we can modify a line drawing routine to display thick lines by plotting a vertical span of pixels at each x position along the line.

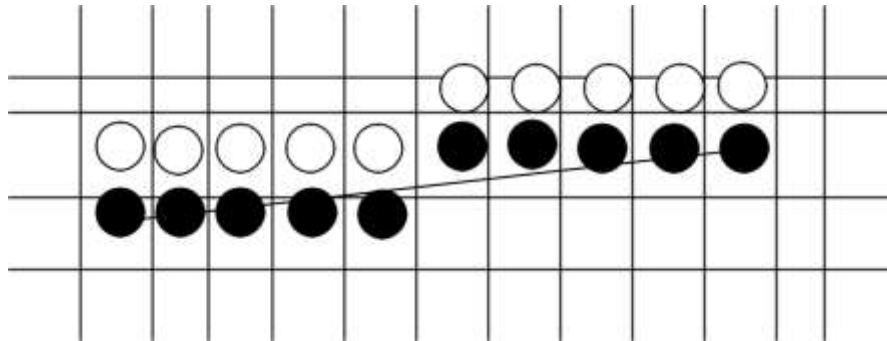


Fig. Double wide raster line with slope <1 generated with vertical spans.

For Lines with slope magnitude greater than 1, we can plot thick lines with horizontal spans, alternatively fixing up pixels to the right and left of the line path.

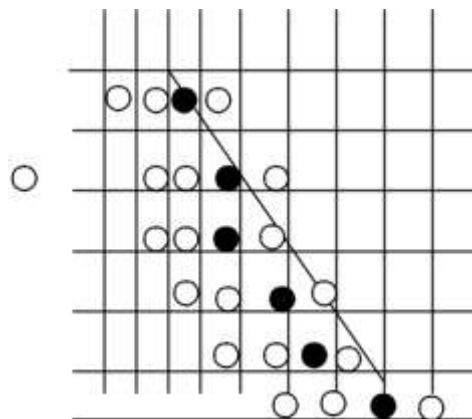


Fig. Raster line width slope >1 and line width parameter $1w=4$ plotted with horizontal pixel spans.

The problem of using horizontal and vertical spans for a lines produce very thick lines. To adjust the shape of the line ends to given the better appearance by adding line caps. Three kind of line caps are

- Butt cap
- Round cap
- Projecting square cap

Butt cap:

Butt cap obtained by adjusting the end positions of the component parallel lines so that the thick line is displayed with square ends that are perpendicular to the line path. If the specified line has slope m the square end of the thick line has slope $1/m$.

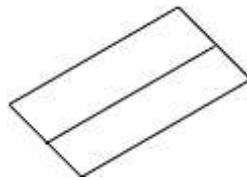


Fig. Butt cap

To obtain a rectangular representation for the line boundary, we calculate the position of the rectangle vertices along perpendiculars to the line path. So that the vertex coordinates are displayed from the line end-point by one-half the line width.

Round cap:

Round cap obtained by adding a filled semicircle to each butt cap. The circular arcs are centered on the line end points and have a diameter equal to the line thickness.

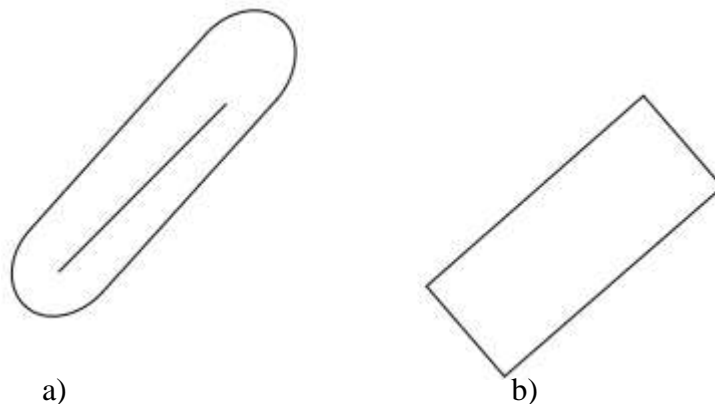


Fig. a) Round cap b) Projecting square cap

Projecting square cap:

Simply extend the line and add butt caps that are positioned one half of the line width beyond the specified end points. To generate thick polylines that is smoothly joined at the cost of additional processing at the segments end points. Three possible methods for smoothly joining two line segments are

- Miter join
- Round join
- Bevel join

Miter joins:

It is accomplished by extending the outer boundaries of each of the two lines until they meet.

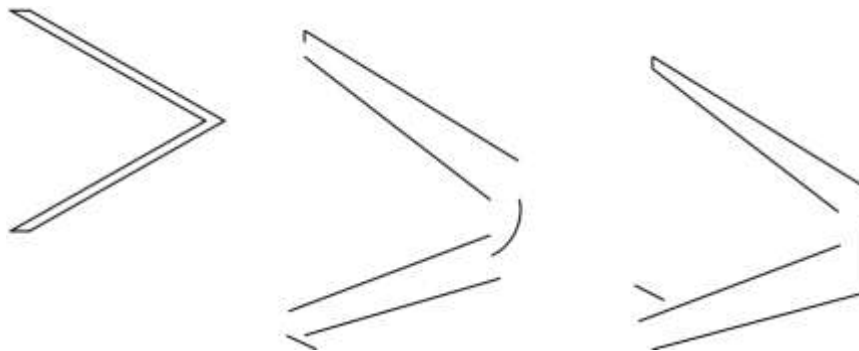


Fig.Miter joins, round join and bevel join

If the angle between two connected line segments is very small, a miter join can generate a long spike that distorts the appearance of the poly line.

Round join:

A Round join is produced by capping the connection between the two segments with a circular boundary whose diameter is equal to the line width.

Bevel join:

Bevel join is generated by displaying the line segments with butt caps and filling in the triangular gap where the segments meet.

Pen and brush options:

Lines can be displayed with pen or brush selections. The pen or brush options include

- Shapes
- Size
- Pattern

Pen or brush shapes can be stored in a pixel mask that identifies the array of pixel positions that are set along line path. To avoid setting pixels more than once in the frame buffer, we accumulate the horizontal spans generated at each position of the mask. Lines generated with pen or brush shapes can be displayed in various widths by changing the size of the mask. For example, the rectangular pen line could be narrowed with 2*2 rectangular masks or widened with a 4*4 mask.

Line color

When a system provides color options, a parameter giving the current color index is included in the list of system-attribute values. A polylines routine displays a line in the current color by setting this color value in the frame buffer at pixel locations along the line path using setPixel procedure.

The number of color choices depends on the number of bits available per pixel in the frame buffer. The function for the line color is

setPolylineColorIndex(1c)

Parameter 'lc' is a color choice and it should be a non-negative integer values.

Example

```
setLineType(2);  
setLinewidthScaleFactor(2);  
setPolylineColorIndex(5);  
PolyLine (n1,wcpoints1);  
setPolylineColorIndex(6);  
PolyLine (n2,wcpoints2);
```

8. CURVE ATTRIBUTES

We can display varying colors, widths, dot dash curves with patterns and available pen or brush options. Parameters for curve attributes are the same as those for line segments. The pixel mask options are also used in raster curve algorithm to generate dashed and dotted patterns.

We can generate the dashes in various octants using circle symmetry. We must shift the pixel positions to maintain correct sequence of dashes and spaces as we move from one octant to the next.

To display constant length dashes, we need to adjust the number of pixels plotted in each dash around the circle circumference. Instead of applying a pixel mask with constant spans, we plot a pixel along equal angular arcs to produce equal length dashes.

Color Code	R	G	B	Displayed color
0	0	0	0	Black
1	0	0	1	Blue
2	0	1	0	Green
3	0	1	1	Cyan
4	1	0	0	Red
5	1	0	1	Magenta
6	1	1	0	Yellow
7	1	1	1	White

Fig. Color codes for a three – bit per pixel frame buffer

Raster curves of displayed using the vertical pixel spans. If curve slope is less than If the magnitude is the horizontal spans. Various widths can be method of horizontal or the magnitude of the 1, we plot vertical spans. greater than 1 we plot.

Using circle symmetry, we generate the circle path with vertical spans in the octant from $x=0$ to $x=y$, and then reflect pixel position about the line $y=x$. Circle sections in the other quadrants are obtained by reflecting pixel positions in the first quadrant about the coordinate axes. According to the function of curve slope the thickness of curve will be displayed. For displaying thick curves between two parallel curve paths, whose separation distance is equal to the desired width. Using the curve path as one boundary and setting up the second boundary either inside or outside the original curve path.

We can maintain the original curve position by setting two boundaries curved at a distance of one half the widths on either side of the specified curve path. Example Radius =16, width =4. The boundary arcs are then set at a separation distance of 2 on either side. Pen or brush is used to display the curves. Using rectangular shape pen curves will be displayed thicker, where the magnitude of the curve slope is 1. A uniform curve thickness can be displayed by rotating

the rectangular pen to align it with the slope direction as we move around the curve or by using a circular pen shape. Curves drawn with pen and brush shapes can be displayed in different sizes.

9. COLOR AND GRAY SCALE LEVELS

Depending on the capabilities and design objectives of a particular system, the various colors and intensity level options can be made. Color options are numerically coded with values ranging from 0 through the positive integers.

For CRT monitors, these color codes are then converted to intensity level settings for the electron beams. In color raster system the number of color choices available depends on the amount of storage provided per pixel in the frame buffer.

Two ways to store the color information in the frame buffer are

- We can store color codes directly in the frame buffer
- We can put the color codes in a separate table and use pixel values as an index into this table.

When ever a particular color code is specified in an application program, the corresponding binary value is placed in the frame buffer for each component pixel in the output primitives to be displayed in that color.

From table each of the 3 bit positions is used to control the intensity level of the corresponding electron gun in an RGB monitor. The left most bit controls the red gun, the middle bit controls the green gun, and the right most bit controls the blue gun. Adding more bits per pixel to the frame buffer increases the number of color choices.

With 6 bits per pixel, 2 bits can be used for each gun. This allows four different intensity settings for each of the three color guns. With a resolution of 1024 by 1024, a full color RGB system needs 3 megabytes of storage for the frame buffer.

Color Tables

Frame buffer values are now used as indices into the color table. In this example, each pixel can reference any one of the 256 table positions, and each entry in the table uses 24 bits to specify an RGB color.

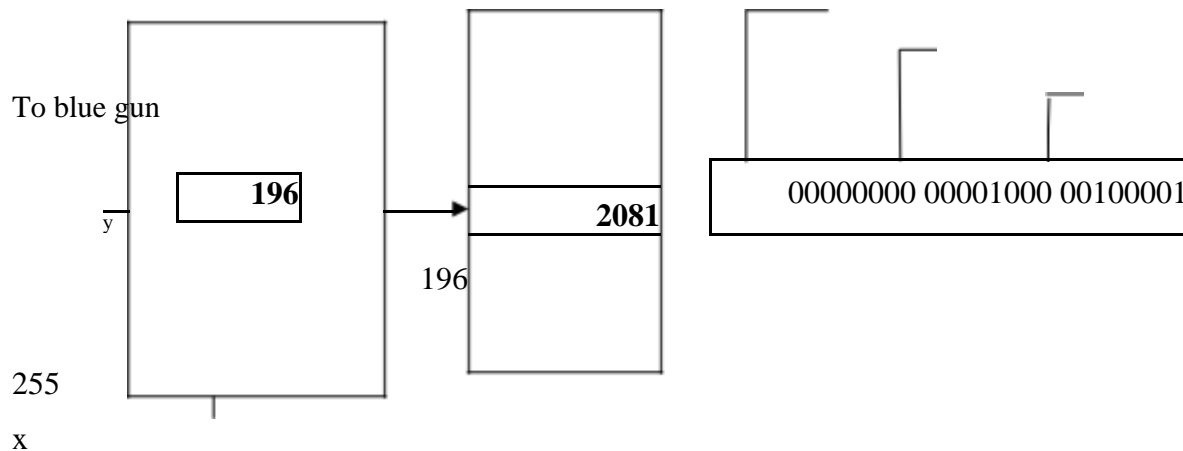


Fig. Storing color values in a color lookup table

For the color code 2081, a combination green blue color is displayed for pixel location(x, y). When comparing this to full color system this scheme reduces the number of simultaneous colors that can be displayed. It also reduces the frame buffer storage requirements. In PHIGS application a user can set color entries

```
setColorRepresentation(ws, ci, colorptr)
```

Parameter ws represents work station output device, ci represents color index, which is the color table position number (0, 255), colorptr points to trio of RGB color values, range from 0 to 1

ws=2

Ci	Color
0	(0, 0, 0)
1	(0, 0, 0.2)
-	-----
-	-----
192	(0, 0.3, 0.13)

ws=1

Ci	Color
0	(1,1,1)
1	(0.9, 1, 1)
2	(0.8, 1,1)
-	-----
-	-----

Fig. Work station color tables Advantages of using color lookup table

- Without requiring large frame buffers we can provide reasonable number of colors.
- Allowing a user to experiment with different color combinations in design, scene or graph.
- Without changing pixel values we can store visualization application values.

Gray scale

With monitors that have no color, capacity color functions can be used in an application program to set the shades of gray or gray scale. Numeric values over the range from 0 to 1 can be used to specify gray scale levels.

Intensity codes for a 4 level gray scale system

Intensity codes	Stored intensity values in the frame buffer (binary values)	Displayed Grayscale
0.0	0	(0, 0)
0.33	1	(0,1)
0.67	2	(1, 0)
1.0	3	(1,1)

Fig. Intensity codes for a four – level grayscale system

In this example, any intensity input value near 0.33 would be stored as binary value 01 in the frame buffer and it would be displayed as dark gray. If additional bits per pixel are available in the frame buffer, the value of 0.33 mapped to the nearest level.

With 3 bits per pixel we accommodate 8 gray levels, while 8 bpp 256 shades of gray. Alternate scheme for storing the intensity code directly to the voltage value that produces this gray scale level on output device.

$\text{Intensity} = 0.5[\min(r, g, b) + \max(r, g, b)]$.

10. AREA FILL ATTRIBUTES

Options for filling a defined region include a choice between a solid color or a patterned fill and choices for the particular colors and patterns. These fill options can be applied to polygon regions or to area defined with curved boundaries. Areas can be painted using various brush styles, colors and transparency parameters.

Fill styles

There are four basic fill styles

- Hollow
- Solid
- Patterned
- Hatch

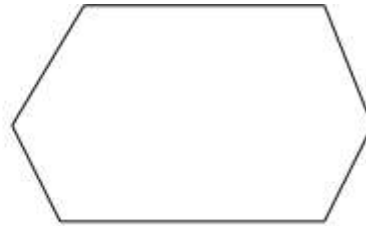
The function for the fill style in PHIGS program is `setInteriorStyle(fs)`

Values for the fill style parameter `fs` include hollow, solid, and pattern, hatch. This parameter normally applied for to polygon areas and can also implemented to fill regions with curved boundaries.

Hollow areas are displayed using only the boundary line outline, with interior color the same as the background color. A **solid** fill is displayed in a single color up to and including the borders of the region.



a) Solid



b) hollow



c) pattern

Fig. Polygon fill styles

Hatch is used to fill an area with selected hatching patterns that is parallel lines or crossed lines. The color for a solid interior or for a hollow area outline is chosen with `setInteriorColorIndex (fc)`

Where fill color parameter `fc` is set to the desired color code. Fill options include specifications for the edge type, edge width and edge color of region. These attributes are set independently of the fill style or fill color.

Pattern Fill

We select fill patterns with `setInteriorStyleIndex (pi)`. Where pattern index parameter `pi` specifies a table position.

For example, the following set of statements would fill the area defined in the fillArea command with the second pattern type stored in the pattern table:

```
setInteriorStyle(pattern);
```

```
setInteriorStyleIndex(2);
```

```
fillArea(n, points);
```

For fill style pattern, table entries can be created on individual output devices with

```
setPatternRepresentation (ws, pi, nx, ny, cp)
```

Parameter pi sets the pattern index number for workstation code ws and cp is a two dimensional array of color codes with nx columns and ny rows. The following program segment illustrates how this function could be used to set the first entry in the pattern table for work station.

```
cp[1,1] := 4                                cp[2,2] := 4
```

```
cp[1, 2] := 0                                cp[2,1] := 0
```

```
setPatternRepresentation (1, 1, 2, 2, cp);
```

When a color array cp is to be applied to fill a region, we need to specify the size of the area that is to be covered by each element of the array. To set the size for the rectangular pattern

```
setPatternSize(dx, dy)
```

Index	Pattern		
(pi)	(cp)		
1	4	0	
	0	4	
2	2	1	2
	1	2	1
	2	1	2

Fig. A work station pattern table with two entries using the color code

Where parameter dx and dy give the coordinate width and height of the array mapping.

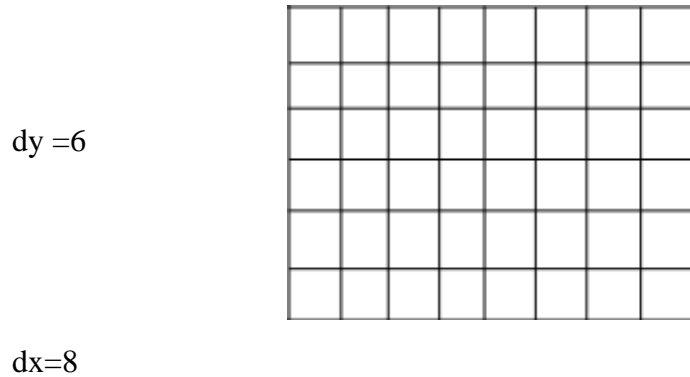


Fig. A pattern array with 4 columns and 3 rows mapped to an 8 by 6 coordinate rectangle

If the values for dx and dy in this fig 10.3 are given in screen coordinates, then each element of the color array would be applied to a 2 by 2 screen grid containing four pixels. A reference position for starting a pattern fill is assigned with statement

setPatternReferencePoint(position)

Parameter position is a pointer to coordinates (xp, yp) that fix the lower left corner of the rectangular pattern. From this starting position, the pattern is then replicated in the x and y directions until the defined area are covered. The process of filling an area with a rectangular pattern is called **tiling** and rectangular fill patterns referred to as **tiling patterns**.

Soft Fill

Modified boundary-fill and flood-fill procedures that are applied to repaint areas. So that the fill color is combined with the background colors are referred to as **soft-fill** or **tint-fill**.

Uses For this fill are

- To soften the fill colors at object borders that have been blurred to antialias the edges.
- To allow repainting of a color area that was originally filled with semi transparent brush, where the current color is then mixture of the brush color and the background color behind the area.

As an example of this type of fill, the linear soft-fill algorithm repaints an area that was originally painted by merging a foreground color F with a single background color B, where $F \neq B$. The current RGB color P of each pixel within the area to be refilled is some linear combination of F and B:

$$P = tF + (1-t)B \quad \text{-----}(1)$$

Where the ‘transparency’ factor t has a value between 0 and 1 for each pixel. For values of t less than 0.5, the background color contributes more to the interior color of the region than does the fill color.

Each RGB components of the colors, with

$$P = (P_R, P_G, P_B), \quad F = (F_R, F_G, F_B), \quad B = (B_R, B_G, B_B) \quad \text{-----}(2)$$

We can thus calculate the value of parameter t using one of the RGB color components as

$$T = (P_k - B_k) / (F_k - B_k) \quad \text{-----}(3)$$

Where $k = R, G, B$; and $F_k \neq B_k$

This value of t is then used to mix the new fill color NF with the background color, using either a modified flood-fill or boundary- fill procedure. When two background colors B_1 and B_2 are mixed with foreground color F, the resulting pixel color P is

$$P = t_0 F + t_1 B_1 + (1 - t_0 - t_1) B_2$$

Where the sum of the coefficients t_0 , t_1 , and $(1 - t_0 - t_1)$ on the color terms must equal to 1. These parameters are then used to mix the new fill color with two background colors to obtain the new pixels color.

11. CHARACTER ATTRIBUTE

1. The appearance of displayed characters is controlled by attributes such as font,
2. size, color, and orientation.
3. Attributes can be set both for entire character strings (text) and for individual characters defined as marker symbols.
4. There are a great many text options that can be made available to graphics programmers.
5. The characters in a selected font can also be displayed with assorted underlining styles
6. (double), in boldface, in *italics*. and in outline or shadow styles.
7. A particular font and associated style is selected in a PHIGS program by setting an integer
8. Code for the text font parameter t f in the function

Set text font (tf)

- Font options can be made available predefined sets of grid patterns or as character sets designed with polylines and spline curves.
- Color settings for ,displayed text are stored m the system attribute list and used by the procedures that load character definitions into the frame buffer.
- When a character string is to be displayed, the current (color i; used to set pixel values in the frame huffer corresponding to the character shapes and positions
- Control of text color (or intensity) is managed from an application program with

Set TextColorIndex (tc)

We can adjust text t size by scaling the overall dimensions (height and width) of characters or by scaling only the character width.Character size is specified by printers and compositors in points, where 1 point is 0.013837 inch depending on the design of the typeface.

The distance between the bottom line and the topline of the character body is the same for all characters in a particular size and typeface, but the body width may vary.

Proportionally spaced fonts assign a smaller body width to narrow characters such as i, j, l, and f compared to broad characters such as W or M.

Character height is defined as the distance between the baseline and the capline of characters.

Kerned characters, such as f and j in Fig 11.1 typically extend beyond the character-body limits, and letters with descenders (g, j, p, q, y) extend below the baseline. Each character is positioned within the character body by a font designer to allow suitable spacing along and between print lines when text is displayed with character bodies touching. Text size can be adjusted without changing the width-to-height ratio of character.

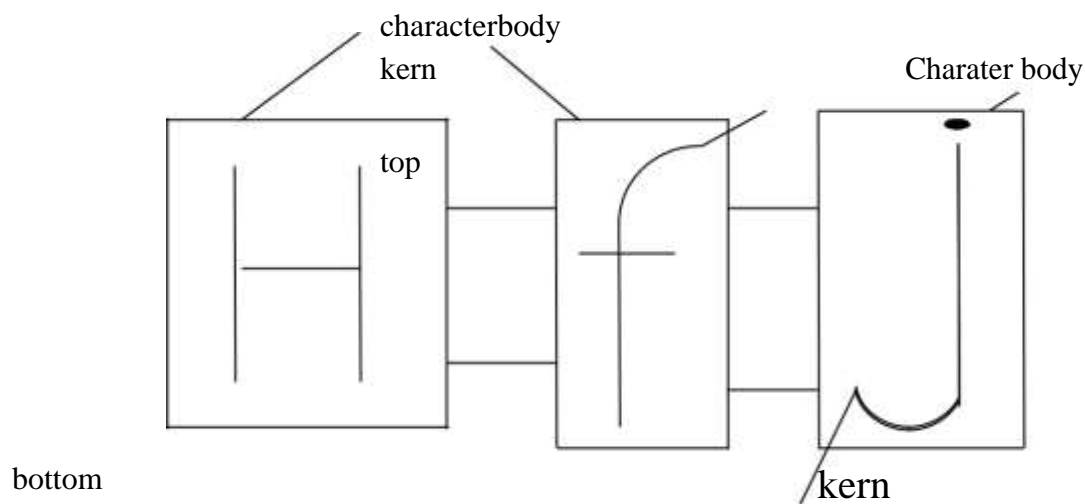


Fig. Character Body

Height 1

Height 2

Height

3

Fig. The effect of different character height settings on displayed text

Parameter `ch` is assigned a real value greater than 0 to set the coordinate height of capital letters: the distance between baseline and capline in user coordinates. This setting also affects character-body size, so that the width and spacing of characters is adjusted to maintain the same text proportions.

For instance, doubling the height also doubles the character width and the spacing between characters. Fig 11.2 shows a character string displayed with three different characters heights. The width only of text can be set with the function.

set character Expansion factor (`cw`)

Where the character-width parameter `cw` IS set ton positive real value that scales the body width of characters. Text height is unaffected by this attribute setting. Examples of text displayed with different character expansions are given in Fig 11.3.

setCharacterSpacing (`cs`)

Spacing between characters is controlled separately with where the character-spacing parameter `cs` can he assigned any real value. The value assigned to `cs` determines the spacing between character bodes along print lines.

Negative values for `cs` overlap character bodies; positive values insert space to spread out the displayed characters.

Assigning the value 0 to `cs` causes text to be displayed with no space between character bodies. The amount of spacing to be applied is determined by multiplying the value of `cs` by the character height (distance between baseline and capline).

In Fig11.4 character string is displayed with three different settings for the character-spacing parameter. The orientation for a displayed character string is set according to the direction of the character up vector:

setcharacterupvector (upvect)

width0.15 .0

width 2.0

Fig. The effect of different character-width settings on displayed text.

Spacing^{0.5}

1.0

Spacing 2.0

Fig. The effect of different character spacing on displayed text

Parameter **upvec** t in this function is assigned two values that specify the x and vector components Text is then displayed so that the orientation of characters from baseline to capline is in the direction of the up vector. For example, with **upvect** = (I, I), the direction of the up vector is 45° and text would be displayed as shown in Fig. 11.5 (a). A procedure for orienting text rotates characters so that the sides of character bodies, from baseline to capline, are aligned with the up vector. The rotated character shapes are then scan converted into the frame buffer.



Fig. (a) Up vector



Fig. (b) Controls the displayed text

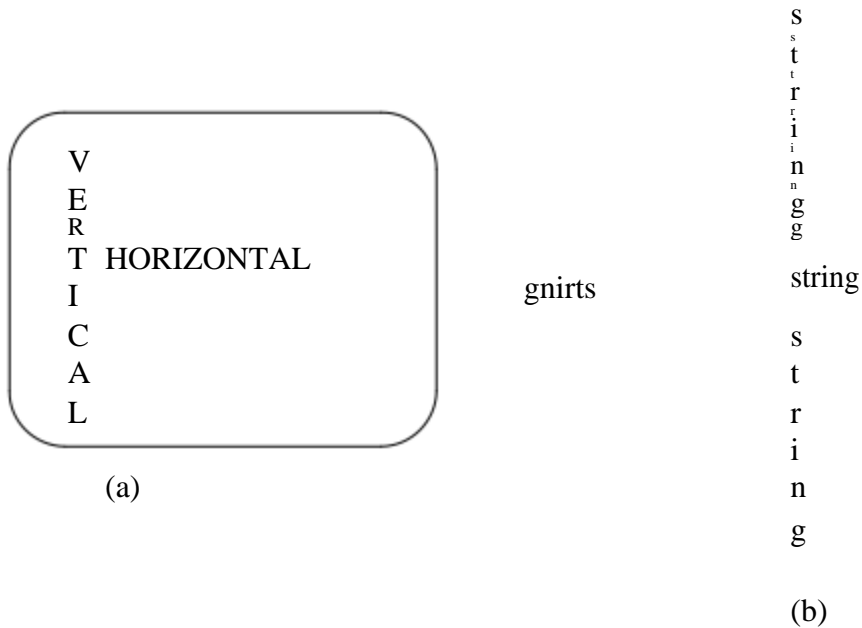


Fig. (a & b) Text path attributes to produce vertical and horizontal and text displayed with the four text – path option

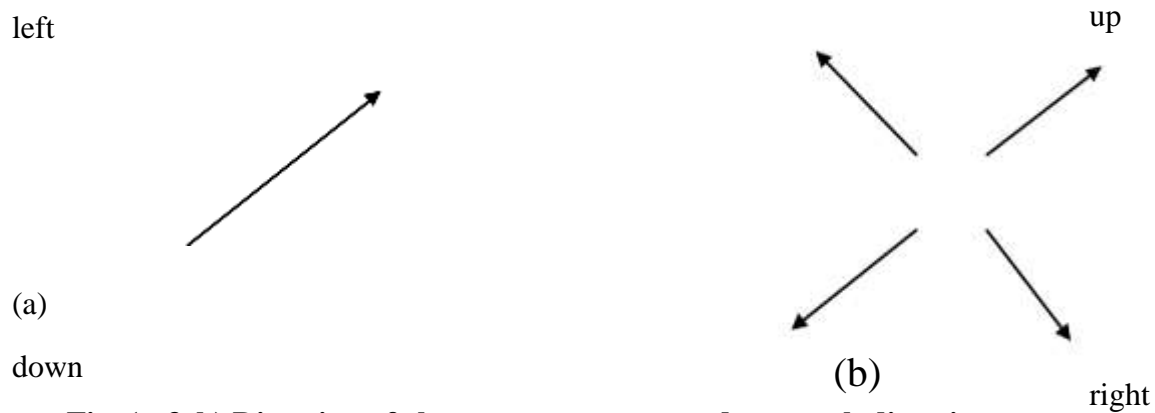


Fig. (a & b) Direction of character up vector and text path direction

Where the text-path parameter *tp* can be assigned the value: right, *left*, ***up***, or down. Examples of text displayed with these four options are shown in Fig.11.5(b). It is useful in many applications to be able to arrange character strings vertically or horizontally. Fig. 11.6(a). An attribute parameter for this option is set with the statement.

setTextPath(tp)

A procedure for implementing this option must transform the character patterns into the specified orientation before transferring them to the frame buffer. Character strings can also be oriented using a combination of up-vector and text-path specifications to produce slanted text. Fig. 11.6(b) shows the directions of character strings generated by the various text-path settings for a **45°** up vector. Examples of text generated for text-path values down and right with this up vector are illustrated in Fig. 11.8(a&b). Another handy attribute for character strings is alignment. This attribute specifies how text is to be positioned with respect to the start coordinates. Alignment attributes are set with

setTextAlingnment (h , v)

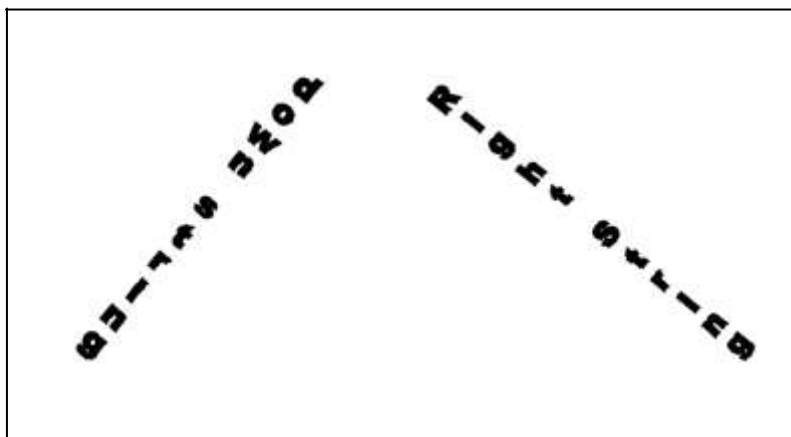


Fig. (a&b) Text generated for text-path values down and right with this up vector

Where parameters *h* and *v* control horizontal and vertical alignment, respectively. Horizontal alignment is set by assigning *h* a value of left, centre, or right. Vertical alignment is set by assigning *v* a value of top, cap, half, base, or bottom.

The interpretation of these alignment values depends on the current setting for the text path. Fig 11.9 shows the position of the alignment settings when text is to be displayed horizontally to the right or vertically down. Similar interpretations apply to text path value5 of left and up. The "most natural" alignment for a particular text path is chosen by assigning the value norm01 to the *h* and *v* parameters. A precision specification or text display is given with

setTextPrecision(*tpr*)

where text precision parameter *tpr* is assigned one of the values: string, char, or stroke. The highest-quality text is displayed when the precision parameter is set to the value stroke. For this precision setting, greater detail would be used in defining the character shapes, and the processing of attribute selections and other string-manipulation procedures would be carried out to the highest possible accuracy. The lowest-quality precision setting, string, is used for faster display of Character strings. At this precision, many attribute selections such as text path are ignored, and string-manipulation procedures are simplified to reduce processing time.

POINTS TO REMEMBER

- Digital differential analyzer is a scan conversion line algorithm based on calculating either Δx or Δy .
- The graphics programming packages provide functions to describe a scene in terms of these basic geometric structures referred to as output primitives. Another name of random scan system is vector system.
- The rounding of coordinate values to integers causes lines to be displayed with a stairstep appearance can be called as jaggies.

- Each screen point is referred to as a pixel or pel –picture element. Picture definition is stored in a memory area called the refresh buffer or frame buffer. Stored intensity values are then retrieved from the refresh buffer and painted on the screen one row scan line.
- Any parameter that affects the way a primitive is to be displayed is referred to as an attribute parameter.
- Color table is used for storing color values. Monitor doesn't have color capabilities in that case color function can be used in an application program to set the shades of gray or gray scale.
- Supplementary name of
Color look up table- video look up table.
Gray –gray scale
Pixel-pel
Soft fill-tint fill

UNIT-II: 2D Geometric Transformations: Basic Transformations – Matrix Representations – Composite Transformations – Other Transformations. 2D Viewing: The Viewing Pipeline – Viewing Co-ordinate Reference Frame – Window-to-Viewport Coordinate Transformation - 2D Viewing Functions – Clipping Operations.

2D GEOMETRIC TRANSFORMATION

1. BASIC TRANSFORMATION

Transformations are a fundamental part of computer graphics. Transformations are used to position objects, to shape objects, to change viewing positions, and even to change how something is viewed. There are 4 main types of transformations that one can perform in 2 dimensions.

- translations
- scaling
- rotation
- shearing

Representation of Points/Objects

A point **p** in 2D is represented as a pair of numbers: **p** = (x, y) where x is the x-coordinate of the point **p** and y is the y-coordinate of **p**. 2D objects are often represented as a set of points (vertices), {p₁, p₂, ..., p_n}, and an associated set of edges {e₁, e₂, ..., e_m}. An edge is defined as a pair of points e = {p_i, p_j}.

For example the three points and three edges of the triangle given here are p₁=(1,0), p₂=(1.5,2), p₃=(2,0), e₁={p₁, p₂}, e₂={p₂, p₃}, and e₃={p₃, p₁}.

Translation

A translation is applied to an object by repositioning it along a straight-line path from one coordinate location to another. Assume you are given a point at (x,y)=(2,1). Where will the point be if you move it 3 units to the right and 1 unit up? The Answer is (x',y') = (5,2). This is obtained by (x',y') = (x+3,y+1). That is, to move a point by some amount *dx* to the

right and *dy* up, you must add *dx* to the x-coordinate and add *dy* to the y-coordinate. For example to move the green triangle, represented by 3 points given below, to the red triangle we need *dx* = 3 and *dy* = -5.

greentriangle = { p1=(1,0), p2=(2,0), p3=(1.5,2) }

A translation can also be represented by a pair of numbers, $t=(t_x, t_y)$ where t_x is the change in the x-coordinate and t_y is the change in y coordinate. To translate the point p by t , we simply add to obtain the new (translated) point $p' = p + t$.

$$q = p + t = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \end{pmatrix}$$

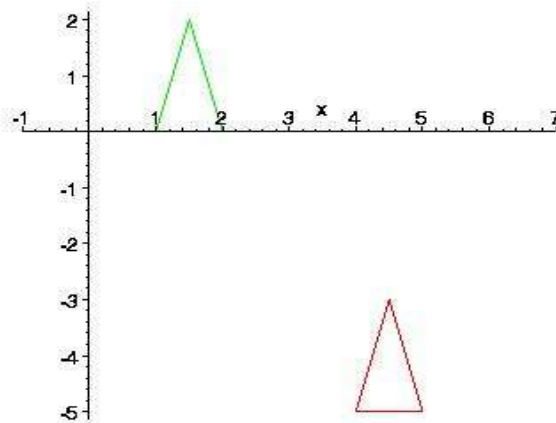


Fig. Moving a polygon

Rotation

Consider rotation of a point (x,y) with respect to origin in the anti clock wise direction. Let (x', y') be the new point after rotation and let the angular displacement (ie. Angle of rotation) be θ as shown in figure.

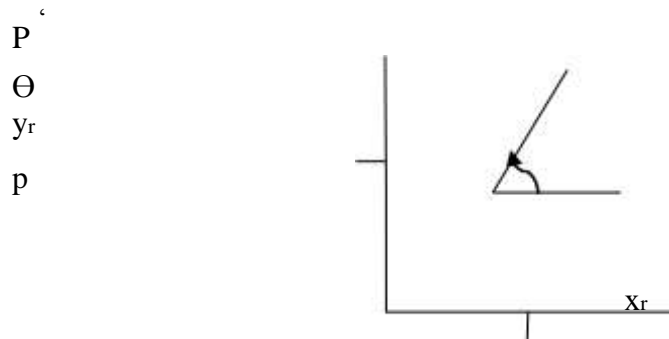


Fig. Rotation of an object through angle Θ about the pivot point (x_r, y_r)

Let ρ be the distance of the points from the origin. And let θ be the angle between x-axis and the line joining the point (x,y) to the origin. Now applying trigonometric identities, we get the following equations for (x', y')

$$x' = \rho \cos(\phi + \theta) = \rho \cos \phi \cos \theta - \rho \sin \phi \sin \theta$$

-----a

$$y' = \rho \sin(\phi + \theta) = \rho \cos \phi \sin \theta + \rho \sin \phi \cos \theta$$

Similarly for (x,y) , we get the following equation

$$x = \rho \cos \phi$$

-----b

$$y = \rho \sin \phi$$

Substituting (b) in (a), we get the equation for rotating a point with respect to origin as follows

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

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

Scaling

Suppose we want to double the size of a 2-D object. Double in size, width only, height only, along some line only, scaling we usually mean some amount of scaling along each dimension. That is, we must specify how much to change the size along each dimension. Below we see a

triangle have been doubled in both width and height (note, the area is more than doubled). Scaling factor s_x and s_y to produce the transformed coordinates (x', y') .

$$x' = x \cdot s_x \text{ and } y' = y \cdot s_y$$

Scaling factor s_x scales object in x direction while s_y scales in the y direction.

$$P' = S \cdot P$$

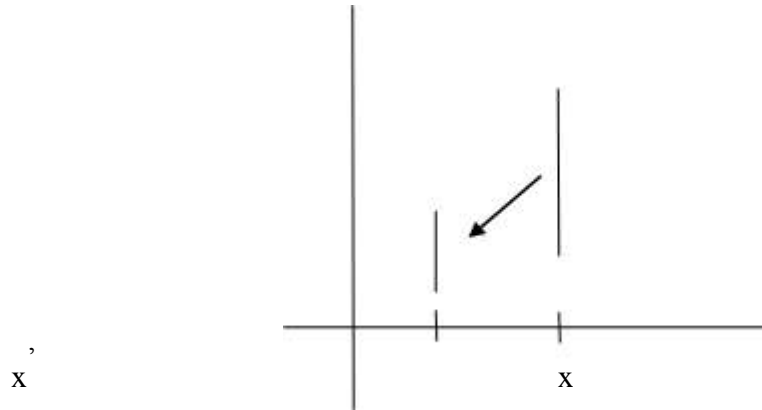


Fig. A line scaled with the equation $P' = S \cdot P$

The scaling for the x dimension does not have to be the same as the y dimension. If these are different, then the object is distorted.

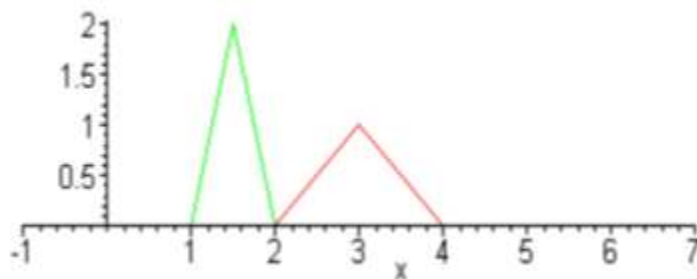


Fig. Scaling relative to the fixed point

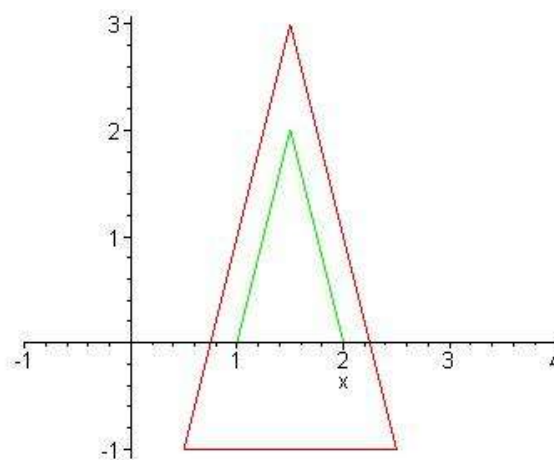
In the Fig, the scaled object is always shifted to the right. This is because it is scaled with **respect to the origin**. That is, the point at the origin is left fixed. Thus scaling by more than 1

move the object away from the origin and scaling of less than 1 move the object toward the origin.

This is because of how basic scaling is done. The above objects have been scaled simply by multiplying each of its points by the appropriate scaling factor. For example, the point $p = (1.5, 2)$ has been scaled by 2 along x and 0.5 along y. Thus, the new point is $q = (2*1.5, 5*2) = (1, 1)$.

Scaling about a Particular Point

If we want to scale the objects their center as show below:



Let the fixed point (x_f, y_f) be the center of the object, then the equation for scaling with respect to (x_f, y_f) is given by

$$x' = x_f + (x - x_f) S_x$$

$$y' = y_f + (y - y_f) S_y$$

Combining Transformations

We saw that the basic scaling and rotating transformations are always with respect to the origin. To scale or rotate about a particular point (the fixed point) we must first translate the object so that the fixed point is at the origin. We then perform the scaling or rotation and then the inverse

of the original translation to move the fixed point back to its original position. For example, if we want to scale the triangle by 2 in each direction about the point

$fp = (1.5, 1)$, we first translate all the points of the triangle by $T = (-1.5, 1)$, scale by 2 (S), and then translate back by $-T = (1.5, 1)$. Mathematically this looks like

$$q = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} -1.5 \\ 1 \end{bmatrix} + \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}$$

2. MATRIX REPRESENTATION

In general, when you want to perform a complex transformation, you usually make it by combining a number of basic transformations. The above equation for q , however, is awkward to read because scaling is done by matrix multiplication and translation is done by vector addition. In order to represent all transformations in the same form, computer scientists have devised what are called homogeneous coordinates. Do not try to apply any exotic interpretation to them. They are simply a mathematical trick to make the representation be more consistent and easier to use.

Homogeneous coordinates (HC) add an extra virtual dimension. Thus 2D HC are actually 3D and 3D HC are 4D. Consider a 2D point $p = (x, y)$. In HC, we represent p as $p = (x, y, 1)$. An extra coordinate is added whose value is always 1. This may seem odd but it allows us to now represent translations as matrix multiplication instead of as vector addition. A translation (dx, dy) which would normally be performed as

$$q = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

is now written as

$$q = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T * P = \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Now, we can write the scaling about a fixed point as simply a matrix multiplication:

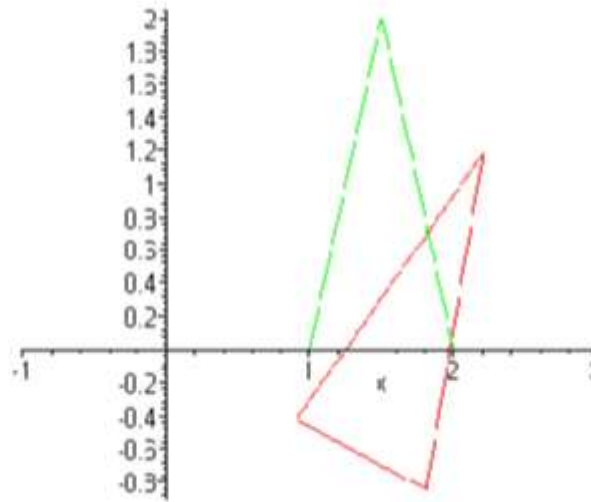
$$q = (-T) S T p = A p, \text{ Where } A = (-T) S T$$

The matrix A can be calculated once and then applied to all the points in the object. This is much more efficient than our previous representation. It is also easier to identify the transformations and their order when everything is in the form of matrix multiplication.

Matrix/Vector Representation of Rotations

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Now suppose we want to rotate an object with respect to some fixed point (x_f, y_f) as shown in the following figure. Then what will be the equation for rotation for a point with respect to the fixed point (x_f, y_f) .



The equation for rotation of a point with respect to a fixed point (x_f, y_f) can be given as

$$x' = x_f + (x - x_f) \cos \theta - (y - y_f) \sin \theta$$

$$y' = y_f + (x - x_f) \sin \theta + (y - y_f) \cos \theta$$

Matrix/Vector Representation of Scaling

Scaling transformations are represented by matrices. For example, the above scaling of 2 and 0.5 is represented as a matrix:

$$\text{Scale matrix: } S = \begin{bmatrix} S_x & 0 \\ 0 & s_y \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$\text{New Point: } P = \begin{bmatrix} S_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \times S_x \\ y \times s_y \end{bmatrix}$$

The matrix for scaling in HC is

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

and the matrix for rotation is

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

3. COMPOSITE TRANSFORMATION

Forming products of transformation matrix is often referred to as a concatenation, or composition of matrix. We can set up a matrix for any sequence of transformation as a composite transformation matrix by calculating the matrix product of individual transformations.

Translation:

If two successive translation vectors (t_{x1}, t_{y1}) and (t_{x2}, t_{y2}) are applied to a coordinate position p , the final transformed position p' is calculated as

$$\begin{aligned} P' &= T(t_{x2}, t_{y2}) \cdot \{T(t_{x1}, t_{y1})\} \cdot P \\ &= \{T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1})\} \cdot P \end{aligned}$$

Where P and P' are represented as homogeneous-coordinate column vectors.

We can verify this result by calculating the matrix product for the two associative groupings.

Also, the composite transformation matrix for this sequence of translation is

$$\begin{pmatrix} 1 & 0 & t_{x2} \\ 0 & 1 & t_{y2} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & t_{x1} \\ 0 & 1 & t_{y1} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_{x1} + t_{x2} \\ 0 & 1 & t_{y1} + t_{y2} \\ 0 & 0 & 1 \end{pmatrix}$$

Or

$$T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1}) = T(t_{x2} + t_{x1}, t_{y2} + t_{y1})$$

In which it is to demonstrate that two successive translations are additive.

Rotations

To successive rotations applied to point P produce the transformed position

$$\begin{aligned} P' &= R(\Theta_2) \cdot \{R(\Theta_1) \cdot P\} \\ &= \{R(\Theta_2) \cdot R(\Theta_1)\} \cdot P \end{aligned}$$

By multiplying the two rotation matrices, we can verify that two successive rotations are additive:

$$R(\Theta_2) \cdot R(\Theta_1) \cdot P = R(\Theta_1 + \Theta_2) \cdot P$$

So that the final rotated coordinates can be calculated with the composite rotation matrix as $P' = R(\Theta_1 + \Theta_2) \cdot P$

Scaling:

Concatenation transformation matrices for two successive scaling operation produces the following composite scaling matrix:

$$\begin{pmatrix} S_{x2} & 0 & 0 \\ 0 & S_{x1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_{x1} & 0 & 0 \\ 0 & S_{y1} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_{x1} \cdot S_{x2} & 0 & 0 \\ 0 & S_{y1} \cdot S_{y2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Or

$$S(S_{x2}, S_{y2}) \cdot S(S_{x1}, S_{y1}) = S(S_{x2} \cdot S_{x1}, S_{y2} \cdot S_{y1}) \quad \textbf{General Pivot-Point Rotation}$$

With a graphics package that only provides a rotate function for revolving objects about the coordinate origin, we can generate rotations about any selected pivot point (**x, y**,) by performing the following sequence of translate-rotate translate operations:

1. Translate the object so that the pivot-point position is moved to the coordinate origin.
2. Rotate the object about the coordinate origin.
3. Translate the object so that the pivot point is returned to its original position.

It can be expressed in the form

$$T(x_r, y_r) \cdot R(\theta) \cdot T(-x_r, -y_r) = R(x_r, y_r, \theta)$$

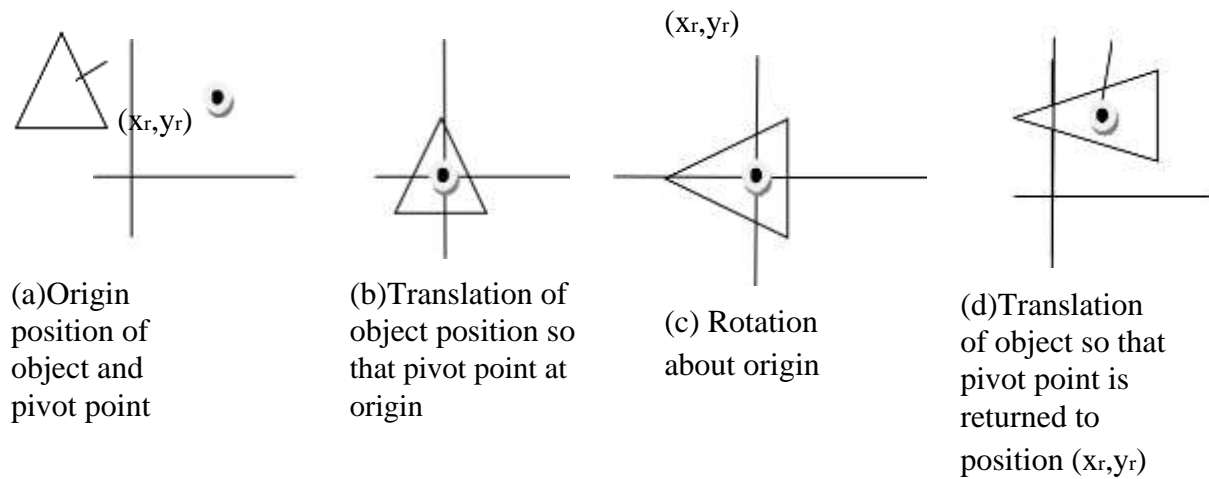


Fig. Transformation sequence of rotating object about specified pivot point General Fixed-Point scaling

1. Translate object so that the fixed point coincides with the coordinate origin.
2. Scale the object with respect to the coordinate origin.
3. Use the inverse translation of step 1 to return the object to its original position.

General scaling direction

Parameters s_x and s_y scale objects along the x and y directions. We can scale an object in other directions by rotating the object to align the desired scaling directions with the coordinate axes before applying the scaling transformation. Suppose we want to apply scaling factors with values specified by parameters S_1 and S_2 in the directions. To accomplish the scaling without changing the orientation of the object, we first perform a rotation so that the directions for s_x and s_y coincide with the x and y axes, respectively. Then the scaling transformation is applied, followed by an opposite rotation to return points to

their original orientations. The composite matrix resulting from the product of these three transformations is

$$\mathbf{R}^{-1}(\Theta) \cdot \mathbf{S}(s_1, s_2) \cdot \mathbf{R}(\Theta)$$

$$= \begin{bmatrix} s_1 \cos^2 \Theta + s_2 \sin^2 \Theta & (s_1 - s_2) \cos \Theta \sin \Theta & 0 \\ (s_1 - s_2) \cos \Theta \sin \Theta & s_1 \sin^2 \Theta + s_2 \cos^2 \Theta & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

As an example of this scaling transformation, we turn a unit square into a unit square into a parallelogram stretching it along the diagonal from (0, 0) to (1, 1).s, *am* to be applied in We rotate the diagonal onto they axis and double its length with the transforms orthogonal directions defined by the angulation parameters $\theta = 45^\circ$, $s_1 = 1$, and $s_2 = 2$.displacement 6. we assumed that scaling was to be performed relative to the origin.

We could take this **scaling** operation one step further and concatenate thematrix with translation operators, so that the composite **matrix** would include parameters for the specification of a scaling fixed position.

Concatenation Properties

Matrix multiplication is associative. For any three matrices, A, B, and C, the **matrix** product A - B . C can be performed by first multiplying A and B or by first multiplying B and C:

$$\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C} = (\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C})$$

Therefore, we can evaluate matrix products using either a left-to-right or a righttleft associative grouping. On the other hand, transformation products may not be commutative: The matrix product A . B is not equal to B . A, in general. This means that if we want to translate and rotate an object, we must be careful about the order in which the composite matrix is evaluated For some special cases, such as a sequence of transformations all of the same kind, the multiplication of transformation matrices is commutative. As an example, two

successive rotations could be performed in either order and the final position would be the same. This commutative property holds also for two successive translations or two successive scalings. Another commutative pair of operations is rotation and uniform scaling ($S_x=S_y$)

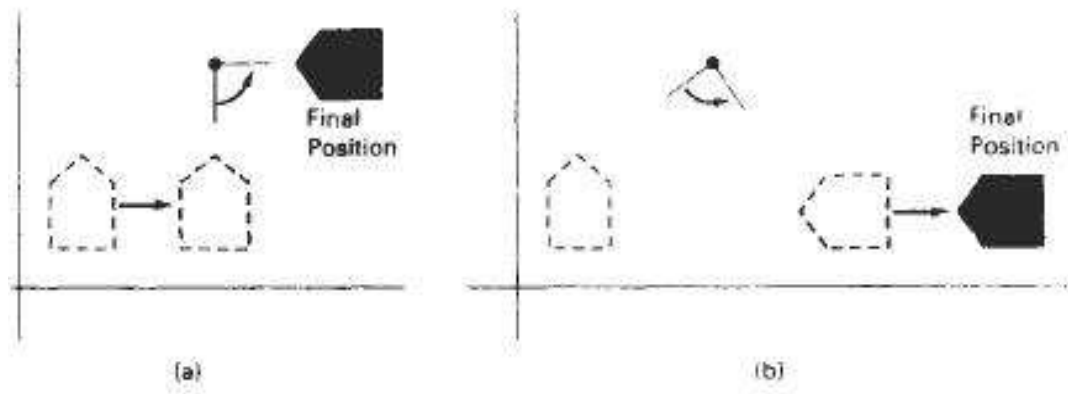


Fig. Reversing the order in which sequence of transformation is performed General Composite Transformations and Computational Efficiency

A general two-dimensional transformation, representing a combination of translations, Rotations, and scaling, can be expressed as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} rs_{xx} & rs_{xy} \\ rs_{yx} & rs_{yy} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} trs_x \\ trs_y \end{pmatrix}$$

The four elements $r_{s,}$, are the multiplicative rotation-scaling terms in the transformation that involve only rotation angles and scaling factors. Elements tr_s , and tr_y , are the translational terms containing combinations of translation distances, pivot-point and fixed-point coordinates, and rotation angles and scaling parameters.

For example, if an object is to be scaled and rotated about its centroid coordinates (x_c, y_c) and then translated, the values for the elements of the composite transformation matrix are

$$T(t_x, t_y) \cdot R(x_c, y_c, \Theta) \cdot S(x_c, y_c, s_x, s_y)$$

$$\begin{aligned} & \begin{matrix} s_x \cos \theta & -s_y \sin \theta & x_c(1 - s_x \cos \theta) + y_c s_y \sin \theta + t_x \\ s_x \sin \theta & s_y \cos \theta & y_c(1 - s_y \cos \theta) + x_c s_x \sin \theta + t_y \\ 0 & 0 & 1 \end{matrix} \end{aligned}$$

A general rigid-body transformation matrix, involving only transformations and rotations, can be expressed in the form

$$\begin{bmatrix} r_{xx} & r_{xy} & tr_x \\ r_{yx} & r_{yy} & tr_y \\ 0 & 0 & 1 \end{bmatrix}$$

where the four elements $r_{,}$, are the multiplicative rotation terms, and elements tr_x , and tr_y are the translational terms. A rigid-body change in coordinate position is also sometimes referred to as a rigid-motion transformation. All angles and distances between coordinate positions are unchanged by the transformation. In addition, matrix 5-40 has the property that its upper-left 2x2 sub matrix is an orthogonal matrix. This means that if we consider each row of the sub matrix as a vector, then the two vectors (r_{xx}, r_{xy}) and (r_{yx}, r_{yy}) form an orthogonal set of unit vectors: Each vector has unit length

$$r_{xx}^2 + r_{xy}^2 = r_{yx}^2 + r_{yy}^2 = 1$$

and the vectors are perpendicular (their dot product is 0):

$$r_{xx} r_{yx} + r_{xy} r_{yy} = 0$$

Therefore, if these unit vectors are transformed by the rotation submatrix, (r_{xx}, r_{xy})

is converted to a unit vector along the x axis and (r_{yx}, r_{yy}) is transformed into a unit vector.

Composite Transformations

Unit vector along the x axis of the coordinate system:

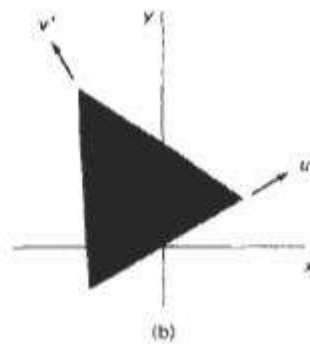
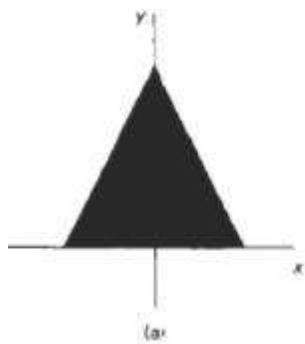
$T(t_x, t_y) \cdot R(x_r, y_r, \theta)$

$$\begin{aligned} & \cos \theta & X_r (1 - \cos \theta) + Y_r \sin \theta + t_x \\ & & Y_r (1 - \cos \theta) + X_r \sin \theta + t_y \\ = & \sin \theta & r \cos \theta & r \sin \theta & 0 & 0 \end{aligned}$$

$$\begin{aligned} & r_{xx} & r_{xy} & tr_x r_{xx} & 1 \\ & r_{yx} & r_{yy} & 0 & r_{xy} = 0 \end{aligned}$$

$$\begin{aligned} & 0 & 0 & 1 & 1 \\ & r_{xx} & r_{xy} & tr_x r_{yx} & 0 \\ & r_{yx} & r_{yy} & 0 & r_{yy} = 1 \end{aligned}$$

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



4. OTHER TRANSFORMATIONS

Shear

A transformation that distorts the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other is called a shear.

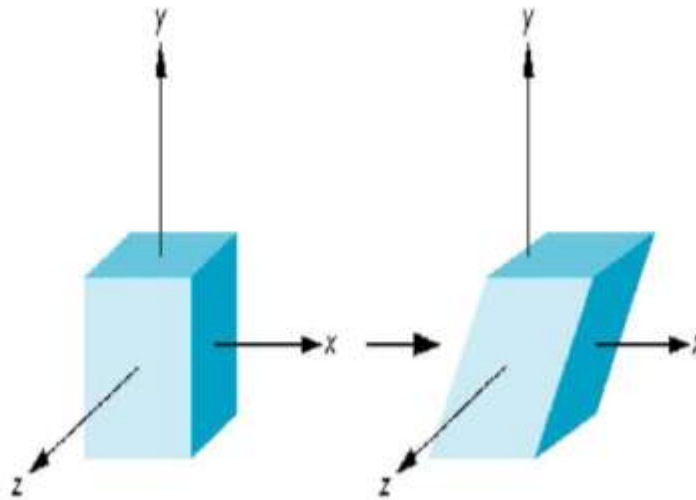


Fig 4.1 Unit square is turned into a shifted parallelogram

An x-direction shear relative to x-axis can be given as

$$x' = x + sh_x (y - y_{ref}), \quad y' = y$$

Similarly, y-direction shear relative to y-axis can be given as

$$x' = x \quad y' = sh_y (x - x_{ref}) + y$$

Matrix/Vector Representation of Shearing

In matrix representation, the x-direction shear equation can be given as

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

Similarly, the y-direction shear can be given as

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

5. 2D VIEWING

When we define an image in some world coordinate system, to display that image we must map the image to the physical output device. This is a two stage process. For 3 dimensional images we must first determine the 3D camera viewpoint, called the View Reference Point (VRP) and orientation. Then we project from 3D to 2D, since our display device is 2 dimensional. Next, we must map the 2D representation to the physical device. We will first discuss the concept of a Window on the world (WDC), and then a View port (in NDC), and finally the mapping WDC to NDC to PDC.

6. THE VIEWING PIPE LINE

A world-coordinate area selected for display is called a window. An area on a display device to which a window is mapped is called a viewport. The window defines what is to be viewed; the viewport defines where it is to be displayed. Often, windows and viewports are rectangles in standard position, with the rectangle edges parallel to the coordinate axes.

Other window or viewport geometries, such as general polygon shapes and circles, are used in some applications, but these shapes take longer to process. In general, the mapping of a part of a world-coordinate scene to device coordinates is referred to as a viewing transformation. Sometimes the two-dimensional viewing transformation is simply referred to as the window-to-viewport transformation or the windowing information. But, in general, viewing involves more than just the transformation from the window to the viewport.

Figure 6-1 illustrates the mapping of a picture section that falls within a rectangular window onto a designated angular viewport. In computer graphics terminology, the term window originally referred to an area of a picture that is selected for viewing, as defined at the beginning of this section. Unfortunately, the same term is now used in window-manager systems to refer to any rectangular screen area that can be moved about, resized, and made active or inactive.

7. VIEWING COORDINATE REFERENCE FRAME

This coordinate system provides the reference frame for specifying the world coordinate window. We set up the viewing coordinate system using the procedures. First, a viewing-coordinate origin is selected at some world position: $P_o = (x_o, y_o)$. Then we need to establish the orientation, or rotation, of this reference frame. One way to do this is to specify world vector V that defines the viewing y , direction. Vector V is called the view up vector. Given V , we can calculate the components of unit vectors $v = (v_x, v_y)$ and $u = (u_x, u_y)$ for the viewing y , and x , axes, respectively. These unit vectors are used to form the first and second rows of the rotation matrix R that aligns the viewing x, y , axes with the world x, y , axes.

Two-Dimensional Viewing world I view / x view

A viewing-coordinate frame is moved into coincidence with the world frame in two steps: (a) translate the viewing origin to the world origin, then (b) rotate to align the axes of the two systems. We obtain the matrix for converting world coordinate positions to viewing coordinates as a two-step composite transformation: First, we translate the viewing origin to the world origin, then we rotate to align the two coordinate reference frames. The composite two-dimensional transformation to convert world coordinates to viewing coordinate is

1. where T is the translation matrix that takes the viewing origin point P_o to the
2. world origin, and R is the rotation matrix that aligns the axes of the two reference
3. frames. Figure 6-4 illustrates the steps in this coordinate transformation.

8. WINDOWS TO VIEWPORT COORDINATE TRANSFORMATION Window

When we model an image in World Device Coordinates (WDC) we are not interested in the entire world but only a portion of it. Therefore we define the portion of interest which is a polygonal area specified in world coordinates, called the "window".

View port

The user may want to create images on different parts of the screen so we define a view port in Normalized Device Coordinates (NDC). Using NDC also allows for output device independence. Later we will map from NDC to Physical Device Coordinates (PDC).

The 2D viewing transformation performs the mapping from the window (WDC) to the view port (NDC) and to the physical output device (PDC). Usually all objects are clipped to the window before the viewing transformation is performed. We can see from above that to maintain relative position we must have the following relationship:

$$\frac{X_W - X_{WMIN}}{X_{WMAX} - X_{WMIN}} = \frac{X_V - X_{VMIN}}{X_{VMAX} - X_{VMIN}}$$

$$\frac{Y_W - Y_{WMIN}}{Y_{WMAX} - Y_{WMIN}} = \frac{Y_V - Y_{VMIN}}{Y_{VMAX} - Y_{VMIN}}$$

We can rewrite above as

$$X_V = X_{VMIN} + \frac{(X_{VMAX} - X_{VMIN})(X_W - X_{WMIN})}{X_{WMAX} - X_{WMIN}}$$

$$X_V = S_X(X_W - X_{WMIN}) + X_{VMIN}$$

$$(Y_{VMAX} - Y_{VMIN})(Y_W - Y_{WMIN})$$

$$Y_V = Y_{VMIN} + \frac{(Y_{VMAX} - Y_{VMIN})(Y_W - Y_{WMIN})}{Y_{WMAX} - Y_{WMIN}}$$

$$Y_V = S_Y(Y_W - Y_{WMIN}) + Y_{VMIN}$$

$$Y_V = S_Y(Y_W - Y_{WMIN}) + Y_{VMIN}$$

$$(X_{VMAX} - X_{VMIN})$$

Where $S_X = \frac{\text{-----}}{X_{WMAX} - X_{WMIN}}$

$$(Y_{VMAX} - Y_{VMIN})$$

Where $S_Y = \frac{\text{-----}}{Y_{WMAX} - Y_{WMIN}}$

Note that S_X , S_Y are "scaling" factors. If $S_X = S_Y$ the objects will retain same shape, else will be distorted, as shown in the example.

9. 2D VIEWING FUNCTIONS

The following function: evaluate View Orientation Matrix (xO, yO, xV, yV, error, viewMatrix). xV and yV are the world-coordinate positions for the view up vector. An integer error code is generated if the input parameters are in error; otherwise, the viewMatrix for the world-to-viewing transformation is calculated. Any number of viewing transformation matrices can be defined in an application. To set up the elements of a window-to-viewport mapping matrix, we invoke the function. Next, we can store combinations of viewing and window-viewport mappings for various workstations in a *viewing table* with setViewRepresentation (ws, viewIndex, viewMatrix, viewMappingMatrix, xclipmin, xclipmax, yclipmin, yclipmax, clippxy) where parameter ws designates the output device (workstation), and parameter viewIndex sets an integer identifier for this particular window-viewport pair.

The matrices viewMatrix and viewMappingMatrix can be concatenated and referenced by the viewIndex. Additional clipping limits can also be specified here, but they are usually set to coincide with the viewport boundaries. And parameter clippxy is assigned either the value *rroclrp* or the value *clip*. This allows us to turn off clipping if we want to view the part of the scene outside the viewport. We can also select *rloclip* to speed up processing when we know that the entire scene is included within the viewport limits. The function selects a particular set of options from the viewing table. This view-index selection is then applied to subsequently specified

output primitives and associated attributes and generates a display on each of the active workstations. At the final stage, we apply a workstation transformation by selecting a workstation window-viewport pair:

setViewIndex (5);

polyline (3 , axes);

10. CLIPPING OPERATIONS Introduction

Clipping refers to the removal of part of a scene. Internal clipping removes parts of a picture outside a given region; external clipping removes parts inside a region.

The different clippings are:

- Point Clipping
- Line Clipping
- Area Clipping (Polygons)
- Curve Clipping
- Text Clipping

Point Clipping

Assuming that the clip window is a rectangle in standard position, We save a point $P = (x, y)$ for display if the following inequalities are satisfied:

$$x_{w \min} \leq x \leq x_{w \max}$$

$$y_{w \min} \leq y \leq y_{w \max}$$

Where the edges of the clip window ($x_{w \min}$, $x_{w \max}$, $y_{w \min}$, $y_{w \max}$) can be either the world coordinate window boundaries or view port boundaries. If any one of these four inequalities is not satisfied, the point is clipped.

Line Clipping

Line clipping is the process of removing lines or portions of lines outside of an area of interest. Typically, any line or part thereof which is outside of the viewing area is removed.

This section treats clipping of lines against rectangles. Although there are specialized algorithms for rectangle and polygon clipping, it is important to note that other graphic primitives can be clipped by repeated application of the line clipper.

Simultaneous Equations

To clip a line, we need to consider only its endpoints, not its infinitely many interior points. If

both endpoints of a line lie inside the clip rectangle, the entire line lies inside the clip rectangle and can be trivially accepted. If one endpoint lies inside and one outside, the line intersects the clip rectangle and we must compute the intersection point. If both endpoints are outside the clip rectangle, the line may or may not intersect with the clip rectangle, and we need to perform

further calculations to determine whether there are any intersections.

The brute-force approach to clipping a line that cannot be trivially accepted is to intersect that line with each of the four clip-rectangle edges to see whether any intersection points lie on those edges; if so, the line cuts the clip rectangle and is partially inside. For each line and clip-rectangle edge, we therefore take the two mathematically infinite lines that contain them and intersect them. Next, we test whether this intersection point is "interior" -- that is, whether it lies within both the clip rectangle edge and the line; if so, there is an intersection with the clip rectangle.

Cohen-Sutherland Line Clipping

The Cohen-Sutherland algorithm clips a line to an upright rectangular window. The algorithm extends window boundaries to define 9 regions:

Top-left, top-center, top-right,
center-left, center, center-right,
bottom-left, bottom-center, and bottom-right.

See Fig 10.1 below. These 9 regions can be uniquely identified using a 4 bit code, often called an out *code*. We'll use the order: left, right, bottom, top (LRBT) for these four bits. $P = (x, y)$. In particular, for each point

- Left (first) bit is set to 1 when p lies to left of window
- Right (second) bit is set to 1 when p lies to right of window
- Bottom (third) bit is set to 1 when p lies below window
- Top (fourth) bit set is set to 1 when p lies above window

The LRBT (Left, Right, Bottom, Top) order is somewhat arbitrary, but once an order is chosen we must stick with it. Note that points on the clipping window edge are considered inside (the bits are left at 0).

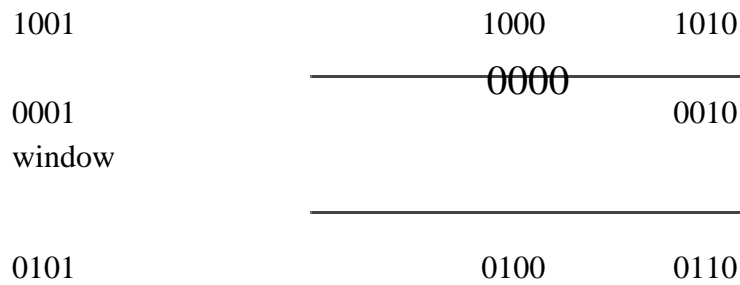


Fig. The nine region defined by an upright window and their out codes.

Given a line segment with end points $p_0 = (x_0, y_0)$ and $p_1 = (x_1, y_1)$, here's the basic flow of the Cohen-Sutherland algorithm:

1. Compute 4-bit outcodes LRBT0 and LRBT1 for each end-point
2. If both outcodes are 0000, the *trivially visible* case, pass end-points to draw routine. This occurs when the bitwise OR of outcodes yields 0000.
3. If both outcodes have 1's in the same bit position, the *trivially invisible* case, clip the entire line (pass nothing to the draw routine). This occurs when the bitwise AND of outcodes is not 0000.
4. Otherwise, the *indeterminate* case, - line may be partially visible or not visible. Analytically compute the intersection of the line with the appropriate window edges. Let's explore the indeterminate case more closely. First, one of two end-points must be outside the window, pretend it is.

1. Read P_1 's 4-bit code in order, say left-to-right.
2. When a set bit (1) is found, compute intersection point I of corresponding window edge with line from p_0 to p_1 .

As an example, pretend the right bit is set so we want to compute the intersection with the right clipping window edge, also, pretend we've already done the homogeneous divide, so the right edge is $x=1$, and we need to find y . The y value of the intersection is found by substituting $x=1$ into the line equation (from p_0 to p_1)

$$y_1 - y_0$$

$$y - y_0 = \frac{y_1 - y_0}{x_1 - x_0} (x - x_0)$$

and solving for y

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0)$$

Liang-Barsky Line Clipping

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.

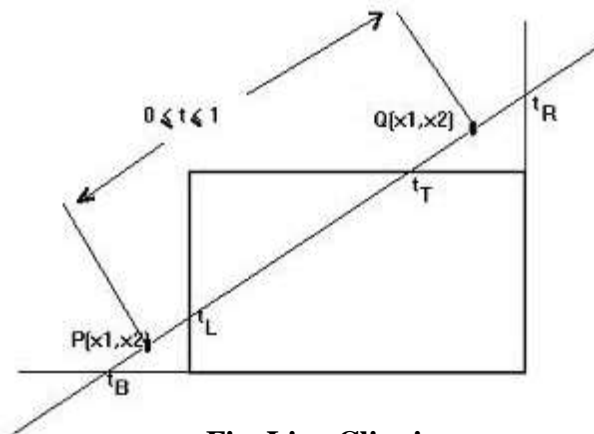


Fig. Line Clipping

Let $p(x_1, y_1)$ and $Q(x_2, y_2)$ be the line which we want to study. The **parametric equation of the line segment** from gives x-values and y-values for every point in terms of a **parameter** that ranges from 0 to 1. The equations are

$$x = x_1 + (x_2 - x_1) * t = x_1 + dx * t$$

$$y = y_1 + (y_2 - y_1) * t = y_1 + dy * t$$

We can see that when $t = 0$, the point computed is $P(x_1, y_1)$; and when $t = 1$, the point computed is $Q(x_2, y_2)$.

Algorithm

1. Set $t_{\min} = 0$ and $t_{\max} = 1$
2. Calculate the values of t_L , t_R , t_T , and t_B (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 $(x_1 + dx \cdot t_{\min}, y_1 + dy \cdot t_{\min})$ to $(x_1 + dx \cdot t_{\max}, y_1 + dy \cdot t_{\max})$
4. If the line crosses over the window, you will see $(x_1 + dx \cdot t_{\min}, y_1 + dy \cdot t_{\min})$ and $(x_1 + dx \cdot t_{\max}, y_1 + dy \cdot t_{\max})$ are intersection between line and edge.

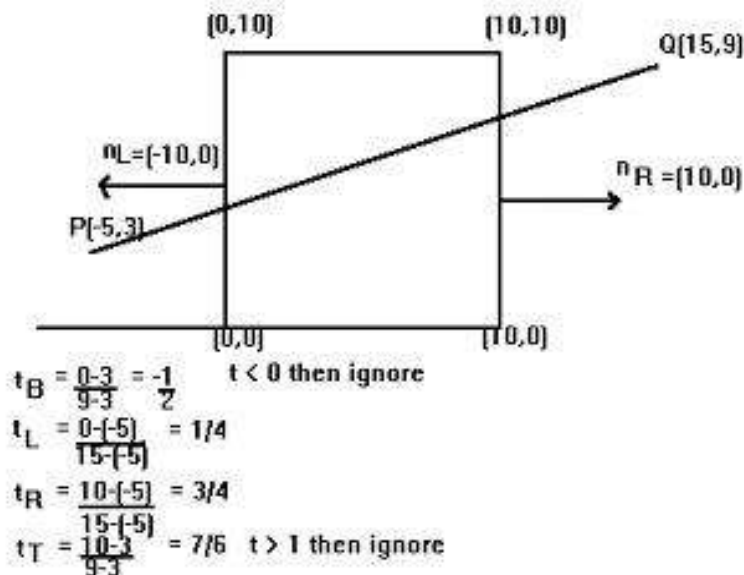


Fig. Line Passing Through Window

The next step we consider if tvalue is entering or exiting by using inner product. $(Q-P) = (15+5, 9-3) = (20, 6)$

- **At left edge** $(Q-P)n_L = (20, 6)(-10, 0) = -200 < 0$ **entering** so we set $t_{min} = 1/4$
- **At right edge** $(Q-P)n_R = (20, 6)(10, 0) = 200 > 0$ **exiting** so we set $t_{max} = 3/4$
- Because then we draw a line from $(-5+(20)*(1/4), 3+(6)*(1/4))$ to $(-5+(20)*(3/4), 3+(6)*(3/4))$

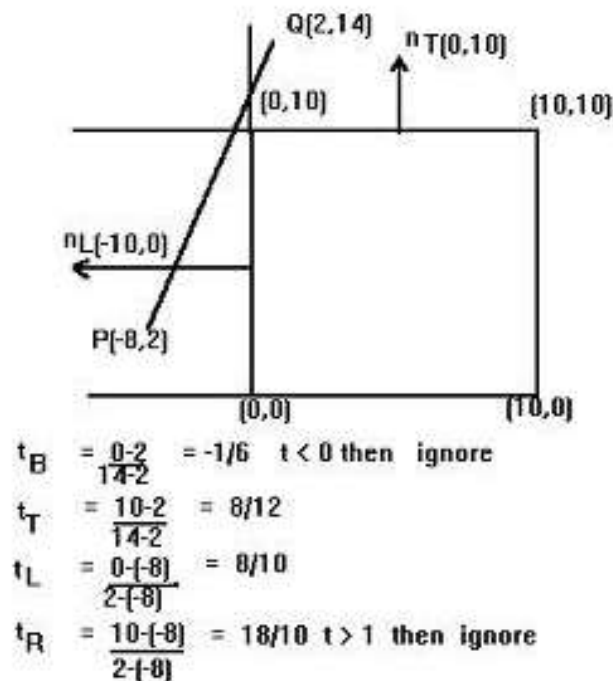


Fig. Line Not Passing Through Window

The next step we consider if tvalue is entering or exiting by using inner product. $(Q-P) = (2+8, 14-2) = (10, 12)$

- **At top edge** $(Q-P)n_T = (10, 12)(0, 10) = 120 > 0$ **exiting** so we set $t_{max} = 8/12$
- **At left edge** $(Q-P)n_L = (10, 12)(-10, 0) = -100 < 0$ **entering** so we set $t_{min} = 8/10$
- Because $t_{min} > t_{max}$ then we don't draw a line.

UNIT-III: Text: Types of Text – Unicode Standard – Font – Insertion of Text – Text compression – File formats. Image: Image Types – Seeing Color – Color Models – Basic Steps for Image Processing – Scanner – Digital Camera – Interface Standards – Specification of Digital Images – CMS – Device Independent Color Models – Image Processing software – File Formats – Image Output on Monitor and Printer.

1. TYPES OF TEXT

Text can be of various types

- “Plain text “ Consisting of fixed sized characters having essentially the same type of appearances formatted text where appearances can be changed using format parameters.
- “Hyper text which can serve to link different electronics document and enables the user to jump from one to the other in a non-linear way.

Unformatted text

- It is also known as plain text.
- The character set of ASCII Each character is represented by a seven bit binary code.
- The character include a to z, A To z and other punctuation characters like and single and double quoted etc

Formatted text

- A part from alphanumeric characters.
- Control characters are used to change the appearance of the character Eg; BOLN, UNDERLINE.ITALIC etc.
- Used in books journals magazines publishing sectors etc....

Hyper text

- Document provides a method of transmitting information.
- A hyper text document can be used to handle multiple reaches at different places for the readers.

- Like normal text a hyper text document can be used to reconstruct knowledge through sequential reading but additionally it can be used to link multiple documents in such a way that users can navigate non-sequentially from one document to the other for cross reference these links are called hyper links.
- Hyper text is mostly used on the WWW for linking different WebPages together and allowing the user to navigate from one page to another. On the web target documents are specified by the website address technically known as uniform resource locates.

2. UNICODE STANDARD

- The Unicode standard is a new universal character coding scheme for written characters and text.
- The Unicode standard goes far beyond ASCII's limited capability by providing the capacity of encoding more than 1 million characters.
- The Unicode standard draws a distinction between characters, which are the smallest component of written language, and glyphs, which represent the shapes, the characters can have when displayed.
- The Unicode standard deals only with character codes, representation of the glyphs are not part of the standard.
- For example, The Hindi character 'Pa' is represented by the Unicode sequence 0000 1001 0010 1010 (U+092A), how it will be rendered on the screen will be decided by the font vendor.

UCS-4, UTF-32

- Uses 32-bit for each character. The simplest scheme as it consists of fixed length encoding how it is not efficient with regard to storage space and memory usage, and therefore rarely used.
- The 4 here means 4 byte, and 32 means 32 bits.

UTF-16

- A 16 bits encoding format. In its native format it can encode numbers upto FFFF, i.e. as XXXXXXXX XXXXXXXX. For coding beyond this, the original number is expressed as a combination of two 16-bit numbers.

UTF-8

- The bits of a Unicode character is divided into a series of 8-bit number

Code range	Input code	Output code
000000-00007F	XXXXXXX	0XXXXXXX
000080-0007FF	XXX XXXXXXXX	110XXXXX 10XXXXXX
000800-00FFFF	XXXXXXX XXXXXXX	1110XXXX 10XXXXXX 10XXXXXX
010000-10FFFF	XXX XXXXXX XXXXXX XXXXXX	111110XXX 10XXXXXX 10XXXXXX 10XXXXXX

Fig 2.1 OUTPUT AND INPUT CODES**3. FONT****Font appearance**

The appearance of each characters in case of formatted text is determined by specifying a font main font name refers to font files which contains the actual description of the character appearance These files are usually in vectors format meaning that character descriptions are stored mathematically windows calls these fonts as true type fonts because their appearances stays the same on different devices like monitors printers and plotters and they have a file extension of TTF

Eg: Of default forms:

- Times Roman.
- Arial

Font size and style

Font may have a of sizes size is specified in a unit called point (PT) where one point equals 1/72 of an inch Standard characters are textual documents usually rangers from 10-12 pts in size some of the common styles one of are:

Bold, *Italics*, underline, superscript x^2 and subscript x_3

Some application packages allow change in horizontal gap between the characters call kerning and the vertical gap between 2 lines of the text called leading.

4. INSERTION OF TEXT

Using keyboard: The most common process of inserting text into digital documents is by typing the text using an input device like keyboard Microsoft word is used to contain the appearances of text which allows the user to manipulate variables like the font size colors style etc.

Copying and pasting: Another way of inserting text into a document is by copying text from a pre-existing digital document.

The text may be selected by using the copy command the selected text is copied into the clip board. By choosing the paste command where upon the text is copied from the clip board in to the target document

Using OCR software

A third way of inserting text in to a digital document is by scanning it from a paper document Text in a paper document including books newspapers magazines letter boards etc can be converted on to the electronic using a device called the scanner. Electronics representation of the paper document can then be saved ad a file on the hard disk of the computer. To be able to edit the text it needs to be converted from the image format into the editable text format using software called optical character recognition (OCR) software The OCR software traditionally works by a method called pattern matching.

5. TEXT COMPRESSION

Large text documents covering a number of pages may take a lot of disk space. we can apply compression algorithm to reduce the size of disk space.

a reverse algorithm must be applied to decompress the file before its contents can be displayed on screen.

Huffman coding

This type of coding is intended for applications in which the text to be compressed has known characteristic in terms of the characters used and their relative frequencies of occurrences. set of

variable length code words is derived such that the shortest code words is used to represent the most frequently occurring characters . this approach is called the

Huffman coding method.

lempel –zip(LZ) coding

In the second approach followed by lempel –zip method , instead of using a sigle charcter as basics of the coding operation, a string of character is used

lempel-ziv (LZ) coding

Most word processing packages have a dictionary associated with them which is used for both spell checking and compression of text. topically they contain in the region of 25000 words and hence, 15 bit ($2^{15} = 32768$) are require to encode the index. to encode the word compression. This method allows the dictionary to build up dynamically by the encoder and decoder for the document under processing. this dictionary becomes the match for a specific document than a standard dictionary.

6. FILE FORMATS

- TXT-Notepad (unformatted) DOS, UNIX, Windows.
- DOC- MSword.
- RTF-RICH TEXT FORMAT TEXT EDIT MAC OS
- POF-Adobe system (text, pic....)
- Post script Perform/interpreter

The different types of file formats used to store texts are as follows:

- **Text document [.TXT]**

This format is used to store unformatted text. notepad on windows platform is the default editor for text documents. This format can be used to transfer information between different platforms i.e. A TXT file can be accessed using MS.DOS.UNIX..

- **Document file [.DOC]**

This file format is developed by Microsoft for starting documents created by Microsoft word. This package had got a rich set of editing options. As of text documents this file type cannot be accessed in other platforms.

- **Rich Text Format [.RTF]**

This file format was developed by Microsoft in 1987 the WordPad editor with windows OS earlier created RTF files to store text documents. This format had very less editing options. This was developed for cross platform document exchange.

- **Portable Document Format (.PDF)**

This format was developed by adobe system for cross platform exchange. Apart from text this also supports images and graphics simple programs can be coded to read and write PDF in other applications.

- **Post script (.PS/.EPS):**

This format was mainly used in Desktop publishing. The file format has a page description language which is used into describes the contents of the page so that the output device display the accurate data. Past script offers universal language which is accepted by all printers the postscript compatible documents contrasts the input into PS format which is being sent to the printer. The postscript interpreter in the printer contrasts the I/P file into dot patterns.

7.IMAGES

Images are pictures that have been capture through electronic means from the real world.

IMAGE TYPES:

- **Continuous images**

Continuous images are ordinary autographs. They are usually composed of large number of shapes of color.

- **Half or partial tone**

Due to the limitations of the display or printing device all colors cannot be represented. In this case only a few colors are being display. Patches of color are applying to form the images. Pictures are broken into dots having the maximum use colors from the real images. Example: Images on news paper.

- **Bitone**

This type uses only two colors namely black and white. This image does not have any grey shades.

8. SEEING COLOR

The phenomenon of seeing color is dependent on a traid of factors :

- The nature of light
- The interaction of light and matter
- The physiology of human vision

Light is a form of energy known as electromagnetic radiation. Electromagnetic radiation consists of large number of waves with varying frequencies and wavelengths. The second part of color traid is human vision. The retina is the light sensitive part of the eye and its surface is composed of photoreceptors or nerve endings. These receive the light and pass it along through the optic nerve as a stimulus to the brain.

9. COLOR MODELS

Color Model is used to recognize and express information of colors. Most of the colors are from mixing few elementary colors. The mixing is done by specifying the color name and proportional of the elementary color.

Types of Color Models

RGB

This pattern is used in television and computer monitor. These colors mixed on different proportions reduce the composite colors. The proportions are measured in percentage. All three colors with full intensity (100%) produce white color. The three secondary colors of RGB are CMYK.

CMYK

CMYK is used to represent printed colors. Primary colors of this model are CMY(Cyan, Yellow, Magenta). When this combination is mixed in equal proportion it produce black. The original color produce is dark brown due to the impurities in the ink and hence an extra black color is added to get pure black. The 'k' represents the color black. C + M -> It produce blue.

M + Y -> It produce red.

Y + C -> It produce green.

The secondary colors of CMYK are same as RGB and are known as complimentary models.

10. BASIC STEPS FOR IMAGE PROCESSING

Image processing is the entire process involved with the input, editing and output of the images.

The three basic steps involved in the processing are input, editing and output.

Input

This is the first stage of image processing which involve getting the natural images into your system. This is done with the help of two devices namely the scanner and the digital camera. The scanner converts the printed image into a digital form. The digital camera digitizes the real world images into your digital form.

Editing

After the images have been digitized and stored they are changed or modified as per user's requirement. This process is called editing which involves one or more editing software. Editing software provides various tools and functionality.

Output

This process is concerned with the display of the edited image to the user. The output can be displayed in a screen or in application like presentation or web pages.

11. SCANNERS

Scanners are a physical device used to convert natural images into digital forms. The scanner is similar to a photocopy machine, with a glass panel and a moving head below it. The document to be scanned is placed facing down on the glass and the scanner is activated using software. This device is connected to a computer through an interface parallel port or USB (Universal Serial Bus).

Types of scanners

1. Flat Bed Scanner

This is the common type of scanner that is widely used for office purposes. The scanner head moves slowly from one end to another end of the document. When the head moves a source of white light falls on the paper. This light gets reflected and falls CCD array (Charge Coupled Device). Depending on the darkness of the regions of the document CCD generates varying voltage signals that are stored in the buffer.

2. Drum Scanner

Drum scanners are used for professional purposes, as it provides high quality and better performance than flat bed scanners. It has a cylindrical drum made of plastic like material. The image to be scanned is wet mounted on the drum. The fluid can be oil or alcoholic based. This fluid clears the scratches and dust grains over the film and gives improved clarity for sensing. Drum scanners use Photo Multiplier Tube (PMT). The multiplier consists of sealed glass tube having an anode and number of dynodes. This works with the principle of photo electronic effect in which absorption of a photo results in an electronic emission.

3. Barcode Scanner

Used to read barcodes printed on various surfaces. A barcode is a machine readable representation of information in a visual format. They are generated using a set of parallel vertical lines whose widths and spacing between them is used to encode data's. The encoded information is directly read by the scanner. The light source can be either a LCD or a LASER. The light falling on the code is reflected and sent to a photoconductor for translating them into electrical impulses. The decoder circuit of the scanner analyzes the data and sends it to the output part of the scanner.

12. DIGITAL CAMERA

The digital camera has a lens similar to the ordinary camera. Instead of the light falling on the film here it falls on a CCD array. The voltage pulses from the CCD array travel to an ADC where they are converted to binary format and stored as digital image files. It has got its own storage space in it. They are compressed to reduce the file size and stored in JPEG format. These files can be later transferred to a PC. Most of the cameras have an LCD screen which can be used for making adjustments before storing it and to view the stored files.

13. INTERFACE STANDARDS

- Interface standards determine how data from acquisition devices like scanners and digital cameras flow to the computer in an efficient way.
- Two main interface standards. exit: TWAIN and ISIS. Interface screen of the software utility in a digital camera

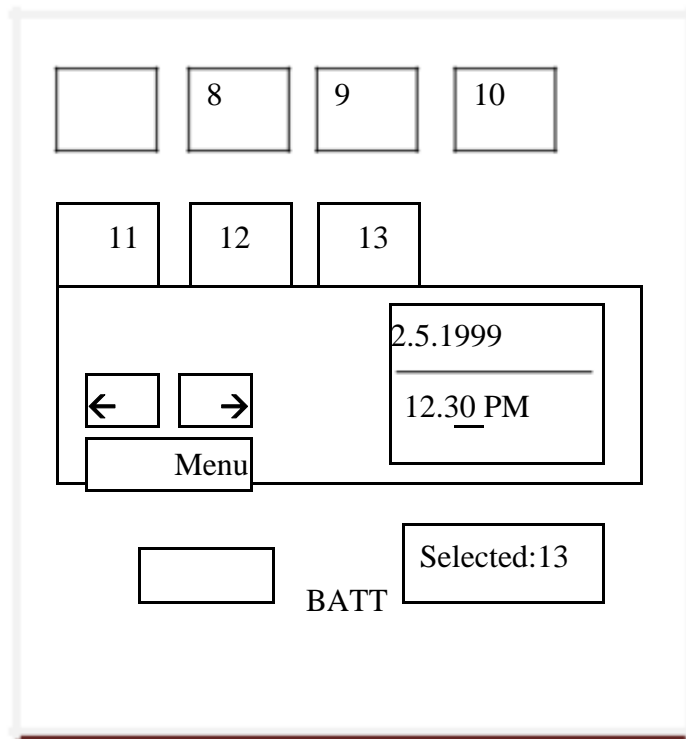


Fig. Interface screen of the software utility in a digital camera

TWAIN

- TWAIN is very important standard in image acquisition, developed by Hewlett-Packard, Kodak, Aldus, Logitech and Caere which specifies how image acquisition devices such as scanners, digital cameras and other devices transfer data to software applications.
- TWAIN is a protocol which regulates the flow of information between software applications and imaging devices like cameras.
- TWAIN working group which is a non – profit organization with representative from leading imaging vendors.

- TWAIN uses a four layer protocol for connecting TWAIN –complaint devices with TWAIN – complaint applications.

Image and scanner interface specification (ISIS)

- The second important standard for document scanner is the image and scanner interface specification (ISIS).
- ISIS was developed by Pixel Translations and they retain control over its development and licensing.
- The ISIS architecture is based on software modules like image acquisition , file conversation, data extraction and file R/W commands.

14. SPECIFICATION OF DIGITAL IMAGES

The scanner or digital scanners are used to digitize an image. After an image has been digitized we can identify characteristics of image which deter mine the quality and storage space.

Pixel dimensions

The number of pixel along the height and width of a bitmap image is known as *pixel dimension*

To find out a display size of an image on a screen we need

- Pixel dimension of the image s
- Size of the monitor
- Setting of monitor

The file size of an image is proportional to its pixel dimensions.

Image resolution

The number of pixels displayed per unit length of the image is known as the image resolution measured by pixel per inch (PPI). High resolution means the pixel per inch in an image is more. Low resolution means the pixel per inch an image is low.

For example Low resolution

A 1 inch by 1 inch with resolution of 72 ppi contains total 72X72 pixels (5184) Example for high resolution

A 1 inch by 1 inch with resolution of 300 ppi contains total 300X300 pixels (90,000) **Low resolution**

- Produce low detail and color transitions
- No need more storage space

High resolution

- Produce more detail and finer color transitions
- Improve the quality
- Need more storage space

File Size

Digital Size of an image measured in

- Kilo bytes
- Mega bytes
- Giga bytes

Image with more pixels produce more detail but they require more storage space. So it is difficult to edit and print. To determine the file size

(Pixel width*pixel height) * (Bit depth / 8)

Example width w pixel, height h pixel bit depth b bits

File size (whb)/(8*1024) in KB

Color Depth

This defines the number of bits required to store the information of each pixel in the image. It determines the total number of possible colors that can be displayed in the image. For example 8 bit requires total 2^8 or 256 colors. Increasing bit depth means

- Increase capability of an image
- Increases file size that is storage space of an image

Bit map VS Vector representation

There are two types of storing an image internally in hard disk of a computer. They are

- Bit map
- Vector Types

Bit map

- Bit map is stored internally as collection of dots called pixel.
- Each pixel representing a particular color.
- A bit map image is suitable for representing continuous tone photographic pictures.
- Collection of pixel arranged in a special shape
- Images are stored as bitmap entities
- Bitmap images generated by scanning analog images using scanner.
- A bitmap image/graphic may be scaled larger size leads to increase in distance between pixel and the image appearance with degradation in quality.
- Need to store pixel information, so it should be a larger file size.
- Difficult to edit and manipulate
- Difficult to animate because of larger size
- Photographic continuous tone images
- It is used for displaying photographic quality images with complicated forms or large variations of colors.

Vector Image

- Vector type image is stored internally as mathematical entities
- For example to draw wheel and circle with a specific center location and radius
- Graphics are stored as vector entities
- Vector graphics are created using software like Macromedia flash.

- A vector image /Graphic may be scaled to various sizes without degradation on quality
- Vector image/Graphic is quite compact and has a small file size. No pixel information.
- Easy to edit & manipulate
- Easy to animated (translation and rotation) because of small size
- Not able to represent accurately a large number of fine color variations
- It is used for simple pictures with limited number of colors that needs to scaled or animated.

Pixilation

Individual pixel become visible as single colored squared and the image takes on boxy appearance with degradation in quality. This phenomenon is called pixilation.

15. COLOR MANAGEMENT SYSTEM (CMS)

A color management system (CMS) is a collection of software tools designed to reconcile the different color capabilities of scanners, monitors, printers and image setters.

To achieve reliable and reproducible color throughout the entire reproduction process for users the computer and color publishing industry form a new approach called International Color Consortium (ICC).

ICC CMS has 3 major components

- Device profile
- Color management module
- Device Independent color space

ICC Profile

It defines the color characteristics of a particular device

The set of colors actually displayed by a specific device example monitor or printer is known as color space of that device.

A CMS works by determining the color space of each device stored as a set of parameters in a file called *device profile*

Example the monitor has the following parameters

- Brightness
- Contrast
- Mid tone settings
- Characteristics of the phosphor

Specialized software is used to obtain the parameters and compile them into a device profile file. For example using image processing software like Adobe Photoshop, the device profile is attached with the image file when the image is created.

Color Management Module

It interprets the devices profile and carries out the instructions on what to do with different devices color spaces.

When an image created on one monitor is displayed on another monitor, the difference in the characteristics between the two monitors would change the observed colors of the image.

CMM would read the two device profiles, analyze the difference between them and adjust the image colors. So that it looks same when displayed in second monitor. It attempts to minimize the difference and it never gives 100% guarantees.

Device Independent Color Space

It is also called Reference color space. It is an interpreter for translating from one color space to another that is the CMM needs to map the first color space to a device independent color space and then maps against to the second color space.

For example assume four people in a room their languages are unique.

For the group to communicate they need an interpreter who known all languages likewise the device independent color space also a interpreter to communicate among the system.

16. DEVICE INDEPENDENT COLOR MODELS

Primary colors RGB intercepts different colors when mixed with the choice of the user.

- CIEL Model
- HSB Model

CIEL Model

Commission International Eclairage illumination is a model where

- +ve a ->Red
- -ve a ->Green
- +ve b ->Yellow
- -ve b ->Blue

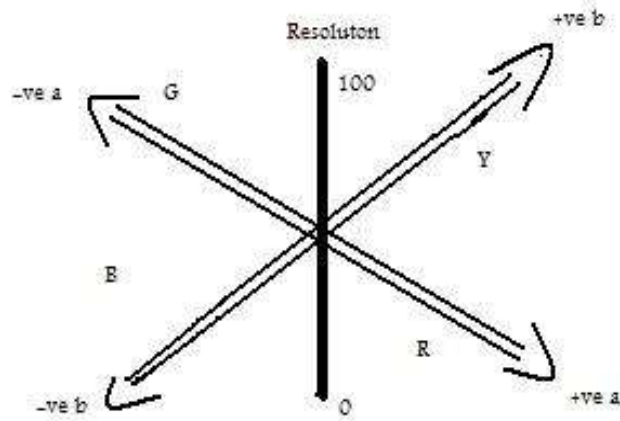


Fig. The CIE L*a*b* model

HSB Model (Hue Saturation Brightness)

Hue defines the color itself

- Saturation defines the degree which hue differs from noted color.
- Brightness indicates the illumination or the intensity that shows the level of brightness/lightness.

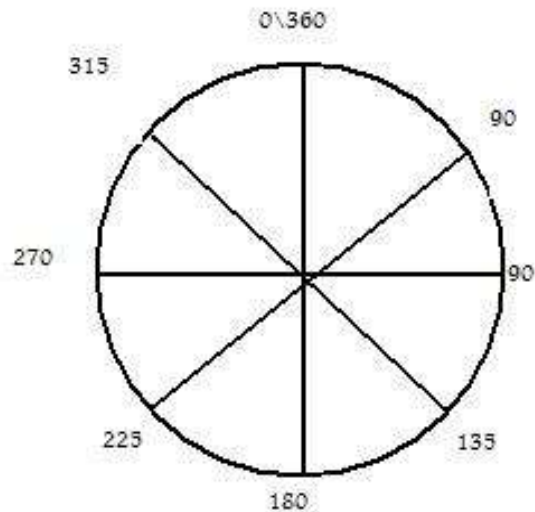


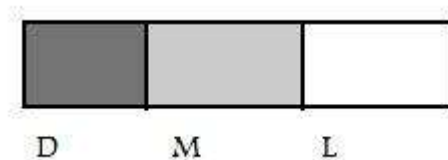
Fig. Color Wheel

Gamma

The brightness level in the monitor is divided into three

- Darkest region
- Midtone region
- Lightest region

Gamma refers to Midtone region



Gamma Correction

To prevent all the pictures on the screen from looking excessively dark. A correction is applied to a monitor which is known as Gamma correction.

$1/\text{gamma-corrected}$

$\text{o/p} \rightarrow \{ (1/p \text{ pixel}) \}$

17. IMAGE PROCESSING SOFTWARE

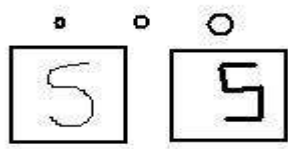
- Selection tool
- Paint and Drawing tool
- Color selection tool
- Gradient tool
- Clone tool
- Transformation tool
- Retouching tool
- Text tool

Selection tool

It enables the user to select a portion of an image and copy to another image. The selection border may be geometrical in shape like rectangular, circular, polygonal, or may be irregular in shape. Selection may also be done based on color instead of shape.

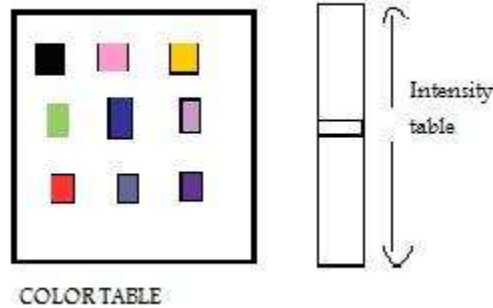
Paint and Drawing tool

They are used to paint lines, shapes or anything the user wishes to draw.



Color selection tool

These tools are used to select the foreground and background color from the color palette with the intensity selected by the user.

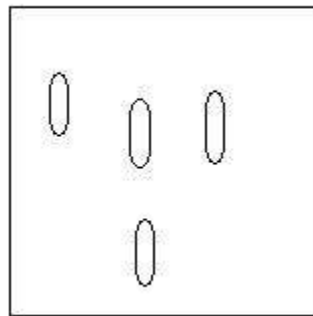


Gradient tool

These tools are used to create smooth blend of multiple colors or an individual color with light, lightest and brightest variation.

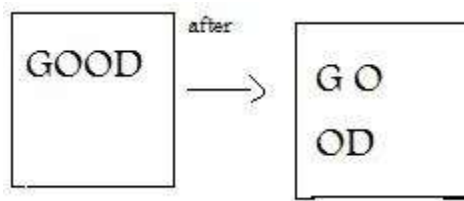
Clone tool

Clone tools are used to create multiple copies of any selected image.



Transformation tool

These tools are used to transform specific portions of an image in various ways like moving, rotating, scaling, skewing, distorting etc.

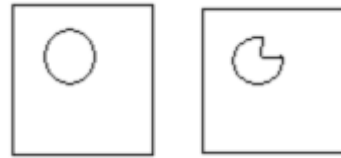


It is used to transform a specific portion of an image in various ways like

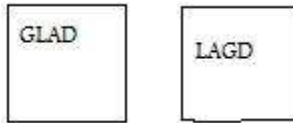
- Moving
- Rotating
- Scaling
- Skewing
- Distorting



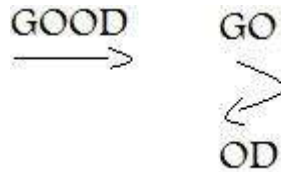
Moving



Skewing



Distorting



Rotating

Retouching tool

It is used to change the brightness or the contrast of an image or text.

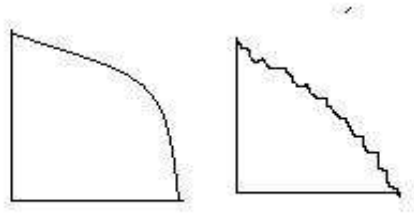
Text tool

These tools allow the user to include text in various styles and sizes.

Eg: Word art

Anti-aliasing

A staircase effect replaces the original smooth curve or diagonal lines. This effect is called aliasing.



Dithering

Dithering refers to graphic technique which is used to improve the appearance of an image.



18. FILE FORMATS

Images can be displayed in a variety of file formats.

- BMP
- JPEG
- GIF
- TIFF
- PNG
- TGA
- PICT
- PSD

BMP(Bitmap image)

These are used for 4 bit and 8 bit images. It supports RGB color, Index color and so on. Eg: Paint files using .bmp format

JPEG(Joint Picture Expert Group)

These are used for storing photographs so that it gives high quality outcome.

GIF(Graphic Interchange Format)

It is the format commonly used to display, animate in HTML document over WWW and other online services.

TIFF(Tagged Image File Format)

TIFF is a flexible image format supported by paint image editing and other layout applications.

PNG(Portable Network Graphics)

PNG format are used to display an image on WWW and online network services.

TGA(Targa format)

This format is designed for systems, which is normally supported by MS Dos color application.

PICT(Picture)

Picture format is widely used in Macintosh tutorial or OS graphics which is an inter mediatory file format for transferring images between applications.

PSD(Photoshop document)

This is the default file format used in Photoshop package.

19. IMAGE OUTPUT ON MONITOR AND PRINTER

- Image have specific resolution which determine the gap between pixels.
- The image pixels are actually strings referred to as logical pixels.
- When the images are displayed on the monitor the logical pixel mapped on the phosphor dots of the monitor this is
- On screen appearance of an image will depend

*Monitor resolution

*Monitor size

Dependence on Monitor Resolution

- Let us consider an image having dimensions 1 inch by 1 inch and resolution of 72 dpi.
- The image is made up of 72 logical pixel vertically. When this image is displayed on a 15" monitor having resolution 72dpi.

- On screen the image size is exactly 1 inch by 1 inch because the monitor resolution is equal to image resolution.
- Let the image be resampled at a higher resolution of 144ppi and then displayed on the same 15" monitor.
- The internal size of the image is still 1 inch by 1 inch but now there are 144 logical pixels per inch of the image.
- The monitor resolution is changed and on screen the image takes up 2 inches along the monitor length.
- Similarly the height of the image also increases to 2 inches along the monitor height. Internal representation of an image is still 1 inch by 1 inch, on screen size increases to 2 inches by 2 inches.
- In this case the monitor resolution is less than the image resolution and this makes the image look bigger. In the case of decreasing image resolution the monitor resolution and this makes the image look smaller.

Dependence on monitor size

- Let us consider the monitor which displays 640 pixels horizontally and 480 pixels vertically.
- An image with pixel dimensions of 640x480 would fill up the entire screen.
- The same image is displayed on a 20" monitor with a 640 by 480 setting, the image would again fill up the entire screen.
- Because the physical pixels of the 20" monitor are themselves larger than the phosphor dots of the 15" monitor.
- If the resolution of 20"

- Monitor is increased by 800*600 then the image will occupy only that portion.

PRINTER

Image Output On Printer

Two types are mostly used for printing multimedia content :the LASER printer and the inkjet printer .The no.of dots printed per inch of the printed page is called the **printer resolution** and expressed as dots per inch (dpi). Modern printers are capable of supporting resolutions of over 300 dpi.This is much larger than the typical 72 dpi to 96 dpi resolution of a monitor .

Laser Printer

The LASER printer was introduced by Hewlett – Packard in 1984 , based on technology developed by Canon.It worked in a similar way to a photocopier ,the difference being the light source.With a photocopier a page is scanned with a bright light ,while with a laser printer the light source is ,not surprisingly , a LASER (Light Amplification by Stimulated Emission of Radiation) beam. LASER printers quickly became popular due to the high quality of their print and low running costs.

LASER beam is activated over the drum which rotates in the clock wise direction .Wherever the LASER strikes the drum surface , the points are charged positively. The image is therefore translated to a collection of charged points over the drum surface .The developer roll applies the ink , a black powdered material toner ,which is attracted by the positively charged points.

Inkjet printer

Like LASER printers , employ a non- impact method meaning that there is no head or hammer striking the paper to produce the print , like typewriters or the dot – matrix printer . Ink is emitted from nozzles as they pass over a variety of possible media. A print head scans the page in horizontal strips , using a motor assembly to move it from left to right back , as another motor assembly rolls the paper in vertical steps .A strip of the image is printed , then the paper moves on ,ready for the next strip.

Thermal technology

Most inkjets use thermal technology, whereby heat is used to fire ink on to the paper . There are three main stages with this method.

Piezo- electric technology

Epson's proprietary Piezo- *electric technology* uses a piezo crystal at the back of the ink reservoir. It uses the property of certain crystals that causes them to oscillate when subjected to electrical pressure (voltage). So , whenever a dot is required , a current is applied to the piezo element , the element flexes and in so doing forces a drop of ink out of the nozzle.

Inks

The ink used in inkjet technology is water based and this poses a few problems. The results from some of the earlier inkjet printers were prone to smudging and running, but over the past few years there have been enormous improvements in ink chemistry. oil – based ink is not really a solution to the problem because it would impose a far higher maintenance cost on the hardware.

UNIT-IV: Audio: Introduction – Acoustics – Nature of Sound Waves – Fundamental Characteristics of Sound – Microphone – Amplifier – Loudspeaker – Audio Mixer – Digital Audio – Synthesizers – MIDI – Basics of Staff Notation – Sound Card – Audio Transmission – Audio File formats and CODECS – Audio Recording Systems – Audio and Multimedia – Voice Recognition and Response - Audio Processing Software.

1. AUDIO

- Sound is a form of energy capable of flowing from one place to another through a material medium,
- It is generated from vibrating objects when part of the energy of vibration is converted to sound energy,
- Sound is usually represented as waves
 - ❖ Periodic waves corresponding to musical sound.
 - ❖ Aperiodic waves corresponding to noise.
- The amplitude of sound waves corresponds to its loudness, frequency corresponds to its pitch and the wave form to the quantity or timber,
- Sound of frequencies between 20Hz to 20,000Hz,
- To process sound the following system components are required.
 - ❖ Microphones for sound input,
 - ❖ Amplifiers for boosting the loudness levels,
 - ❖ Loud speakers for output or play back of sound.
- Music generated from synthesizers are pure digital sounds which do not need to be digitized.
- to digitizing and play back of sound on a computer sound card is required.

2. ACOUSTICS

- Sound is a form of energy similes to heat and light. Sound is generated from vibrating objects and can flow through a material medium from one place to another.

- During generation the kinetic energy of the vibrating body is converted to sound energy.
- Acoustic is the branch of science dealing with study of sound and is concerned with the generation, transmission and reception of sound waves.
- The application of acoustics in technology is called acoustical engineering.
- The main sub disciplines of acoustics are,

a) Aero acoustics.

Concerned with low gas flows produce sound and has applications in aeronautics, Eg: Sound & shock waves by jets.

b) Architectural-acoustics.

Concerned with the study of sound in buildings like halls and auditoriums.

c) Bio-acoustics.

Concerned with the sound made by the animals like elephants, whales, bats etc...

d) Biomedical-acoustics.

Concerned with the study of sound in medicine.

Eg: ultra-sonography.

e) Psycho-acoustics.

Concerned with the hearing, perception and localization of sound related to human beings.

f) Physical-acoustics.

Concerned with interaction of sound with materials and fluids. E.g.:

how shock waves travel through the solids and liquid portion of earth crust.

g) Speech communication.

Concerned with the production, analysis, transmission and recognition of speech.

h) Ultrasonic.

Concerned with the study of high frequency sounds beyond human hearing range.

i) Musical-acoustics.

Concerned with the study of sounds in relation to musical instruments to name a few.

3. NATURE OF SOUND WAVES

Sound is an alternation in pressure, particle displacement or particle velocity propagated in an elastic material. As sound energy propagates through the material medium, it sets up, regions of compression and rarefaction by shifting the particles of the medium

- Two characteristics of sound waves.

1) Longitudinal waves.

Which means the direction of propagation of sound is the same as the direction along which is the medium particles oscillates.

2) Mechanical waves.

Which means that they are capable of being compressed and expand like springs. On compression, the frequency of sound increases and its appears more high pitched. While on expansion, the frequency decreases making it appear more dull and flat. This is known as Doppler Effect.

4. FUNDAMENTALS CHARECTERSTICS OF SOUND The physical characteristics of sound waves are

- Amplitude
- Frequency
- Wave form
- Speed of propagation.

Amplitude

- Amplitude of a wave is the maximum displacement of a particle in the path of a wave and is a measure of the peak-to-peak height of the wave.
- Sound waves correspond to the loudness of sound. Loudness is measured in a unit called decibel.(db)
- The sound wave with higher amplitude will be louder to hear than the wave the lower amplitude

Frequency

- This measure the number of vibrations of a particle in the path of a wave, in one second.
- Frequency is measured in a unit called Hertz (Hz).
- A sound of 1Hz is produced by an object vibrating at the rate of 1 vibration per second.
- The audible frequencies are referred to as ultra sonic or super sonic.
- Frequencies higher than sonic are referred to as ultra sonic while frequencies lower than sonic are called infra-sonic (or) sub-sonic.

Wave form

- This indicates the actual shape of the wave when represented pictorially.
- Shapes of the waves can be,
 - ❖ Sinusoidal
 - ❖ Square
 - ❖ Triangular
 - ❖ Saw tooth
- Complex sounds can be of any arbitrary and irregular shape.
- Two sounds having the same loudness & pitch but having different wave forms will have different hearing perceptions in our ears.

Speed

- The speed of the sound depends on the mediums through which the sound travels & the temperature of the medium but not on the pressure.
- The speed is about 340m/sec in air and 1500m/sec in water.

Elements of audio systems

- An elementary sound system is where we can record, manipulate and playback sound. It consists of a number of major component. These microphone is responsible for input of sound to the audio system.
- The microphone converts the environmental sound into electrical form, i.e. conversion of sound energy into electrical energy. Once convert the electrical signals can then be recorded onto magnetic material like audio tape in an audio recording system. However before

that is done, we need another component called the amplifier to boost the level (i.e. amplitude) of the electrical signal.

- Once recorded, the sound can be played back from the recording media by converting it back to environmental sound. This is done in speaker (or loud speaker). The speaker function just opposite to that of the microphone, i.e. it converts electrical energy back into sound energy.

5. MICROPHONE

- A microphone records sound by converting the acoustic energy to electrical energy. Sound pressure exists as patterns of air pressure. The microphone changes this information into patterns of electrical current.
- Microphone may be of two types: moving coil type and condenser type.

Dynamic microphone

A moving coil (or dynamic) microphone consists of a thin metallic diaphragm and an attached coil of wire. A magnet produces a magnetic field which surrounds the coil. As sound impinges on the diaphragm attached to the coil, it causes movement of the coil within the magnetic field. A current is therefore produced proportional to the intensity of the sound hitting the diaphragm.

Condenser microphone

In the condenser microphone, the diaphragm is actually the plate of a capacitor. The incident sound on the diaphragm moves the plate thereby changing the capacitance and generating a voltage. This causes a current to flow in an attached wire which in this case too is proportional to the intensity of the sound on the diaphragm. Based on the directional properties, microphones may be classified into three types;

- omni-directional,
- bi-directional,
- uni-directional.

Omni-directional microphone

An **omni-directional** microphone is equally sensitive to sounds coming from all directional. These are used to record sound coming from multiple sources, e.g. environmental sounds in a wild-life video clip. It consists of a container open at one end, containing a diaphragm causing it

to vibrate. This vibration is translated to electrical signals through either of the above mechanism.

Bi-directional microphone

A **Bi-directional** microphone is sensitive to sound coming from two directions: the front and rear. It is used to record two sources of sound simultaneously, e.g. conversation between two persons on opposite sides of a table. It consists of a container with a diaphragm inside and two openings on opposite sides. Sound produced near the front of microphone will enter through the first opening and cause the diaphragm to vibrate. A short time later the sound will also travel to the back of the microphone and enter it through the rear opening striking the diaphragm from the opposite side.

Uni-directional microphone

A **uni-directional** microphone is designed to record sound from a single source, e.g. a single individual speaking. Its construction is similar to that of bi-directional one, with a Single exception. On the rear side of the microphone is a resistive material like foam or cloth near the diaphragm. This tends to absorb some of the energy of sound entering through the rear opening. Sound produced at the front of the microphone strikes the diaphragm directly from the front while a part of the energy travels to the back; gets reduce by the resistive material and strikes the diaphragm with a smaller force from the opposite direction.

The diaphragm vibrates with the differential force and the microphone responds to the sound. When sound is produced at the back of the microphone, the direct energy wave gets reduced by the resistive material before striking the diaphragm from the back. A part of the original sound energy traveling a longer distance to the front also gets reduced before striking the diaphragm from the front in the opposite direction.

Polar plot

The polar plot of a microphone is a graph plotting the output level of the microphone against the angle at which the incident sound is produced. By definition the omni-directional microphone produces equal outputs for all angles of incidence. Hence, its polar plot is a circle, as shown in fig. for bi-directional microphone, the output are maximum for sounds coming from the front (0°) and rear (180°).

The output gradually decreases as the incident sound shifts from front (a rear) to the sides (90°)

and 270°). The polar plot therefore resembles the fig. for a unit-directional microphone, the output is maximum at the front and minimum at the rear, resulting in a decreased but non-zero value at the sides. The polar plot is heart shaped as shown in fig. due to which such microphones are also known as 'cardioids' microphones.

6. AMPLIFIER

In general, amplifier is the name given to a device in which a varying input signal controls a flow of energy to produce an output signal that varies in the but has a larger amplitude. Amplifier used in audio processing is electronic in nature and use a series of transistors as their principal components, connected in a printed circuit board. The amplifier takes power from a power supply and use the energy to produce an output electrical signal which has the same shape as the input signal but larger in amplitude. The ratio of the output to the input amplitude is known as gain of the amplifier. Amplifier circuits are designated as A, B, AB and C for analogue design and D and E for digital designs.

Class-A

Class- A amplifiers use 100% of the input cycle for generating the output. They are not very efficient, a theoretical maximum of 50% efficiency is obtained and are usually used for small signal levels. The amplifying transistor is biased such that the device is always conducting; due to which more power is wasted.

Class- A amplifiers are not generally used as audio amplifiers nowadays due to the generation of large amount of heat which necessitates large heat sinks for their dissipation.

Class-B and Class-AB

Class-B amplifiers only use half of the input cycle for amplification. Through they produce a large amount of distortion, these amplifiers are more efficient than class-A because the amplifying element is switched off during half of the cycle.

Class-B amplifiers are used in RF power amplifiers where distortion is unimportant. These amplifiers are rarely used singly in a circuit, rather they are mostly used in pairs in a push-pull configuration, and i.e. each element is used to amplify the opposite half of the input wave after which they are joined, known as crossover distortion.

The solution is to keep the elements on even after the corresponding half-cycles are over. The elements are biased in such a way that they operate over a linear portion of their characteristic curve during a half cycle, but still conduct a small amount of current in the other half. This arrangement is called Class-AB operation and is mostly used for audio amplifiers.

Class-C

Class-C amplifiers use less than half of the input cycle for amplification. Though they produce a large amount of distortion, they are the most efficient. These are used mostly a power amplifiers for filtering or suppressing certain unwanted harmonics from the input wave. They can also be used in situations where other auxiliary equipment can be used to reduce the distortion produced.

Negative Feedback

The amplifying elements are essentially non- linear, distortion in the output wave is an inherent problem that needs to be dealt with. The distortion is perceived as noise at the output. Through good design can reduce the distortion, there is a limit to how much it can be reduced. One way of reducing distortion further is to introduce a negative feedback.

This involves feeding back a portion of the output back to the input so that it is subtracted from the original input. The negative distortion combine with the positive distortion produce subsequently by the amplifier with the result that output signal is more or less linear. This is a popular mechanism of reducing noise in the output signal.

7. LOUDSPEAKER

- Loudspeaker is a device that converts electrical energy back to acoustics energy, function is just opposite to microphone.
- It generate environmental sound whose loudness is proportional to the amplitude of the sound electrical signal.

Dynamic loudspeakers

- It is based on traditional design made up of wire coil and paper cone.
- Cone made up of paper or fiber known as diaphragm, attached to coil of wire kept near to magnet.
- When current is passed through the coil a magnetic field is generated around the coil.

- This field generate vibrating force which oscillates diaphragm, the diaphragm oscillates frequency and therefore reproduce sounds.
- All these components enclosed in container to provide stability in vibration.

Woofers and tweeters

- A loudspeaker is divided into small units ,each unit tuned for small frequency range .these units called woofers handle low frequency,**midrange** handle mid frequency, and **tweeters** handle high frequency.
- Woofers handle frequency range from 20Hz to 400 Hz.such low frequency sounds are known as **bass**.
- Mid range speakers are to handle frequency range between 400Hz to 4KHz.
- Tweeters handle frequency range between 4KHz to 20 kHz.
- Such high frequency range is known as Treble.
- Modern speakers system often includes a sub-Woofer for handling very low frequency.

8. AUDIO MIXER

- Audio mixer used to record individual tracks and edit them separately. Each track have number of controls for adjusting volume, tempo, mute, etc.
- Using these controls each separate track of sound eg guitar piano track, voice track could be edited and manipulated independently.
- Controls and special effects like chorus, echo, reverb, panning are there to set up.
- All these tracks are finally combined into two channel or multiple channel doing these for to play in the individual speaker.

9. DIGITAL AUDIO

Analog signal converted into digital following methods are invoked to provide the sound.

PRACTICAL AUDIO SAMPLING PARAMETERS

Sampling frequency need to be twice the input frequency. Minimum sampling frequency range is from 40KHz slightly higher to this to be provided. Increasing sampling increase accuracy of the digital signal with respect to analog signal ,also increase file size.

- **ALIASING**

Aliasing is a consequence of violating the sampling theorem. If audio frequency is greater than nyquist theorem erroneous signal can appear within the audio bandwidth and it is impossible to distinguish the signal.

- **BIT RATE AND FILE SIZE**

Increasing sampling rate and resolution increase the file size, require high for storage. For 30 sec audio clip require storage space. $(44100 \text{ sam/sec}) \times (30 \text{ sec}) \times (16 \text{ bit/sam}) \times (2 \text{ channel}) = 42336000$ bits.

- **STREAMING AUDIO**

It is used to downloading files on the internet. If music begins to play buffer memory on the receiving device fills up ,while remaining portion continues to be downloaded. A popular type of streaming format is **realnetwork's realaudio**. **Real audio** file can coded at variety of bit rates. **Apple's quick time** software offers several compression option for option for music and speech application for downloading and streaming. **Macromedia's shockwave** is a streaming audio package based on MPEG layer III coding it allow compression bit rate range form 8 to 128 kbps.

- **HIGH FIDELITY(HI-FI)**

Hi-fi is a term describing the reproduction of sound and image almost identical to original in quality ie with minimum distortion and noise it done by RIAA equalization .

RIAA Equalization- this specification established by recording industry association of America for the correct play back of vinyl records based on equalization. Equalization is a process of modifying the frequency envelope the sound. Peaking equalizers change a wide range of frequency around the central point , while shelving equalizers change wide range of frequency by a fixed amount. Graphics equalizer which consists of series of band pass filters with independent gain control for each.

10. SYNTHESIZERS

Synthesizers are electronic instruments which allow us to generate digital samples of sounds of various instruments synthetically without actual instrument present.

Types of Synthesizers

i) FM synthesizers

It generate sound by combining elementary sinusoidal tones to build up a note . This method uses Fourier transform using waveform can be decomposed into its elementary sinusoidal component.

ii) wavetable synthesizers

Wavetable produced sound by retrieving high quality digital recording of actual instruments form memory and playing them on demand. The sounds associated with synthesizers are called patches. And the collection of al patches is called patch map.

Characteristics of Synthesizer

The polyphony of a synthesizer refer to its ability to play more than one note at a time. Polyphony is general measured or specified as a number of notes or voices. A synthesizer is said to be multitimbral if its capable of producing two or more different instrument sounds simultaneously. Polyphony of multitimbral synthesizer is usually allocated dynamically among different parts being used.

11. MIDI (MUSICAL INSTRUMENT DIGITAL INTERFACE) What is MIDI?

The MIDI is a protocol or set of rules for connecting digital synthesizer to each other or to digital computers. Two synthesizer communicate with the MIDI. The information exchange between two MIDI is musical in nature.

MIDI information can also be more hardware specific. MIDI information can be used to indicate the starting and stopping points of song.

MIDI Manufacture Association (MMA)

The MMA is the only source for up to date MIDI specification issues unique manufacture IDs and licenses logos that identify MMA standards.

MIDI specification

Hardware:

- MIDI uses of special five conductor cable to connect the synthesizers ports.
- MIDI cable is specially grounded and shielded to ensure efficient data transmission. MIDI cable is little more expensive. For connencting MIDI devices.
- MIDI connector is there but more recent device use USB or fire Wire connectors.

Messages

MIDI message constitute an entire music description language in binary form. Each word describing an action of musical performance is assigned a specific binary code. To a sound a note in MIDI language you to send a note on message and the assign velocity which determine how fast key is pressed.

File format

To save synthesizer audio in a separate file format called MIDI files.

- MIDI files are extremely compact compared to WAV files. Other advantage of using MIDI generate sounds include the ability to easily edit the music and the ability to change the play back speed and the pitch or key of the sounds independently.
- MIDI files with DLS (downloadable sound) are the ideal solutions for composer of all kinds who want predictable playback of digital audio also need the compactness and interactivity of standard MIDI files for delivering their music

12. BASIC OF STAFF NOTATION

A staff is simply five horizontal parallel lines on which we write music.



Fig. staff

A note is a open dot or letter O turned on its side. A note represents a single tone and a series of notes represent a melody . notes are written on a staff to record a melody on paper.



Fig. Clef

Clefs provided the reference point needed to know what the notes on staff mean. There are three types: G, F, and C. G clef is more commonly called the treble clef. The G clef wraps around the G line.

There are 12 notes in western music. They are labeled using the first seven letters of the alphabet. The remaining five notes are labeled by modifying this letter by adding the word sharp (#) or flat (b) to it.

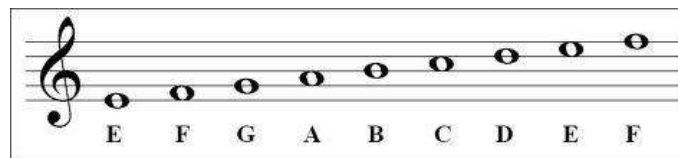


Fig. Notes on the staff

Sharp (#) refers to the key directly next to and above flat (b) refers to the key directly next to and below. A note that is neither sharp nor flat is called natural.

Music is made up of short notes and long notes. The combination of notes into patterns is called rhythm. Silence is also part of music.

To show this silence we use a rest. The time signature denotes how many beats per measure and what note gets one beat. The time signature is placed after key signature. Bars denote measures of appropriate length as determined by the time signature.




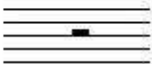



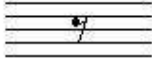
Name	Note	Name	Rest	Beats
Whole Note		Whole Rest		4 beats
Half Note		Half Rest		2 beats
Quarter Note		Quarter Rest		1 beat
Eight Note		Eight Rest		1/2 beat

Fig. length of notes and rests

13. SOUND CARD

The sound card is an expansion board in your multimedia PC which interfaces with the CPU via slots on the mother-board. Externally it is connected to speakers for playback of sound.

Basic Components:

- **Memory Banks**

This depicts the local memory of the sound card for storing audio data during digitization and playback of sound files. When digital audio files need to be played back, the audio data is first stored in the memory buffer being processed by the DAC for conversion to analog format.

- **DSP**

Newer sound cards make use of open architecture which uses a multi-purpose digital signal processor (DSP) as the main controller of all audio signals in the digital domain. The key feature of this system is a dedicated microprocessor specifically designed to meet the computation intensive algorithms that are the backbone of the new audio technology.

- **DAC/ADC**

The digital-to-analog and analog-to-digital converters are for digitizing analog sound and reconvert digital sound files to analog digital fed in from a microphone via the input port of the sound card are converted to the digital form by the ADC following the standard techniques of sampling and quantization.

- **Wave Table/FM Synthesizer chip**

A MIDI Synthesizer chip is necessary to recognize MIDI sequences recorded onto the disk or input from an external synthesizer.

- **CD Interface**

This is the internal connection between the CD drive of the PC and the sound card. This allows a connecting a CD-ROM drive to the sound card. 16-bit ISA Connector:- Interface for exchanging audio data between the CPU and sound card.

I/O PORTS

- **I/P port1: MIC** Input port for feeding audio data to the sound card through a microphone connected to it.

- **I/P Port 2 : Line In** Input port for feeding audio data from external CD? Cassette players for recording or playback.

- **III) O/P Port1: Speakers** Output port for attaching speakers for playback of sound files.

- **IV) O/P Port2: Line out** Output port for connecting to external recording devices like a cassette player or an external amplifier.

- **V) MIDI:** Input port for interfacing with an external synthesizer. This is present in some cards like sound blaster 16. Using this connection, MIDI songs can be composed on the PC using software and then can be sent to the sound modules of external synthesizers for playback.

Processing Audio Files

- **WAV File:**

From the microphone or audio cassette player a sound card receives a sound as an analog signal. The signals go to an ADC chip which converts the analog signal to digital data. The ADC sends the binary data to the memory buffer, from where these are sent to the DSP, which optionally compresses the data so that it takes up less space. The DSP then sends the data to the PC's main

processor which in turn sends the data to the hard drive to be stored. To play a recorded sound the CPU fetches the file containing the compressed data sends the data to the DSP. The DSP decompressed the data and sends it to the DAC chip which converts the data to a time varying electrical signal.

MIDI Files

To handle MIDI files the sound card requires a synthesizer chip which can recognize MIDI instructions and produces corresponding sounds . MIDI files are textual files which can be created by writing instructions using appropriate software . eg., MIDI soft recording session These files are stored on the hard disk just like text files. The instruction contains the information regarding what instruments to play and how they should be played. For playback of the files, the data is send by the main CPU to the DSP chip.

The DSP on receiving the MIDI instructions, generate a query to a synthesizer chip Syn which contains the actual audio samples and retrieves the appropriate samples. The DSP also manipulates the sample parameters in order to change there loudness, pitch etc.

The synthesized sound is then sends to the DAC for conversion to analog form and output via the output port to the speakers. If the card uses FM syntheses instead of wave table the DSP tells the synthesizer chip syn to produce the note.

14. AUDIO TRANSMISSION

To Convey Digital Audio Data In Real Time Between different hardware devices, there must be both a data communication channel and a common clock synchronization.

Aes/ebu

The audio engineering society/European broadcasting union) is a standard for carrying digital audio signal between devices and components .the transmission rate is such that samples of audio data one from each channel are transmitted in time division multiplex in one sample period.

Sony Philips digital interconnect format

It is a standard format for transmission of digital audio signals between devices and components. common uses of SPDIF is to format is transmit compressed audio data from a DVD player to a home theatre system supporting Dolby digital.

Phone audio jack

Common audio connector is the phone jack. The miniature jack and sub-miniature jack were

designed for audio output from transistor radio

RCA pack

One problem with RCA connector is that each signal type requires its own cable which quickly increase the number of cables for a multi device installation XLR audio connector.

15. AUDIO FILE FORMATS AND CODECS WAV (Waveform audio)

- This is the format for sampled sounds defined by Microsoft for use with windows. It is an expandable format which supports multiple data formats and compression schemes.
- WAV is a Microsoft and IBM audio file format standard for storing audio on PCs.
- It is a variant of RIFF bitstream format method for storing data in 'chunks', and thus also close to the IFF and AIFF format used on macintosh computers.
- A WAV file can hold audio compressed using some lossless CODECs like DPCM and ADPCM, but the most common format is PCM audio data.
- Since PCM uses the uncompressed, lossless storage method which keeps all the samples of an audio track, professional users or audio expert may use the WAV format for maximum audio quality.
- There are also efficient, lossless CODECs available, such as Monkeys Audio, FLAC, TTA or Apple Lossless.

AIFF (Audio Interchange File Format)

- The Interchange file format (IFF) is a generic file format developed by Electronic Arts in 1985 in order to facilitate data transfer between software programs of different vendors.
- An IFF file is build up from chunks. Each chunk begin with a TypeID or OSType followed by a 32-bit integer specifying the size of the following data.
- Chunks can hold different data types like text, numeric data and raw data.
- 8SVX (audio)
- AIFF (audio)
- ANIM (animation)
- DOC (pre Word97 text)
- ILBM (raster image)

Audio interface file format: it is a format standard used for storing audio data on PCs. The format was co-developed by Apple based on Electronic Arts file format (IFF).

Types of commonly used chunks in AIFF files include

- Common chunk
- Sound data chunk
- Marker chunk
- Instrument chunk
- Comment chunk
- Name chunk
- Author chunk
- Copyright chunk

MID (MIDI)

MIDI files are textual files which contains instruction on how to play a piece of music. The actual music is generated from a digital synthesizer chip which can recognize the instruction and retrieve corresponding audio samples using a repository of sounds.

AU (Audio)

It is developed by sun microsystem, this audio file format consists of a header of six 32-bit words which defines the metadata about audio data following it.

- Word 0 in an identifier defining the file type
- Word 1 data offset value.
- Word 2 is the data size in bytes.
- Word 3 is the data encoding format.
- Word 4 number of sample per second.
- Word 5 indicates number of channels.

MP3 (MPEG Layer III)

- A highly compressed audio format providing almost CD-quality sound.
- MP3 can compress atypical song into 5 MB for which it is extensively used for putting audio content on internet.

VOC (voice)

- It is used with Sound Blaster sound card.
- Sound up to 16-bit stereo is supported along with the compressed formats.
- These files can contain marker for looping and synchronization markers for multimedia application.

RMF (rich Music Format)

Beatnik is software based high performance music and audio playback technology created by Beatnik Inc. to take full advantage of the unique features of the Beatnik audio engine, Beatnik Inc. developed their own audio file format called Rich Music Format (RMF).

Mp3PRO

An extended MP3 version from coding technologies that uses Spectral Band Replication /SBR in order to increase its efficiency for bitrates below 96 Kbps/stereo. Mp3PRO has been implemented only in few software and hardware products.

WMA (Windows Media Audio)

Windows Media Audio is a proprietary compressed audio file format used by Microsoft. It is a part of Windows Media framework. An initial reason for the development of WMA might have been that MP3 technology is patented and has licensed from Thomson for inclusion in the Microsoft Windows operating system.

- A Wma file is encapsulated in an Advanced system Format (ASF) file.
- The resulting file may have the filename suffix 'wma' or 'asf'

RA (Real Audio)

RealAudio is a proprietary audio format developed by RealNetworks. It is especially designed to conform to low bandwidths, and it can be used as a streaming audio format, i.e. played at the same time as its download. The first version of RealAudio was released in 1995.

➤ The CODECs and versions of RA is

- 8 kbit /s linear predictive (Real Audio 1)
- Code Excited Linear Predication (Real Audio 2)
- Dolby AC3 (Real Audio 3)
- Sipro Lab Telecom ACELP (Real Audio 4/5)
- G2/Cook CODEC (Real Audio 6)
- Sony ATRAC3 (Real Audio 8)
- MPEG-4 (Real Audio 10)
- RealAudio Lossless Format (Real Audio 10)

Ogg Vorbis

Ogg Vorbis is a completely free and open audio compression project from Xiph.org Foundation, and is part of their ogg effort to create free and open multimedia and single processing standards.

Vorbis uses the Modified Discrete Cosine Transform (MDCT) for the converting the sound data from time domain to frequency domain and back.

ATRAC (adaptive transform acoustics coding)

An audio compression algorithm used to store information on minidisc and other sony branded audio players .

- **ATRAC1** : two stacked QMF(quadrature mirror filters) split the signal into three parts: 0 to 5.5125 to 11.025khz,11.025 to 22.05 khz. It support full stereo encoding with a data rate is 292 kbits/s.it can also be used in mono mode, doubling recording time.
- **ATRAC3 LP2 MODE** : This uses a 132 kbit/s data rate . The quality is similar to that 128 kbit/s MP3.
- **ATRAC3 LP4mode** : This reduce the data rate to 66 kbit/s partly by using joint stereo coding and a lowpass filter around 13.5khz.it allow 324 minutes to be recorded on a 80 minute Minidisc .
- **ATRAC3PLUS** : This CODEC is used in HiMD players, memry stick players and ATRAC CD players . it uses huge information window of 4096 samples, four times bigger than that of ATRAC#.

AAC(Advance Audio Coding)

It is lossy data compression scheme intended for audio streams.AAC was designed as an improved performance CODEC relative to MP3 and MPEG-2 part 3. The AAC CODEC specified in MPEG 2 part 7 and the AAC specified in MPEG 4 part 3 , they are both informally known as AAC

SHN(shorten)

SHN is a file format used to lossless compress CD quality audio files.

Shorten files use .shn extension. The shorten algorithm and the reference code that implement it were developed by tony Robinson of soft sound ltd.

The code was made available under generous non commercial and has since been extended to include seek table.

FLAC(FREE LOSSLESS AUDIO CODEC)

Flac is a popular lossless audio compression CODEC. The flac format is currently well supported by many software and hardware products. The FLAC formats consists of the stream format ,libflac ,libflac++,flac,metaflac.

FLAC achieves compression rates of 30-70%. FLAC uses linear prediction to convert the audio samples to a series of small , uncorrelated numbers which stored efficiently using golomb- rice coding.

APE(monkey s audio)

MONKEY AUDIO is a lossless audio compression CODEC. Monkeys audio files use the .ape extension.

Monkey audio file formats suffers from the downside of lossless CODEC such as FLAC virtually always far lower than lossy option.

APEV2 TAG

The APE tag was originally developed for the muse pack file format as a metadata container for stroing information like title ,artisit,track numberetc. The main advantage of the APEv2 tag is very rigid structure.

TTA(true audio)

True audio is a free ,simple real time lossless audio CODEC supporting multichannel 8,16,and 24 bit data of uncompressed WAV input files. The TTA format support both of ID3v1 and ID3v2 information tags . the TTA project provide free and simple format , plugin for the most popular media players, TTA direct show .

ALAC(apple lossless audio CODEC)

Also known as apple lossless encoder(ALE) it is audio CODEC developed by the apple computer for lossless encoding of digital music .the apple lossless encoder was introduced as a component of quick time 6.5.1

WAVPACK

It is a free open source lossless audio compression format developed by david byrant it allow user to compress both 16 and 24 bit audio files in .wav file extension. Wavpack also incorporate a unique hybrid mode tha provide all advantage of lossless compression with an additional bonus.

MPC(muse pack)

It is a open dsource version of MP2 format development of MPC was started in 1997 and currently maintained by MUSE pack. A series of double blind listening tests conducted in 2004 suggested that musepack and ogg vorbis are two bests lossy CODECs at bitrates around 128 kbps compared to MP3.

SPEEX

It is a free software speech CODEC intended to be free of patent restriction. It is a licensed under the BSD licence and is used with the OGG format. It support variable bit rate which means that it can use higher bit rates for specific sounds for improving quality, speex has componenet called VAD (voice activity direction) which allow encoder to detect the presence of speech or silence.

MOD(MODULE)

Module file format to reperesent computer generated music file in term of note number,track numbers,instrument numbers , etc they store several patterns of musical data and also specify how the pattern should be played. The **exteneded module(xm)**audio file type has been introduced by tritons fast trackers it includes multi sampling instruments with volume and panning envelopes.

16. AUDIO RECORDING SYSTEM Dolby system

Analog recording the first product he made a simple compander called dolby type a noise reduction .dolby noise reduction developed in dolby lab for use in analog maganetic tape recording Many types are there like dolby A type NR,dolby Btype NR, dolby Ctype NR, dolby Stype NR. Dolby system also includes coding algorithm to reduce data capacity and bit rates while still delivering professional quality audio.

Digital theatre system DTS co owned and was co founded by film director Steven provide more technical sounds. DTS digital surround providing 5.1 channel of discrete digital audio in consumer electronics and software. DTS extended surround(ES) the only digital audio format capable of delivering 6.1 discrete audio . it is also fully backwards compatible with DTS.

DTS NEO 6 Its provides up to 6 channels of matrix decoding from stereo matrix martial

DT96/24- offers And unprecedented level of audio quality for multichannel sound on DVD video with on ddts encoder

17. AUDIO AND MULTIMEDIA Types of audio in a presentation

1) Speech

Speech is an important element of human communication and can be used effectively to transmit information,

Advantage

- Using natural speech is the power of the human voice to persuade.
- That speech can potentially eliminate the need to display large amount of text on the screen.

Types of speech

1. Digitized → provide high quality, natural speech
2. Synthesized → not sound as natural as human speech

2) Music

- Music is also an important component of human communication
- Music is used to set a tone or mood, provide connections or translations
- Music when combined with speech and sound can greatly enhance on-screen presentation of text and visuals.

3) Sound effect

- Sound effects are used to enhance or augment the presentation of information or instruction.

Two types of sounds

- 1) Natural → sounds are the commonplace sounds that occur around us
- 2) Synthetic → sounds are those that are produced electronically or artificially.

General categories of sounds

- 1) Ambient → sounds are the background sounds that communicate the context of the scene to the listener.
- 2) Special → sounds are uniquely identifiable sound such as the ring of the telephone or the slam of the door.

Reasons for using audio

The main reasons to use audio is to ensure that the goal and the objectives are met.

Conditions influence the effectiveness

- Information presented in each mode should be congruent
- Identical presentations of worlds in sound and teat should be avoided.
- The scone reason of including audio content is connected to motivation.

Suggestion for using audio content

Speech: to produce high recorded speech a script should be written and professionally recorded. High quality microphones and a professional narrator with clear diction and the ability to correctly pronounce the words are plus points.

Effective a narrator should

- Vary information to motivate, explain, emphasize,
- Use a conversational tone
- Be amiable, sincere, straight forward
- Avoid a lecturing tone
- When speech s being recorded all background noise and ambient sound be eliminate unless it is used to provide a realistic environment.

- To develop narrative scripts for integrating speech, multimedia developers should:
- Write the way people speak
- Use language the audience can understand
- Write in a clear straightforward manner
- Use second person pronouns
- Avoid slang
- Use humor when appropriate
- Interpret what the user is seeing rather than just describing it.
- Time limit requirement

Music

A wide variety of prerecorded music is available in digital form on CDs.

In addition original music can be recorded and edited by connecting synthesizers.

- Establish mood.
- Signal a turn of events
- Provide transitions and introductory information
- Emphasize important points
- Support visual information.

Sound effect:

Sound effect provide information about the following

- **Physical event:** click of the keys of the keyboard, ringing of a telephone or breaking of a glass.
- **Dynamic changes:** as we pour liquid into a glass, we can hear when it is full.
- **Abnormalities:** a car engine malfunctioning
- **Spatial relations:** judging the distance and directions of a person walking by the sound of the footsteps.
- **Govern the use of sound effect:** They must be clear and easily identifiable They should not overwhelm the primary message. They should be appropriate to the intended audience.

18. VOICE RECOGNITION AND RESPONSE

- Computer to understand spoken word and respond verbally too these capabilities are known as **voice recognition** and **voice response**
- These can be used as input/output mechanism in conjunction with the normal I/O devices. Like keyboard and mouse.
- In the voice recognition area, products are typically categorized as utilizing either **continuous recognition** or **discrete recognition** as being able to handle a small or large vocabulary and as either **speaker-dependent** or **speaker-independent**.
- **Speaker dependent** system the system need to be trained to recognize each users voice patterns while for a speaker independent system a general voice model is sufficient.
- **Speaker independent** system can nowadays recognize and respond to hundreds of words on a single Digital signal Processing (DSP)

19. AUDIO PROCESSING SOFTWARE

An audio editing s/w allow you to open, edit, manipulate, translate, transforms, and save digital audio sound files in various formats.

1) opening an existing sound files

After opening, the sound file would be visible as an audio waveform with time along the horizontal axis and amplitude, usually measured in decibels, along the vertical axis. File formats like AIFF may also be supported.

2) playing a file

A playback head is usually seen moving across the displayed wave form while the corresponding sounds is heard on the speakers fig(a) along with the play button there would be other accessory buttons like pause , rewind and forwards buttons

3) Playing selected portions of a file

For playing a selected portion of fig (b) a file the user will have to use the mouse select a specific portion of the waveform and click on the button. In the case instead of playing the entire file only the selected portion is played.

4) Accurately positioning the playback head

The user can select portion of the file by dragging with the mouse pointer using eye estimation. But the editors usually offer some way of accurately positioning the head at a specific point of the file by mentioning the time in hh: mm :ss format.

5) Copying and pasting portions of a file

The editors offer a way of copying and pasting portions of a file either in the same file or in a different file. The user select a portion of a file and then chooses the copy function whereupon the selected portion is copied to the clipboard. The portion can then be placed at a different point in the same file or in different file. Fig(c)

6) Saving a file

The user can save an edited file by specifying the filename and location. Usually the native WAV format is supported by all editors, and a number of additional formats may also be supported.

7) Using cut, trim and undo functions

- The cut function enables the user to select a portion of the audio file and discard that portion
- The file duration will be shorted in the case
- The trim function allows one to select a portion of a file and discard the remaining portion.

8) Magnifying and zooming .

This is similar to the magnification tool of an image processor in that it does not change the actual data of the file, only how the file appears to the user.

9) Mixing sounds

When one portion of a selected sound is pasted in another file, the two portions are heard one after another but not simultaneously and the total duration of the sound remains unchanged.

The procedure involves selecting a portion of a sound file and dragging it to another file using the mix function.

10) Crossfading sounds

- Is a special case of the mix function where the amplitude of the sounds are varied after mixing
- One of the sound starts from a low amplitude and gradually grows to the full intensity.

11) Converting between mono and stereo formats

- The user should be able to convert between mono and stereo sound formats
fig(d) when converting from mono to stereo a single mono channel is duplicated as dual channels.

12) Changing the sampling rate

- When converting from a higher sampling rate to a lower value, a set of sample values are discarded by the editor. When converting from a lower sampling rate to a higher value the conversion algorithm tries to interpolate the existing sample values to generate new value.

13) Changing the depth

- An editor also allow the user to change the bit depth of the sound. 2-values 8-bit and 16- bit are supported by most editors.

14) Recording

- Most sound editors have provision for recoding audio. Recording means an external recording means using an external device and connecting it to the computer for recording purpose.

2- types

1) recording voice and music

2) to record voice of a person

- A micro phone needs to be connected to the input port of an external playback device.

15) Normalization

- Normalization also plays another important role in a different context.
- When different part of a voice recording is done at different times or on different days, the volume of the voice tend to vary.
- This would create a disturbing effect when the entire recording is played once.

16) Changing duration

The time duration of a digital audio file may be changed in two ways

- 1) In the first method the wave may be stretched or compressed to change the duration.
- 2) To change duration without changing the pitch , sample values are either duplicated or discarded .

17) Special effects

- Most of the audio editing software allow application of filters for changing the nature of a sound clip in some pre-determined way , mostly for giving special effects. E.g. filters include echo (single echo), reverb(multiple echo),chorus.
- In some editors the user may also be allowed to change the wave form manually by dragging different points of the wave
- Noise is any unwanted signal that gets mixed the original sound due to various reasons, eg: environmental sounds, electrical interference, low-quality device ect.

18) Embedding markers

- Most of the sound editing software allows inserting of textual markers. This enables quick location of a specific portion of the without listening to it.

UNIT-V: Video: Analog Video Camera – Transmission of Video Signals – Video Signal Formats – Television Broadcasting Standards – PC Video – Video File Formats and CODECs – Video Editing – Video Editing Software. Animation: Types of Animation – Computer Assisted Animation – Creating Movement – Principles of Animation – Some Techniques of Animation – Animation on the Web – Special Effects – Rendering Algorithms. Compression: MPEG-1 Audio – MPEG-1 Video - MPEG-2Audio – MPEG-2 Video.

1.VIDEO

Motion video is combination of image and audio. An image or a paragraph is said to be a still image or frame.

2. ANALOG VIDEO CAMERA

- Monochrome video camera
- color video camera

Monochrome video

The essential component camera consists of a vacuum tube consisting of an electron gun and photosensitive semiconductor plate. The electrons migrate to words positive potential and it is applied to the thin layer of conductive transparent material. Thus a charge pattern appears of the inner surface which produces brightness of the scene.

This consists of three camera tube with select into primary colors. Each tube develops a signal voltage proportional to their respective color intensity received by it.

3. TRANSMISSION OF VIDEO SIGNAL

- Luminance
- Chrominance

Luminance:

Luminance component describes the variation of perceived brightness by HVS [Human Visual System]. In different portions of the image without regard to any color information. It is denoted by Y.

Ex: The image black and white effect as grey scale.

Chrominance:

This component describes the variation of color information in different parts of the image. This is denoted by c.

4. VIDEO SIGNAL FORMAT

- The component video
- The composite video
- s-video
- SCART-video connect

Component video

This refers to the video signal. It is transmitted as three separate component signals [RGB]. The form output of analog video cameras.

Composite video

This signal transmission especially used for TV broadcasting and cable channel requirements. It has three pin connectors to the Red, Yellow and White.

S -video [super video]

It is an analog video signal where the luminance and chrominance are transmitted separately using multiple wires.

SCART –video connect

[Syndicatdes Constructeursd Appareils Radiorecepteurs et Televiseurs] it is a French standard with 21 pin audio and video connected. This is used connect VCR, video player, set of boxes, computer games and computers and two-television sets.

5. TELEVISION BROADCASTING STANDARDS

- NTSC
- PAL
- SECAM

NTSC

It is named after the national television system committee. Which is a television broadcasting system used in no. of countries including carrier, japons, Canada, north America, south America, Mexico and Caribbean islands.

PAL

Pal is the phase alternating lines it is a TV broadcasting standards used in Europe, Asia, Australia & Africa.

SECAM

Sequential color and memory, it is a French standard which means sequential color with memory used in France, Russia and Middle East.

DIGITAL VIDEO

Digital video formats

- EDTV
- CCIR
- CIF
- HDTV

EDTV : [Enhance definition television system]. This is a conventional system modify to offer improved vertical and Horizontal resolutions.

CCIR : [consultative committee for international radio Communications]. It is a color video that signals the three components. That is One Luminance component and two chrominance component.

CLF: [Common Intermediate Format]. It has the resolution in both horizontal and vertical resolution and used in video telephony application.

HDTV: [High definition television]. It is new standard for improving picture quality. It requires high definition monitor on TV screen.

6. PC VIDEO

Analog video needs to be converted to the digital format before it can be displayed on PC screen.

Source and source devices

‘source’ implies the media on which analog video is recorded. The source and source device can be one of the following

- Camcorder with pre-recorded video tape.
- VCP with pre-recorded video cassette
- Video camera with live footage.

Video capture card

A video capture device is essentially an expansion board that can handle a variety of different audio and video input signals. Convert them from analog to digital or vice versa.

Digitizing video

- Video input port to accept the video input signal from NTSC/PAL/SECAM broadcast signals, video camera or VCR. The input port may conform to the composite-video or S-Video standards.
- Video compression

7. VIDEO FILE FORMATS and CODECS

- AVI
- MOV
- MPEG
- REAL VIDEO
- H.261
- IVL
- CINE PAR
- SORENSON VIDEO
- VDOLIVE
- NERO DIGITAL
- OGG THEORER
- WMV

AVI

Audio video interleave. [it is video file format and windows platform.]

Mov

Movie it is the quick time movie format developed by apple system

MPEG

Motion picture experts group

REAL VIDEO

It is the real network video

H.261

It is divided for video telephony and video conferencing application

IVI

An video interactive used for video distributed over internet

CINE PAR

Developed to play small movie

SORENSEN VIDEO

It is high quality data rate and better picture quality.

VDOLIVE

Video delivery by live serve based

NERO DIGITAL

It is the software product it is use MPEG4.

OGG THEORER

It is video code developed to with vp3 code and on technology

WMV

Windows media video

8. VIDEO EDITING

- **Time base:** In the natural world we experience time as a continuous flow of events. However, working with video requires precise synchronization so it is necessary to measure time using precise numbers.
- **SMPTE Time code:** It defines how frames in a movie are counted and affects the way we view and edit a clip. A standard way to represent time code is developed by a global body called Society of Motion Pictures and Television Engineers (SMPTE)
- **Online and offline editing:** It is a practice of doing all editing on the same computer that will produce the final cut. Previously online editing had to be done on expensive high end work station design to meet the picture quality and data processing requirement.

9. VIDEO EDITING S/W

- **Importing clips:** For editing the user is provided with an interface of importing the video clips into the software package.
- **Playback a clip:** Once the clips are placed on the timeline each of them can be selected and played back in a monitor window
- **Split a clip:** A video editor allows a user to spilt a video clip into any number of parts. On a timeline a video clip is represented by a rectangular shape
- **Manipulating the audio:** A video editor enables a audio content of a video clip to be separated from the visual content

- **Adding transitions:** A video editor enables user to insert various types of transition between to video clips

10. ANIMATION

Animation means giving life. Animations are created from sequence of still images. Each image is slightly different from the previous image. Images are display rapidly in succession. So that it produces continues motion due to persistence of vision.

Key Frames And Tweening

A Frame is a individual image created. A set of images are created to show the action change. The Intermediate frames between the key frames are drawn by animators. This process is called tweening.

11. TYPES OF ANIMATION

- Cell animation
- Path animation
- 2-D animation
- 3-D animation

Cell animation

Cell animation is term it refers to transparent piece of film or picture that is used in hand drawn animation. That is one layer on the top of another layer .

Path animation

Here an image is created and it moves through an sprite (path).Path animation is known as sprite animation. A sprite is a motion path typically curve, Zigzag, spinning, straight and so on.

2-D animation & 3-D animation

- A 2-D picture has axis x & y.
- 3-D picture has extra one direction called as z direction, which gives the liveliness to the picture that is the depth of the picture.

12. COMPUTER ASSISTED ANIMATION

Computer assisted animation is also based on the key frame concept. The great advantage of computer assisted animation is that in between frames are created by the animation program itself, not by junior animation doing a lot of tedious work.

Computer animation software typically simulates the process of traditional cell based animation , by placing the animated object on a timeline to depict the flow time.

13. CREATING MOVEMENTS

- Co-ordinate systems.
- Transformation.
- 1. Scaling
- 2. Translation
- 3. Rotation

Co-ordinate system

The object from the originate position is being transformed to the new Co-ordinate points.

Transformation

- **Scaling** : It is enlarging or shrinking the given object.
- **Translational** : The movement of an object in any direction from the original one is called translational.
- **Rotation** : It is rotating an object through a particular angle.

14. PRINCIPLES OF ANIMATION

- Squash and Stretch
- Anticipation
- Staging
- Overlap
- Slow in / Slow out
- Arcs
- Timing
- Exaggeration
- Appeal

Squash and stretch

An animation of a bouncing ball does not change shape. It's not lively therefore realistic effect must be given by flattening the ball and it strikes the ground and the shape comes up. This is called squash and stretch.

Anticipation

Making the motion of an object clear and prepared.

Staging

It is a concept that gives an emphasis and integrate with the background.

Eg: Emboss effect and so on.

Overlapping

While producing an effect like holding hands together the first hand bcome with the fraction of second gap the next hand appears. so that overlapping is avoided for output come perfect.

Slow in / Slow out

While bouncing a ball the effect has to be differentiated like the ball is fast in hitting down and it is slow while racing up.

Arcs

The movement of an object follows the arc drawn.

Timing

Setting time for the object to be translated.

Exaggeration

Exaggeration is an action that makes something startling.

Eg: popping of eyes, shock, amazing, wondering, confused state.

Appeal

It is taking the attention of the user. So that it is lively and concentrated.

15. SOME TECHNIQUES ANIMATION

- Onion Skinning
- Motion Cycling
- Masking
- Adding Sound
- Flip Book
- Rotoscoping
- Blue Screening
- Color cycling
- Morphing
- **Onion Skinning:**

Onion skinning is a drawing technique. That shows the object traveling in a sequence of path and it is transparent cell that follows the path.

- **Motion Cycling:**

The normal animal or human motion like walking, running is a repetitive action which require 10 to 12 frames minimum. This motion cycle is divided in two half's and the variation is carried out in between. Thus the target shows the moving cell.

- **Masking:**

Masking a computer animation effect that protects the parts of a frame from an editing effect. This technique is used to make an animated object move behind the protected area.

- **Adding Sound:**

Sound is an important enhancement for moving images. Background music involve emotions therefore the visual is effective only if sound factor is added. Eg:

Sound produce when opening a door, Hitting the door, Burst of balloon, crackers bursting, clapping hand etc.

- **Flip book animation:**

A flipbook is a book the series of picture varying gradually from one to another with each picture on one page. So that when the page turned rapidly the picture turned to animate.

- **Rotoscoping:**

It is a specials effect which enables the animator to trays the color of object of each frame of an animation and video sequence to create the outcome. This outcome is added with some special visual effect.

Eg: Bullets generated from a gun.

Sparkers generated while sword being hitted.

- **Blue Screening:**

This technique using for shooting live action against a blue color background and then replacing the background by another image. The colors specified blue,fluresent green ans orange.

- **Color Cycling:**

- This technique allows to change the color of the object by cycling through a range of colors.For this RGB color wheel is used.This software provides smooth color transition from one color to another.

- **Morphing:**

Morphing is a special process of smoothly interpolating between different images. Eg: Consider two faces Human face , Animal face. When played back it appears that the human face gradually and seamlessly changes in to the animal face.

16. ANIMATION ON THE WEB

- **Shockwave format**

Shockwave technology was the first animation plug in for browsers. Eg: Macromedia shockwave(s/w) , Macromedia Director(s/w).

- **Macromedia Flash**

Macromedia flash is an animation sequence which creates graphics and motion animation and the frame is saved in shockwave format(.swf)

Eg: Movie.swf

- **Client Pull Animation:**

As the client makes the first move by clicking the button the browser instruction is requested and closed another document automatically on a click of a button. So that the next frame is generated for animation.

- **Server Push Animation:**

This is automatically created by the server which the HTML source code and server as a new document or an image on the web page.

17. SPECIAL EFFECTS Atmospheric Effects

Many 3D computer graphics image have a distinctive crystal- clear quality.

Sometimes this clarity is intended by the artist but sometimes it is not desirable and creates an appearance of unreality and falseness.

Particle Effect

Many phenomenons like smoke, gas, fire, steam and clouds cannot be modeled easily as surface. The particle system technique found on many 3D system packages handles these types of phenomenon that consists of masses of molecule-sized particles rather than surfaces.

18. RENDERING ALGORITHM OF ANIMATION

- **Ray casting Algorithm**

This algorithm determines the color of given pixel and it divide the common approach to look at each pixel and calculate the color render by the object.

- **Shading Algorithm:**

Shading calculates the color at a point and shading of surface. Shading is a process where the surface is directly faces to the light. It shown bright and those facing away from the light shown dark. According to this algorithm the surface is directly faces the light will appear bright and lightly shaded. Where as the point the seen or area behind the seen is differentiated with darker color or darker edges.

- **Z-Buffer Algorithm**

This procedure calculates the distance of each object from the observer. If a point on the surface is far away from the observer then the Z-buffer stores the largest number to that object. According to Z-buffer the top most object takes the 1st priority and the under most surface object takes the last priority. Hence the hidden objects are placed last followed by overlapping objects.

- **Ray Tracing Algorithm**

When you look at a specific point on the surface the color at that point is effected by the light falling on the object from the light source. In this technique, the ray is projected from an eye through the pixel of the first object followed by the second and third. Reflections and transparency are automatically calculated by this method.

- **Polygon Subdivision Algorithm**

Polygon subdivision is otherwise called as rendering Subdivision.If a particular object is hidden in a work area then by applying polygon subdivision the main work area is divided in to 4.It keep searching all the four and at last found in one half. This half is concentrated much get to the object. It is a game divided in 4 half and the process keeps going until the target object is retrieved.

- **Aliasing Algorithm**

A digital image is composed of the grid of tiny rectangular pixels a slanted edge of an object appears as a stepped pattern producing this step wise effect is called aliasing.

19. COMPRESSIONS

MPEG (Motion Picture Experts Group)

The ISO (International Standard Organization) and IEC (International Electro Communication) formed MPEG in 1988.

In MPEG Audio

- MPEG1 Audio and Video
- MPEG2 Audio and Video.

In MPEG1 Audio :

This is a compression algorithm for digital audio. There are 3 layers to do the level of complexities.

- **Layer 1:** The speed is 192KBPS with 4.2 compression and the outcome is same as digital audio cassette.
- **Layer 2:** 128KBPS with 6.1 to 8.1 compressions. The quality is same as digital audio broad casting.
- **Layer 3:** 64KBPS compression is 10.1 to 12.1 and quality is same as CD audio. MPEG1 Audio defined in ISO recommendation has 11172.3 standards. This 11172.3 standards describes 3 layers of coding different application.
- MPEG1 Audio layer 1 (MP 1)
- MPEG1 Audio layer 2 (MP 2)
- MPEG1 Audio layer 3 (MP 3)

- **MP1:**
Describes the sophisticated method that requires relatively high data rate of about 192 KBPS and the compression ration is 4.1.the quality is same as audio cassette.

- **MP2:**
It was 1st initiated by Fraunhofer and Mp2 was financed by the European eurecka program.Mp2 first appeared on Internet in 1993 and was often played back using Xing MPEG player.

- **MP3:**
The single three minutes CD track require more than an hour for downloading through a 56K modem.Mp3 files are stored on a hard disc and recorded to CD ROM using flash memory.Mp3 algorithms includes decoding and encoding method for encryption and decryption players.

MPEG1 Video :

It was completed in 1991 with ISO and IEC specification. This provides VHS quality of video and audio and developed for CD ROM. The MPEG1 video used 3 different frames.

- I frame I – Intracoded
- P frame P Predictive

- B frame B_ Bidirectional

To generate frames Eg: Opening a door three main frames like IBP are created within which there are a sequence of inter related frames between. Therefore the reference frame is transmitted first followed by the next and the next and so on. Sequence of frame can be shown as follows Eg: IBPBBBPPBBI.

MPEG 2 Audio

Is designed for application from digital HDTV to internet downloading. This has standards of MPEG1 with the same encoding and decoding principles and thus named as backward compatible(BC)..So it is also known as MPEG2.

MPEG2 Audio uses the five channel approach and has an Advanced Audio Coding (AAC).Hence this sort of 5 channel usage is said to be MPEG2 AAC.

MPEG 2 BC :

It is a multi channel sound with 5 full bank with channel and 1 low frequency channel. Hence it is referred as 5.1 channel.

MPEG 2 AAC:

This is considered to be an improved channel compared to MPEG 2 BC. It has three different coders and implements complexity and performances. The main application areas are:

- Internet Audio
- Digital Television Audio
- Digital Radio and all portable play back devices.

The standards to MPEG 2 AAC is Dolby stereo system (Audio).

MPEG 2 Video

This was developed to provide video quality up to HDTV quality. The bit rates are 2 to 15 MBPS. Most important features include:

- Interlaced Video
- Frame coding
- Pan and Scalability

Pan: Pan is a provider for parameters to display a specified outcome within a frame.

The levels of MPEG 2 video:

There are 4 levels of video resolution Low level , Main level , High level and High 1440..

- Low level : resolution is 352 by 288 pixels. 352/288.

AJK COLLEGE OF ARTS AND SCIENCE

Navakkarai, Coimbatore - 641 105

- Main level : resolution is 720/576 pixel.
 - High 1440 : resolution is 1440/1152 pixel.
 - High level : resolution is 1920/1152 pixel.
-