SCS1302 COMPUTER GRAPHICS & MULTIMEDIA SYSTEMS

(Common to CSE & IT)

UNIT I BASICS OF COMPUTER GRAPHICS

Output Primitives: Survey of computer graphics – Overview of graphics systems – Line drawing algorithm –Circle drawing algorithm – Curve drawing algorithm - Attributes of output primitives – Anti-aliasing

Computer Graphics

Computer Graphics involves creation, display, manipulation and storage of pictures and experimental data/models or images for proper visualization using a computer. Such models come from diverse and expanding set of fields including physical, biological, mathematical, artistic, and conceptual/abstract structures.Typical graphics system comprises of a host computer with support of fast processor, large memory, frame buffer and

•Display devices (color monitors),

• Input devices (mouse, keyboard, joystick, touch screen, trackball)

• Output devices (LCD panels, laser printers, color printers. Plotters etc.)

•Interfacing devices such as, video I/O, TV interface etc.

Application of Computer Graphics

Computer-Aided Design for engineering and architectural systems

- Objects maybe displayed in a wireframe outline form. Multi-window environment is also favored for producing various zooming scales and views.
- Animations are useful for testing performance.

Presentation Graphics

- To produce illustrations which summarize various kinds of data. Except 2D, 3D graphics are good tools for reporting more complex data.

Computer Art

- Painting packages are available. With cordless, pressure-sensitive stylus, artists can produce electronic paintings which simulate different brush strokes, brush widths, and

colors. Photorealistic techniques, morphing and animations are very useful in commercial art. For films, 24 frames per second are required. For video monitor, 30 frames per second are required.

Entertainment

- Motion pictures, Music videos, and TV shows, Computer games

Education and Training

- Training with computer-generated models of specialized systems such as the training of ship captains and aircraft pilots.
- Drawing Maps, Weather Maps, Satellite Imaging, Medical Imaging, Flight Simulation, Machine design, Photo enhancements etc.

Visualization

- For analyzing scientific, engineering, medical and business data or behavior. Converting data to visual form can help to understand mass volume of data very efficiently.

Image Processing

- Image processing is to apply techniques to modify or interpret existing pictures. It is widely used in medical applications.

Graphical User Interface

- Multiple window, icons, menus allow a computer setup to be utilized more efficiently.

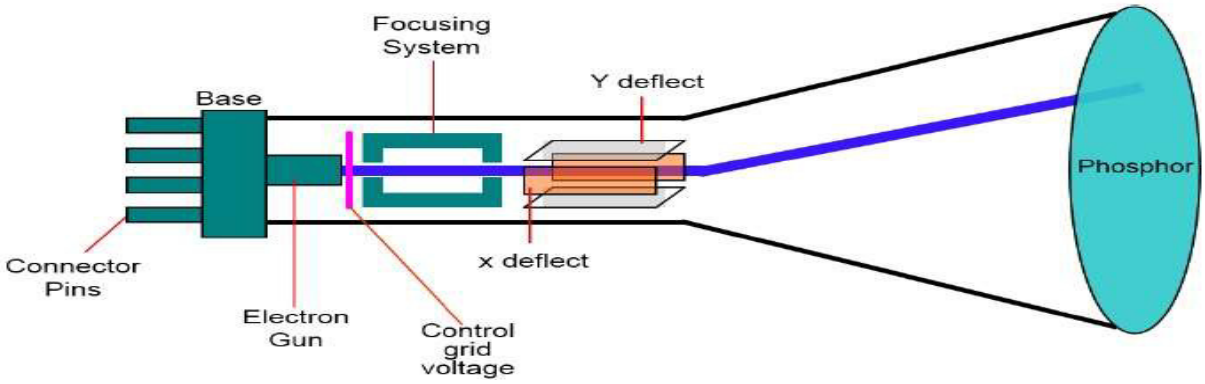
Video Display devices

Cathode Ray Tube (CRT)

The primary output device in a graphical system is the video monitor. The main element of a video monitor is the **Cathode Ray Tube,** shown in the following illustration.

The operation of CRT is very simple:

- The electron gun emits a beam of electrons (cathode rays), when the filament is heated.
- Intensity of the electron beam is controlled by setting voltage levels on the control grid.
- The electron beam passes through focusing and deflection systems that direct it towards specified positions on the phosphor-coated screen. The focusing system is needed to force he electron beam to converge into a small spot as it strikes the phosphor screen, else the bam would spread out as it approaches the screen.
- When the beam hits the screen, the phosphor emits a small spot of light at each position contacted by the electron beam.
- It redraws the picture by directing the electron beam back over the same screen points quickly. This is called refreshing, hence the CRT is called as Refresh CRT.
- The difference between the kinds of phosphors is their persistence- how long they continue to emit light after the CRT beam is removed.



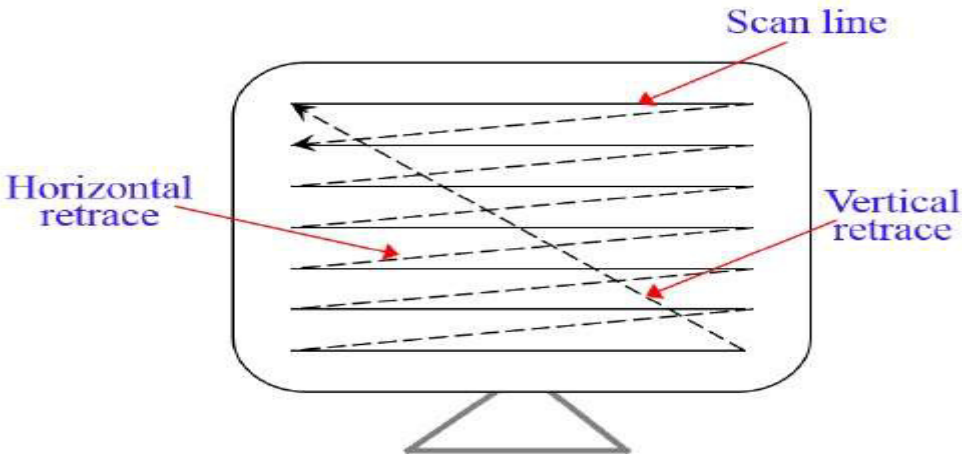
There are two ways (Random scan and Raster scan) by which we can display an object on the screen.

Raster Scan

In a raster scan system, the electron beam is swept across the screen, one row at a time from top to bottom. As the electron beam moves across each row, the beam intensity is turned on and off to create a pattern of illuminated spots.

Picture definition is stored in memory area called the **Refresh Buffer** or **Frame Buffer**. This memory area holds the set of intensity values for all the screen points. Stored intensity values are then retrieved from the refresh buffer and "painted" on the screen one row (scan line) at a time as shown in the following illustration.

Each screen point is referred to as a **pixel (picture element)** or **pel**. At the end of each scan line, the electron beam returns to the left side of the screen to begin displaying the next scan line.

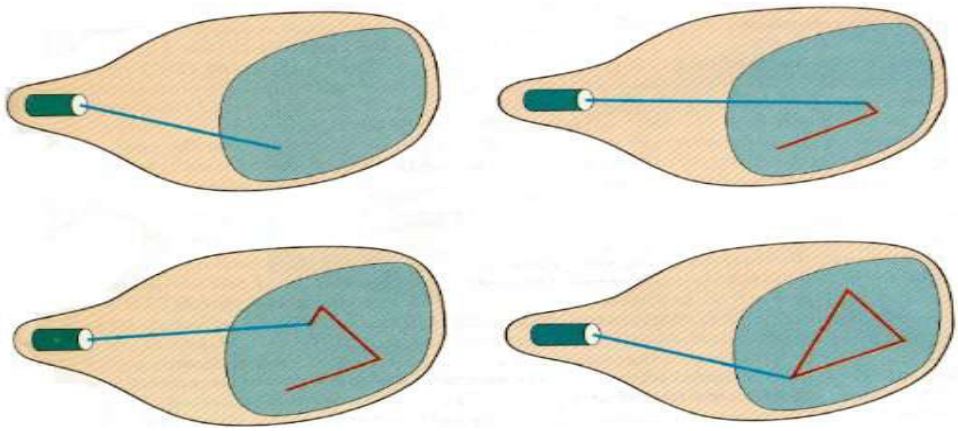


Random Scan (Vector Scan)

In this technique, the electron beam is directed only to the part of the screen where the picture is to be drawn rather than scanning from left to right and top to bottom as in raster scan. It is also called **vector display, stroke-writing display,** or **calligraphic display**.

Picture definition is stored as a set of line-drawing commands in an area of memory referred to as the **refresh display file**. To display a specified picture, the system cycles through the set of commands in the display file, drawing each component line in turn. After all the line-drawing commands are processed, the system cycles back to the first line command in the list.

Random-scan displays are designed to draw all the component lines of a picture 30 to 60 times each second.



Random Scans

Color CRT Monitor

A CRT monitor displays color picture by using a combination of phosphor that emit different-colored light. By combining the emitted light from the different phosphor, a range of colors can be generated. The two basic techniques for producing color displays with a CRT are the beam-penetration method and the shadow-mask method.

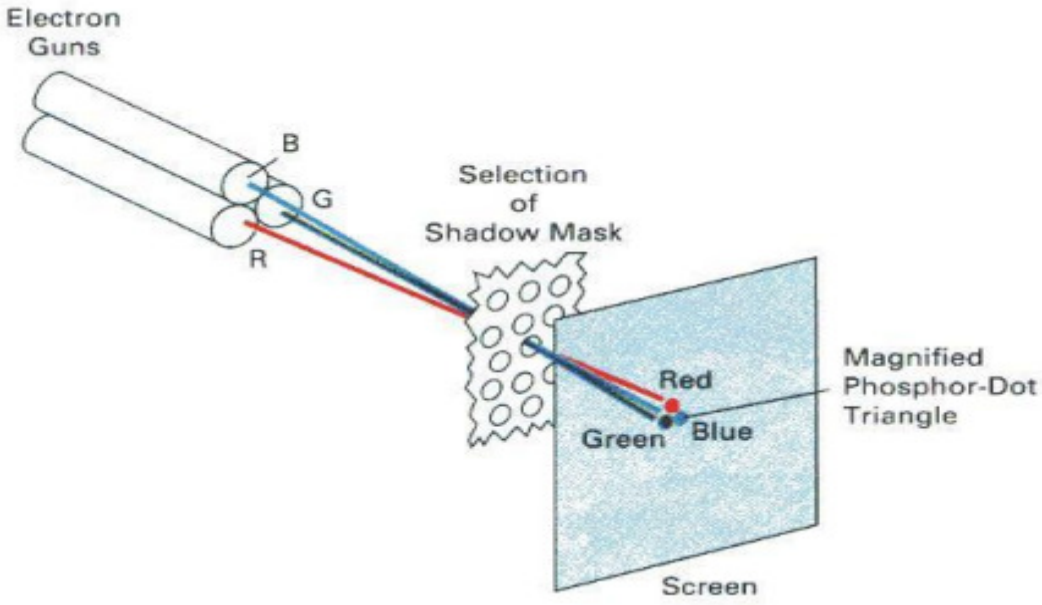
Beam-penetration method

The beam-penetration method for displaying color pictures has been used with random-scan monitors. Two layers of phosphor, usually red and green, are coated onto the inside of the CRT screen, and the displayed color depends on how far the electron beam penetrates into the phosphor layers. A beam of slow electrons excites only the outer red layer. A beam of very fast electron penetrates through the red layer and excites the inner green layer. At intermediate beam speeds, combinations of red and green light are emitted to show two additional colors, orange and yellow. The speed of the electrons, and hence the screen color at any point, is controlled by the beam-acceleration voltage. Beam penetration has been aninexpensive way to produce color in random-scan monitor, but only four colors are possible, and the quality of picture is not as good as with other methods.

Shadow Mask Method

Shadow-mask methods are commonly used in raster-scan system (including color TV) because they produce a much wider range of colors than the beam penetration method. A shadow-mask

CRT has three phosphor color dots at each pixel position. One phosphor dot emits a red light, another emits a green light, and the third emits a blue light. This type ofCRT has three electron guns, one for each color dot, and a shadow-mask grid just behind the phosphor-coated screen. Figure illustrates the delta-delta shadow-mask method, commonly used in color CRT system. The three beams are deflected and focused as a group onto the shadow mask, which contains a series of holes aligned with the phosphor-dot patterns. When the three beams pass through a hole in the shadow mask, they activate a dot triangle, which appears as a small color spot on the screen. The phosphor dots in the triangles are arranged so that each electron beam can activate only its corresponding color dot when it passes through the shadow mask. Another configuration for the three electron guns is an in-line arrangement in which the three electron guns, and the corresponding redgreen-blue color dots on the screen, are aligned along one scan line instead of in a triangular pattern. This in-line arrangement of electron guns is easier to keep in alignment and is commonly used in high-resolution color CRTs.

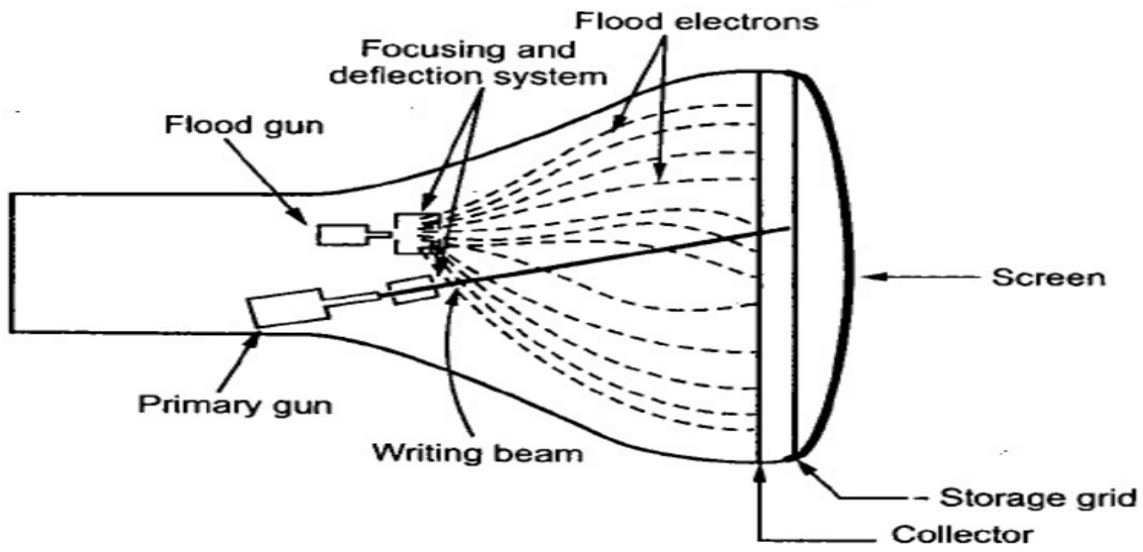


We obtain color variations in a shadow-mask CRT by varying the intensity levels of the three electron beams. By turning off the red and green guns, we get only the color coming from the blue phosphor. Other combinations of beam intensities produce a small light spot for each pixel position, since our eyes tend to merge the three colors into one composite. The color we see depends on the amount of excitation of the red, green, and blue phosphors. A white (or gray) area is the result of activating all three dots with equal intensity. Yellow is produced with the green

and red dots only, magenta is produced with the blue and red dots, any cyan shows up when blue and green are activated equally. In some low-cost systems, the electron beam can only be set to on or off, limiting displays to eight colors. More sophisticated systems can set intermediate level for the electron beam, allowing several million different colors to be generated.

Direct – View Storage Tubes (DVST)

In raster scan display, we do refreshing of the screen to maintain a screen image. The DVST gives the alternative method of maintaining the screen image.A DVST uses the storage grid which stores the picture information as a charge distribution just behind the phosphor-coated screen.The figure below shows the general arrangement of the DVST. It consists of two electron guns: a primary gun and a flood gun.



The primary gun stores the picture pattern and the flood gun maintains the picture display.

The primary gun produces high speed electrons which strike on the storage grid to draw the picture pattern. As electron beam strikes on the storage grid with high speed, it knocks out electrons from the storage grid keeping the net positive charge. The knocked out electrons are attracted towards the collector. The net positive charge on the storage grid is nothing but the picture pattern. The continuous low speed electrons from flood gun pass through the control grid and are attracted to the positive charged areas of the storage grid. The low speed electrons then penetrate the storage grid and strike the phosphor coating without affecting the positive charge

pattern on the storage grid. During this process the collector just behind the storage grid smooth's out the flow of flood electrons.

Advantages:

- Refreshing of CRT is not required.
- Because no refreshing is required, very complex pictures can be displayed at very high resolution without flicker.
- It has flat screen.

Disadvantages:

- They do not display colors and are available with single level of line intensity.
- Erasing requires removal of charge on the storage grid Thus erasing and redrawing process takes several seconds.
- Selective or part erasing of screen is not possible.
- Erasing of screen produces unpleasant flash over the entire screen surface which prevents its use of dynamics graphics applications.
- It has poor contrast as a result of the comparatively low accelerating potential applied to the flood electrons.
- The Performance of DVST is somewhat inferior to the refresh CRT.

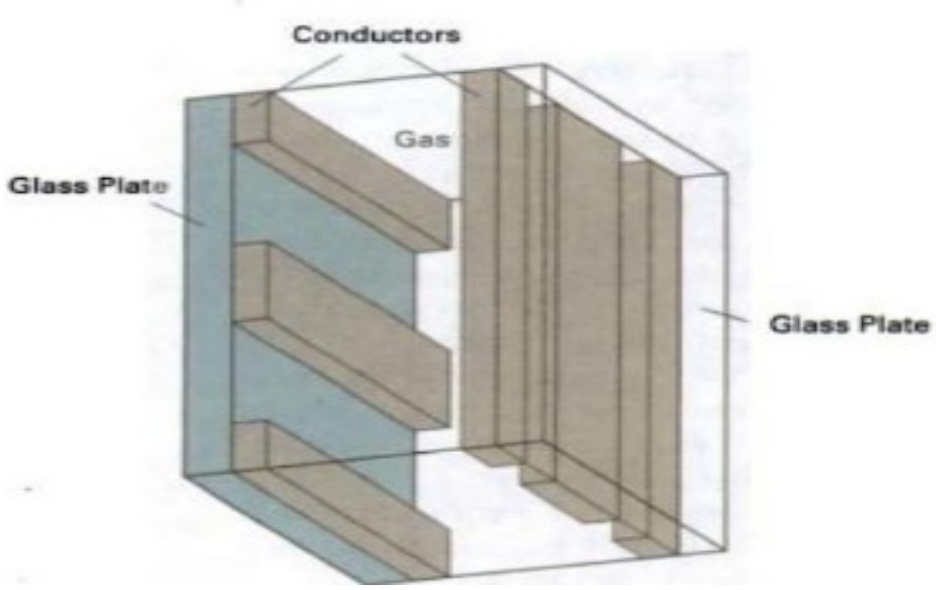
Flat Panel Displays

Although most graphics monitors are still constructed with CRTs, other technologies are emerging that may soon replace CRT monitors. The term Flat panel displays refers to aclass of video devices that have reduced volume, weight and power requirements compared to a CRT. The important feature of flat panel display is that they are thinner than the CRTs. Since we can even write on some flat-panel displays, they will soon be available as pocket notepads. Current uses for flat-panel displays include small TV monitors, calculators, pocket video games, laptop computers, armrest viewing of movies on airlines, as advertisement boards in elevators, and as graphics displays in applications requiring rugged, portable monitors. There are 2 types: Emissive and Non Emissive displays.

Emissive Displays: They covert electrical energy into light energy. E.g. Plasma panels, Thin Film Electroluminescent displays, LEDs.

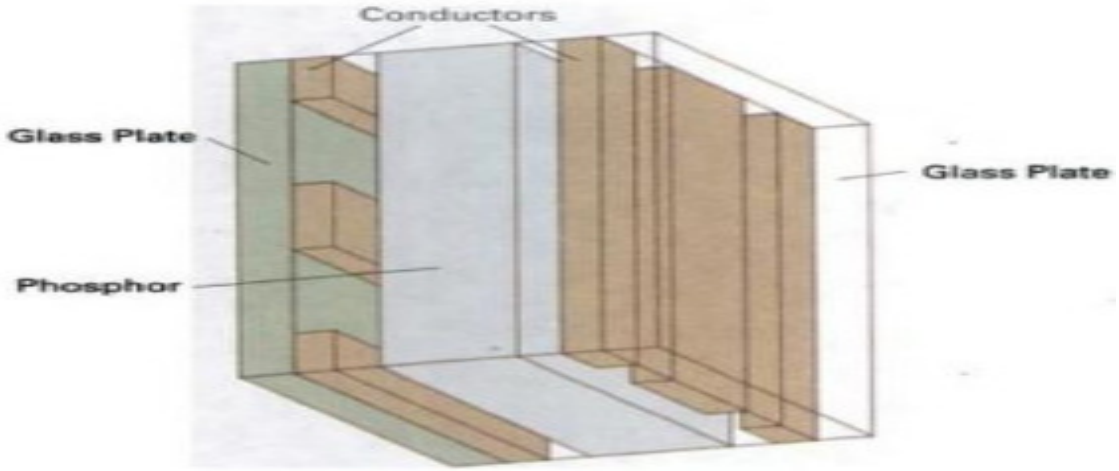
Non Emissive Displays: They use optical effects to convert sunlight or light from some other source into graphics patterns. E.g. LCD (Liquid Crystal Display)

Plasma panels, also called gas-discharge displays, are constructed by filling the region between two glass plates with a mixture of gases that usually includes neon. A series of vertical conducting ribbons is placed on one glass panel, and a set of horizontal ribbons is built into the other glass panel (Fig. below). Firing voltages applied to a pair of horizontal and vertical conductors cause the gas at the intersection of the two conductors to break down into a glowing plasma of electrons and ions. Picture definition is stored in a refresh buffer, and the firing voltages are applied to refresh the pixel positions (at the intersections of the conductors) 60 times per second. Alternating –current methods are used to provide faster application of the firing voltages, and thus brighter displays. Separation between pixels is provided by the electric field of the conductors. Figure below shows a high definition plasma panel. One disadvantage of plasma panels has been that they were strictly monochromatic devices, but systems have been developed that are now capable of displaying color and grayscale.



Basic design of a Plasma Panel display device

Thin-film electroluminescent displays are similar in construction to a plasma panel. The difference is that the region between the glass plates is filled with a phosphor, such as zinc sulfide doped with manganese,instead of a gas (see Fig. below). When a sufficiently high voltage is applied to a pair of crossing electrodes, the phosphor becomes a conductor in the area of the intersection of the two electrodes. Electrical energy is then absorbed by the manganese atoms, which then release the energy as a spot of light similar to the glowing plasma effect in a plasma panel. Electroluminescent displays require more power than plasma panels, and good color and gray scale displays are hard to achieve.

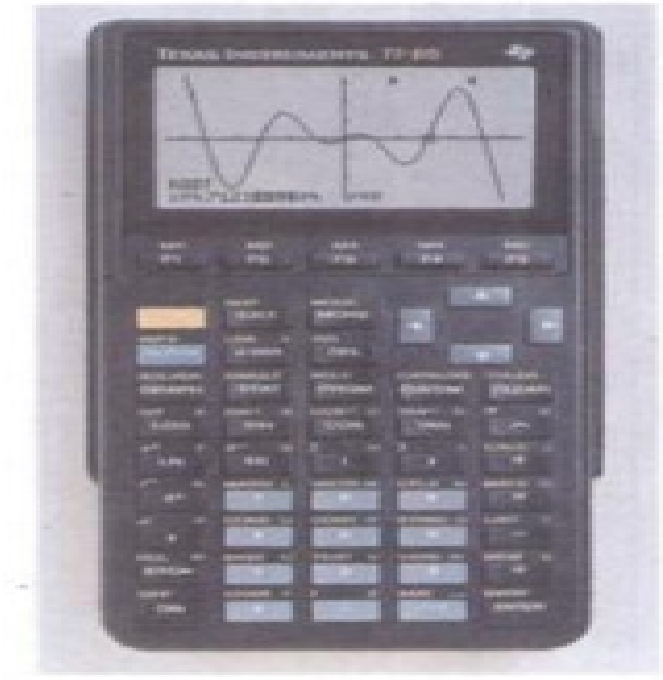


Basic design of a thin-film electroluminescent display device

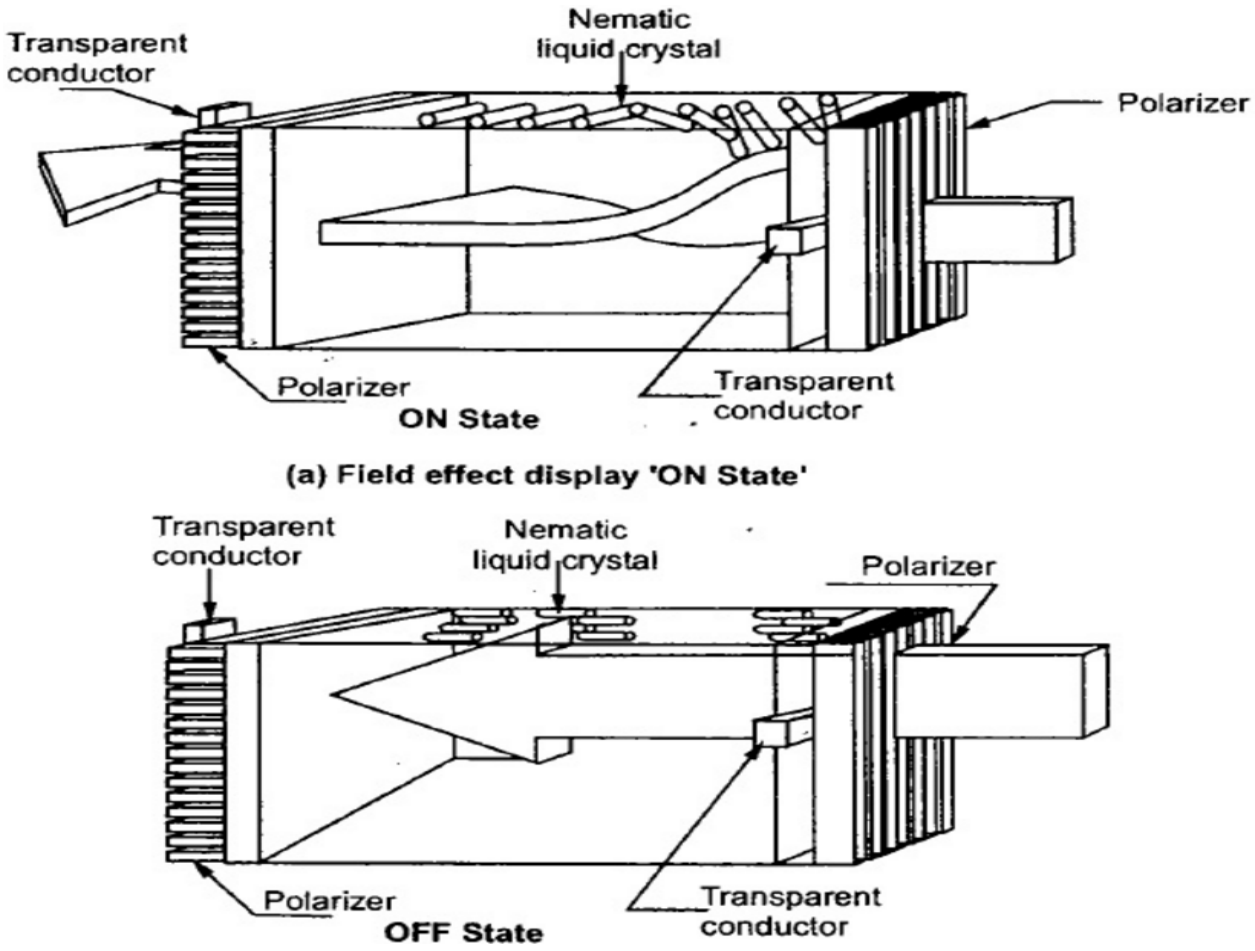
A third type of emissive device is the **Light-Emitting Diode (LED)**. A matrix of diodes is arranged to form the pixel positions in the display, and picture definition is stored in a refresh buffer. As in scan-line refreshing of a CRT, information is read from the refresh buffer and converted to voltage levels that are applied to the diodes to produce the light patterns in the display. **Liquid Crystal Display (LCD)** are commonly used in small systems, such as calculators (see figure below) and portable, laptop computers. These nonemissive devices produce a picture by passing polarized light from the surroundings or from an internal light source through a liquid-Crystal material that can be aligned to either block or transmit the light. The term liquid crystal refers to the fact that these compounds have a crystalline arrangement of molecules, yet they flow like a liquid. Flat-panel displays commonly use nematic (threadlike) liquid-crystal compounds that tend to keep the long axes of the rodshaped molecules aligned. A flat-panel display can then be constructed with a nematic liquid crystal. Two glass plates, each

containing a light polarizer at right angles to the-other plate, sandwich the liquid-crystal material. Rows of horizontal transparent conductors are built into oneglass plate, and columns of vertical conductors are put into the other plate. The intersection of two conductors defines a pixel position.

Normally, the molecules are aligned as shown in the "on state". Polarized light passing through the material is twisted so that it will pass through the opposite polarizer. The light is then reflected back to the viewer. To turn off the pixel, we apply a voltage to the two intersecting conductors to align the molecules so that the light is not .twisted. This type of flat-panel device is referred to as a passive-matrix LCD. Picture definitions are stored in a refresh buffer, and the screen is refreshed at the rate of 60 frames per second, as in the emissive devices. Back lighting is also commonly applied using solidstate electronic devices, so that the system is not completely dependent on outside light sources. Colors can be displayed by using different materials or dyes and by placing a triad of color pixels at each screen location. Another method for constructing LCDs is to place a transistor at each pixel location, using thin-film transistor technology. The transistors are used to control the voltage at pixel locations and to prevent charge from gradually leaking out of the liquid-crystal cells. These devices are called active-matrix displays.



Hand calculator with a LCD screen



The light-twisting, shutter effect used in the design of most liquid crystal display devices.

INPUT DEVICES

- Keyboards, Button Boxes, and Dials
- Mouse Devices
- Trackball, Spaceball
- Joystick
- Data Gloves
- Digitizers
- Image Scanners
- Touch Panels
- Light Pens
- Voice Systems

HARD-COPY DEVICES

- Printers
- Plotters

Line Drawing Algorithms

A line connects two points. It is a basic element in graphics. To draw a line, you need two points between which you can draw a line.

The DDA is a scan conversion line algorithm based on calculating either dy or dx. A line issampled at unit intervals in one coordinate and corresponding integer values nearest the line path are determined for other coordinates.

Considering a line with positive slope, if the slope is less than or equal to 1, we sample at unit x intervals (dx=1) and compute successive y values as

$$Y_{k+1} = Y_k + m$$

Subscript k takes integer values starting from 0, for the 1st point and increases by until end point is reached. Y value is rounded off to nearest integer to correspond to a screen pixel. For lines with slope greater than 1, we reverse the role of x and y i.e. we sample at dy=1 and calculate consecutive x values as

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

Similar calculations are carried out to determine pixel positions along a line with negative slope. Thus, if the absolute value of the slope is less than 1, we set dx=1 if $x_{start} < x_{end}$ i.e. the starting extreme point is at the left.

Digital Differential Analyzer (DDA) line drawing algorithm

Digital Differential Analyzer (DDA) algorithm is the simple line generation algorithm which is explained step by step here.

- Get the input of two end points (X0, Y0) and (X1, Y1).
- Calculate the difference between two end points.
- Based on the calculated difference, you need to identify the number of steps to put pixel. If dx >dy, then you need more steps in x coordinate; otherwise in y coordinate.
- Calculate the increment in x coordinate and y coordinate.
- Put the pixel by successfully incrementing x and y coordinates accordingly and complete the drawing of the line.

Algorithm:

```
void lineDDA (int xa, int ya, int xb, int yb)
{
  int dx = xb - xa, dy = yb - ya, steps, k;
  float xIncrement, yIncrement, x = xa, y = ya;

  if (abs (dx) > abs (dy)) steps = abs (dx);
  else steps = abs  dy);
  xIncrement = dx / (float) steps;
  yIncrement = dy / (float) steps;

  setPixel (ROUND(x), ROUND(y));
  for (k=0; k<steps; k++) {
    x += xIncrement;
    y += yIncrement;
    setPixel (ROUND(x), ROUND(y));
  }
}
```

Advantages:

- It is the simplest algorithm and it does not require special skills for implementation.
- It is a faster method for calculating pixel positions than the direct use of equation y=mx + b. It eliminates the multiplication in the equation by making use of raster characteristics, so that appropriate increments are applied in the x or y direction to find the pixel positions along the line path.

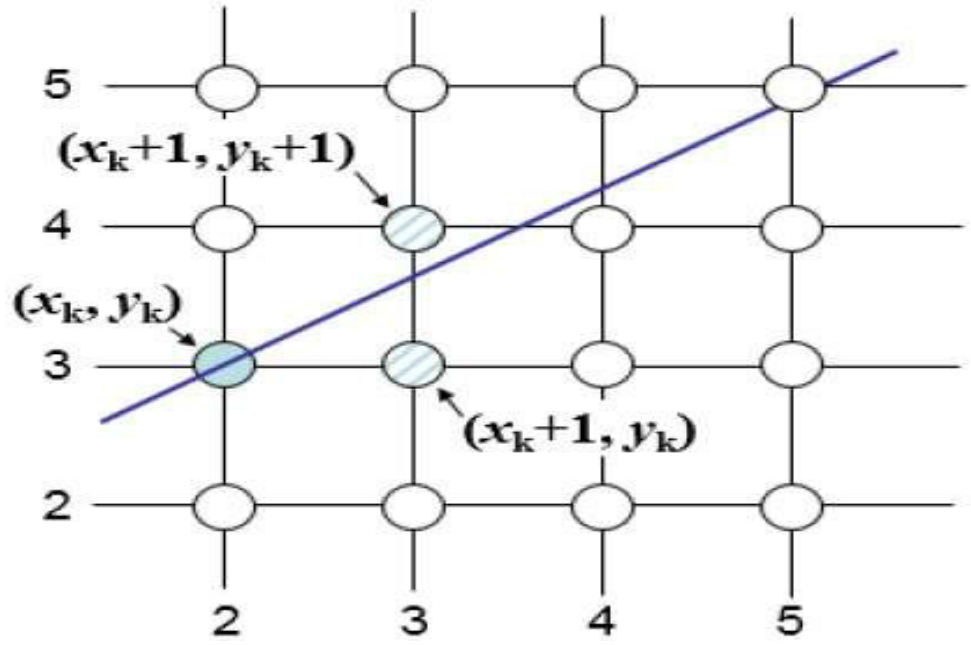
Disadvantages:

- Floating point arithmetic in DDA algorithm is still time-consuming.
- The algorithm is orientation dependent. Hence end point accuracy is poor.

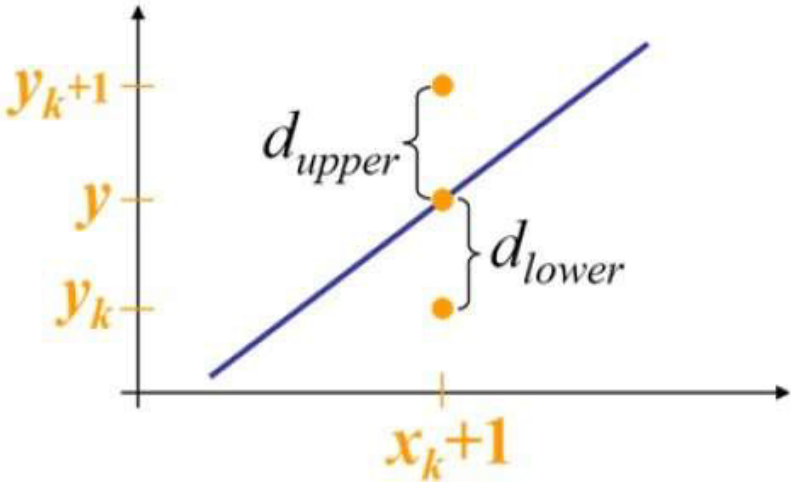
Bresenham's line drawing algorithmalgorithm

The Bresenham's algorithm is another incremental scan conversion algorithm. The big advantage of this algorithm is that, it uses only integer calculations. Moving across the x axis in unit intervals and at each step choose between two different y coordinates.

For example, as shown in the following illustration, from position (2, 3) you need to choose between (3, 3) and (3, 4). You would like the point that is closer to the original line.



At sample position Xk+1,the vertical separations from the mathematical line are labelled as $d_{upper}$and d_{lower}.



Bresenham's Line-Drawing Algorithm for $|m| < 1$

1. Input the two line endpoints and store the left endpoint in (x_0, y_0).
2. Load (x_0, y_0) into the frame buffer; that is, plot the first point.
3. Calculate constants Δx, Δy, $2\Delta y$, and $2\Delta y - 2\Delta x$, and obtain the starting value for the decision parameter as

$$p_0 = 2\Delta y - \Delta x$$

4. At each x_k along the line, starting at $k = 0$, perform the following test: If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and

$$p_{k+1} = p_k + 2\Delta y$$

Otherwise, the next point to plot is $(x_k + 1, y_k + 1)$ and

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

5. Repeat step 4 Δx times.

Algorithm

```
procedurelineBresenham(xa, ya, xb, yb : integer)
var
dx,dy,x, y, xend, p : integer;
begin
dx=abs(xa-xb);
dy=abs(ya-yb);
  p=2*dy-dx;
if(xa>xb) then
begin
 x=xb;y=yb;xend=xa;
end
else
begin
      x=xa; y=ya;xend=xb;
end;
putpixel(x,y,4);
while(x<=xend) do
begin
```

```
    x=x+1;
if (p<0) then
 p=p+2*dy;
else
begin
y=y+1;
p=p+2*(dy-dx);
end;
putpixel(x,y,4);
end
end
```

Difference between DDA Line Drawing Algorithm and Bresenhams Line Drawing Algorithm

	Digital Differential Analyzer Line Drawing Algorithm	**Bresenham's Line Drawing Algorithm**
Arithmetic	DDA algorithm uses floating points i.e. Real Arithmetic.	Bresenham's algorithm uses fixed points i.e. Integer Arithmetic
Operations	DDA algorithm uses multiplication and division in its operations.	Bresenham's algorithm uses only subtraction and addition in its operations.
Speed	DDA algorithm is rather slowly than Bresenham's algorithm in line drawing because it uses real arithmetic (floating point operations)	Bresenham's algorithm is faster than DDA algorithm in line drawing because it performs only addition and subtraction in its calculation and uses only integer arithmetic so it runs significantly faster.
Accuracy & Efficiency	DDA algorithm is not as accurate and efficient as Bresenham's algorithm	Bresenham's algorithm is more efficient and much accurate than DDA algorithm.
Drawing	DDA algorithm can draw circles and curves but that are not as accurate as Bresenham'salgorithm	Bresenham's algorithm can draw circles and curves with much more accuracy than DDA algorithm.
Round off	DDA algorithm round off the coordinates to integer that is nearest to the line	Bresenham's algorithm does not round but takes the incremental value in its operation.
Expensive	DDA algorithm uses an enormous number of floating-point multiplications so it is expensive	Bresenham's algorithm is less expensive than DDA algorithm as it uses only addition and subtraction.

CIRCLE DRAWING ALGORITHM

(Midpoint Circle drawing algorithm)

A Circle is defined as the set of points that are all at a given distance r from a center position (x_c,y_c).

The distance relationship is expressed by Pythagorean theorem in Cartesian coordinates as ,

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

Therefore,

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

In parametric polar form,

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

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

The Circle function is given by,

$$f_{circle}(x,y)= x^2+y^2-r^2 \text{ and}$$

$$f_{circle}(x, y) \begin{cases} < 0, & \text{if } (x, y) \text{ is inside the circle boundary} \\ = 0, & \text{if } (x, y) \text{ is on the circle boundary} \\ > 0, & \text{if } (x, y) \text{ is outside the circle boundary} \end{cases}$$

The Midpoint Circle algorithm is as follows:

```
void circleMidpoint (int xCenter, int yCenter, int radius)
{
  int x = 0;
  int y = radius;
  int p = 1 - radius;
  void circlePlotPoints (int, int, int, int);

  /* Plot first set of points */
  circlePlotPoints (xCenter, yCenter, x, y);

  while (x < y) {
    x++;
    if (p < 0)
      p += 2 * x + 1;
    else {
      y--;
      p += 2 * (x - y) + 1;
    }
    circlePlotPoints (xCenter, yCenter, x, y);
  }
}

void circlePlotPoints (int xCenter, int yCenter, int x, int y)
{
  setPixel (xCenter + x, yCenter + y);
  setPixel (xCenter - x, yCenter + y);
  setPixel (xCenter + x, yCenter - y);
  setPixel (xCenter - x, yCenter - y);
  setPixel (xCenter + y, yCenter + x);
  setPixel (xCenter - y, yCenter + x);
  setPixel (xCenter + y, yCenter - x);
  setPixel (xCenter - y, yCenter - x);
}
```

Attributes of Output Primitives

Any parameter that affects the way a primitive is to be displayed is referred to as an attribute parameter. Example attribute parameters are color, size etc. A line drawing function for example could contain parameter to set color, width and other properties.

1. Line Attributes
2. Curve Attributes
3. Color and Grayscale Levels
4. Area Fill Attributes
5. Character Attributes
6. Bundled Attributes

Line Attributes

Basic attributes of a straight line segment are its type, its width, and its color. In some graphics packages, lines can also be displayed using selected pen or brush options

* Line Type
* Line Width
* Pen and Brush Options
* Line Color

Line type

Possible selection of line type attribute includes solid lines, dashed lines and dotted lines. To set line type attributes in a **PHIGS** application 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 lines that are solid, dashed, dash dotted respectively. Other values for line type parameter it could be used to display variations in dot-dash patterns.

Line width

Implementation of line width option depends on the capabilities of the output device to set the line width attributes.

setLinewidthScaleFactor (lw)

Line width parameter lw is assigned a positive number to indicate the relative width of line to be displayed. A value of 1 specifies a standard width line. A user could set lw to a 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.

Line Cap

We can adjust the shape of the **line** ends to give them a better appearance by adding line caps.

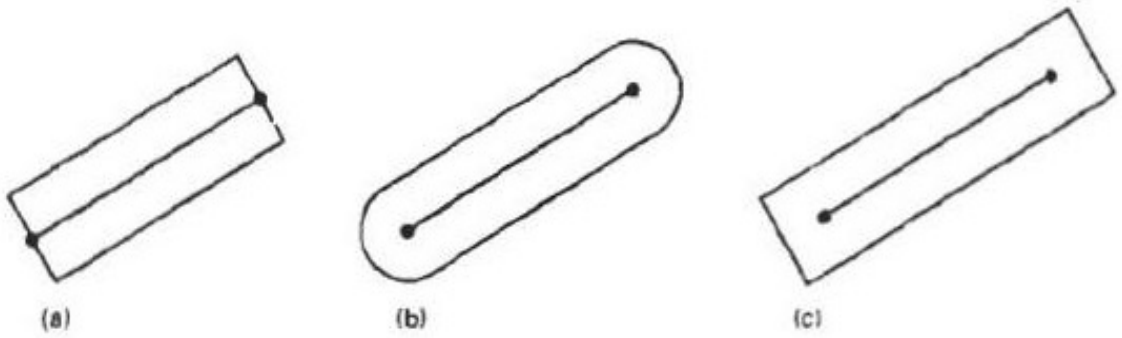
There are three types of line cap. They are

* Butt cap
* Round cap
* Projecting square 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.

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

Projecting square cap extend the line and add butt caps that are positioned one-half of the line width beyond the specified endpoints.



Thick lines drawn with (a) butt caps, (b) round caps, and (c) projecting square caps.

Three possible methods for smoothly joining two line segments

* Mitter Join
* Round Join
* Bevel Join

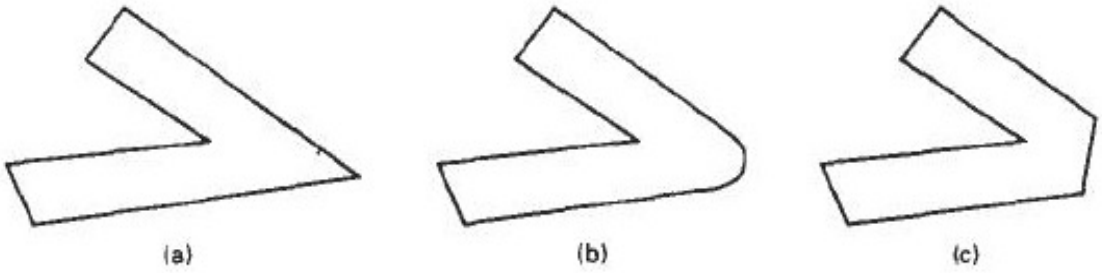
A miter join accomplished by extending the outer boundaries of each of the two lines

until they meet.

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

A bevel join is generated by displaying the line segment with but caps and filling in tri

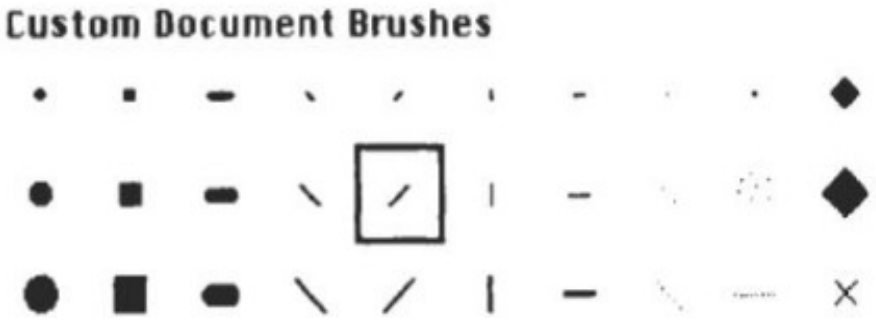
angular gap where the segments meet.



Thick line segments connected with (a) miter join, (b) round join, and (c) bevel join.

Pen and Brush Options

With some packages, lines can be displayed with pen or brush selections. Options in this category include shape, size, and pattern. Some possible pen or brush shapes are given in Figure



Line color

A poly line 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 the set pixel procedure.

We set the line color value in PHIGS with the function

setPolylineColourIndex (lc)

Nonnegative integer values, corresponding to allowed color choices, are assigned to the line color parameter lc

Example: Various line attribute commands in an applications program is given by the following sequence of statements

```
setLinetype(2);
setLinewidthScaleFactor(2);
setPolylineColourIndex (5);
polyline(n1, wc points1);
setPolylineColorIindex(6);
poly line (n2, wc points2);
```

This program segment would display two figures, drawn with double-wide dashed lines. The first is displayed in a color corresponding to code 5, and the second in color 6.

Curve attributes

Parameters for curve attribute are same as those for line segments. Curves displayed with varying colors, widths, dot – dash patterns and available pen or brush options

Color and Grayscale Levels

Various color and intensity-level options can be made available to a user, depending on the capabilities and design objectives of a particular system

In a color raster system, the number of color choices available depends on the amount of storage provided per pixel in the frame buffer

Color-information can be stored in the frame buffer in two ways:

* 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

With the direct storage scheme, whenever 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.

A minimum number of colors can be provided in this scheme with 3 bits of storage per pixel, as shown in Table

3 bits - 8 choice of color

6 bits – 64 choice of color

8 bits – 256 choice of color

A user can set color-table entries in a PHIGS applications program with the function

setColourRepresentation (ws, ci, colorptr)

Parameter **ws** identifies the workstation output device; parameter ci specifies the color index, which is the color-table position number (**0** to **255**) and parameter colorptr points to a trio of RGB color values (**r, g, b**) each specified in the range from **0** to 1

Grayscale

With monitors that have no color capability, color functions can be used in an application program to set the shades of gray, or grayscale, for displayed primitives. Numeric values over the range from 0 to 1 can be used to specify grayscale levels, which are then converted to appropriate binary codes for storage in the raster.

INTENSITY CODES FOR A FOUR-LEVEL GRAYSCALE SYSTEM

Intensity Codes	*Stored Intensity Values In The Frame Buffer (Binary Code)*		*Displayed Grayscale*
0.0	0	(00)	Black
0.33	1	(01)	Dark gray
0.67	2	(10)	Light gray
1.0	3	(11)	White

Intensity = 0.5[min(r, g, b) + max(r, g, b)]

Area fill Attributes

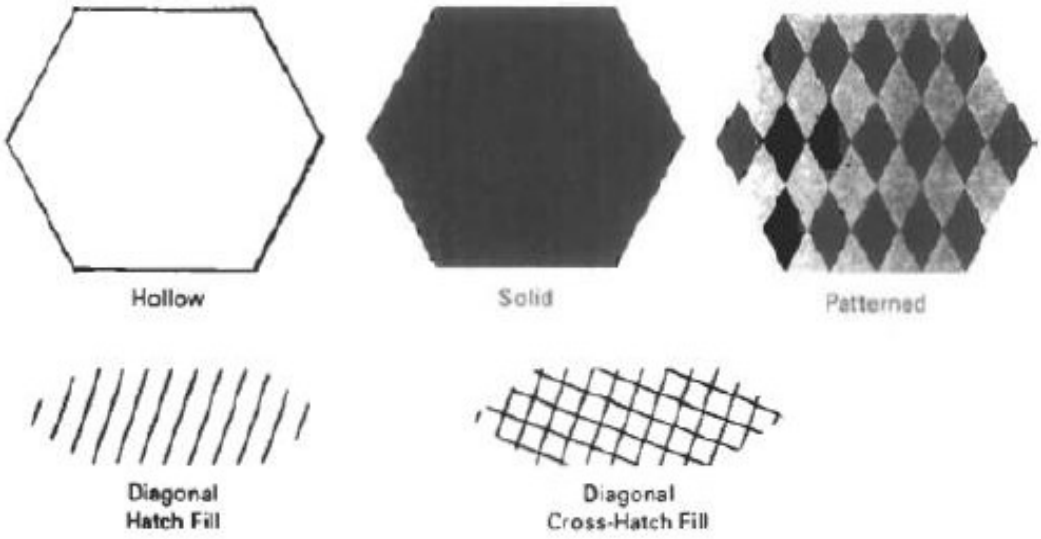
Options for filling a defined region include a choice between a solid color or a pattern fill and choices for particular colors and patterns

Fill Styles

Areas are displayed with three basic fill styles: hollow with a color border, filled with a solid color, or filled with a specified pattern or design. A basic fill style is selected in a PHIGS program with the function

setInteriorStyle (fs)

Values for the fill-style parameter fs include hollow, solid, and pattern. Another value for fill style is hatch, which is used to fill an area with selected hatching patterns-parallel lines or crossed lines



The color for a solid interior or for a hollow area outline is chosen with where fill color parameter fc is set to the desired color code

setInteriorColourIndex (fc)

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);

Fill area (n, points)

Index (pi)	*Pattern (cp)*
1	$\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$
2	$\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}$

Character Attributes

The appearance of displayed character is controlled by attributes such as font, size, color and orientation. Attributes can be set both for entire character strings (text) and for individual characters defined as marker symbols

Text Attributes

The choice of font or type face is set of characters with a particular design style as courier, Helvetica, times roman, and various symbol groups.

The characters in a selected font also be displayed with styles. (solid, dotted, double) in **bold face** in **italics**, and in **outline** or shadow styles.

A particular font and associated stvle is selected in a PHIGS program by setting an

integer code for the text font parameter tf in the function

setTextFont (tf)

Control of text color (or intensity) is managed from an application program with

setTextColourIndex (tc)

Where text color parameter tc specifies an allowable color code.

Text size can be adjusted without changing the width to height ratio of characters with

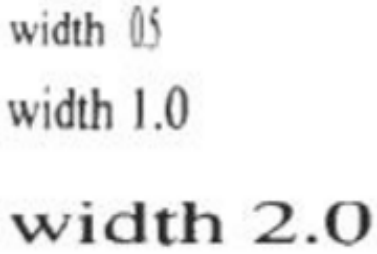
setCharacterHeight (ch)

Parameter ch is assigned a real value greater than 0 to set the coordinate height of capital letters

The width only of text can be set with function.

setCharacterExpansionFactor (cw)

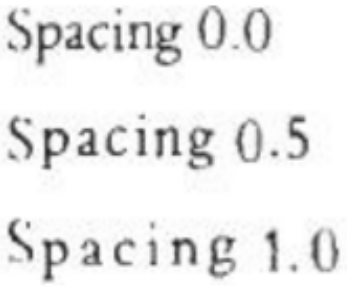
Where the character width parameter cw is set to a positive real value that scales the body width of character



Spacing between characters is controlled separately with

setCharacterSpacing (cs)

Where the character-spacing parameter cs can he assigned any real value

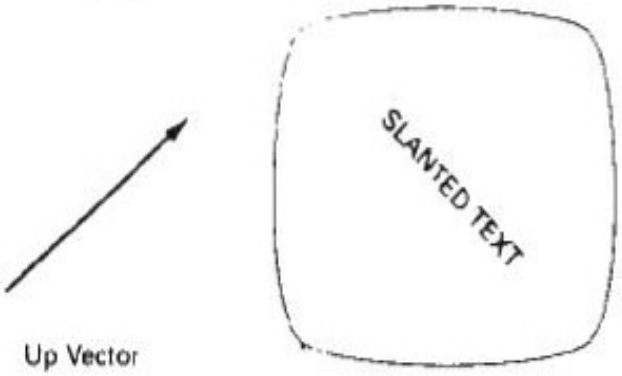


The orientation for a displayed character string is set according to the direction of the character up vector

setCharacterUpVector (upvect)

Parameter upvect in this function is assigned two values that specify the x and y

vector components. For example, with upvect = (1, 1), the direction of the up vector is 45o and text would be displayed as shown in Figure.



To arrange character strings vertically or horizontally

setTextPath (tp)

can be assigned the value: right, left, up, or down

Another handy attribute for character strings is alignment. This attribute specifies how text is to be positioned with respect to the $tart coordinates. Alignment attributes are set with

setTextAlignment (h,v)

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

A precision specification for text display is given with

setTextPrecision (tpr)

tpr is assigned one of values string, char or stroke.

Marker Attributes

A marker symbol is a single character that can he displayed in different colors and in different sizes. Marker attributes are implemented by procedures that load the chosen character into the raster at the defined positions with the specified color and size. We select a particular character to be the marker symbol with

setMarkerType (mt)

where marker type parameter mt is set to an integer code. Typical codes for marker type are the integers 1 through 5, specifying, respectively, a dot (.) a vertical cross (+), an asterisk (*), a circle (o), and a diagonal cross (X).

We set the marker size with

setMarkerSizeScaleFactor (ms)

with parameter marker size ms assigned a positive number. This scaling parameter is applied to the nominal size for the particular marker symbol chosen. Values greater than 1 produce character enlargement; values less than 1 reduce the marker size.

Marker color is specified with

setPolymarkerColourIndex (mc)

A selected color code parameter mc is stored in the current attribute list and used to display subsequently specified marker primitives

Bundled Attributes

The procedures considered so far each function reference a single attribute that specifies exactly how a primitive is to be displayed these specifications are called individual attributes.

A particular set of attributes values for a primitive on each output device is chosen by specifying appropriate table index. Attributes specified in this manner are called bundled attributes. The choice between a bundled or an unbundled specification is made by setting a switch called the aspect source flag for each of

these attributes

setIndividualASF(attributeptr, flagptr)

where parameter attributer ptr points to a list of attributes and parameter flagptr points to the corresponding list of aspect source flags. Each aspect source flag can be assigned a value of individual or bundled.

Bundled line Attributes

Entries in the bundle table for line attributes on a specified workstation are set with the function

setPolylineRepresentation (ws, li, lt, lw, lc)

Parameter ws is the workstation identifier and line index parameter li defines the bundle table position. Parameter lt, lw, tc are then bundled and assigned values to set the line type, line width, and line color specifications for designated table index.

Example

setPolylineRepresentation (1, 3, 2, 0.5, 1)

setPolylineRepresentation (4, 3, 1, 1, 7)

A poly line that is assigned a table index value of 3 would be displayed using dashed lines at half thickness in a blue color on work station 1; while on workstation 4, this same index generates solid, standard-sized white lines

Bundle area fill Attributes

Table entries for bundled area-fill attributes are set with

setInteriorRepresentation (ws, fi, fs, pi, fc)

Which defines the attributes list corresponding to fill index fi on workstation ws. Parameter fs, pi and fc are assigned values for the fill style pattern index and fill color.

Bundled Text Attributes

setTextRepresentation (ws, ti, tf, tp, te, ts, tc)

Bundles values for text font, precision expansion factor size an color in a table position for work station ws that is specified by value assigned to text index parameter ti.

Bundled marker Attributes

setPolymarkerRepresentation (ws, mi, mt, ms, mc)

That defines marker type marker scale factor marker color for index mi on workstation ws.

Inquiry functions

Current settings for attributes and other parameters as workstations types and status in the system lists can be retrieved with inquiry functions.

inquirePolylineIndex (lastli)

and

inquireInteriorcColourIndex (lastfc)

Copy the current values for line index and fill color into parameter lastli and lastfc.

SUMMARY OF ATTRIBUTES

Output Primitive Type	*Associated Attributes*	*Attribute-Setting Functions*	*Bundled-Attribute Functions*
Line	Type	setLinetype	setPolylineIndex
	Width	setLineWidthScaleFactor	setPolylineRepresentation
	Color	setPolylineColourIndex	
Fill Area	Fill Style	setInteriorStyle	setInteriorIndex
	Fill Color	setInteriorColorIndex	setInteriorRepresentation
	Pattern	setInteriorStyleIndex	
		setPatternRepresentation	
		setPatternSize	
		setPatternReferencePoint	
Text	Font	setTextFont	setTextIndex
	Color	setTextColourIndex	setTextRepresentation
	Size	setCharacterHeight	
		setCharacterExpansionFactor	
	Orientation	setCharacterUpVector	
		setTextPath	
		setTextAlignment	
Marker	Type	setMarkerType	setPolymarkerIndex
	Size	setMarkerSizeScaleFactor	setPolymarkerRepresentation
	Color	setPolymarkerColourIndex	

Antialiasing

Antialiasing is the smoothing of the image or sound roughness caused by aliasing . With images, approaches include adjusting pixel positions or setting pixel intensities so that there is a more gradual transition between the color of a line and the background color. With sound, aliases are removed by eliminating frequencies above half the sampling frequencies.

Side effects of Scan Conversion

The most common side effects when working with raster devices are:

- Unequal Intensity
- Overstrike
- Aliasing

Unequal Intensity

- Human perception of light is dependent on density and intensity of light source.Thus a raster display with perfect squareness, a diagonal line of pixels will appear dimmer that a horizontal or vertical line.

Solution: By increasing the number of pixels on diagonal lines.

Overstrike:

- The same pixel is written more than once.
- This results in intensified pixels in case of photographic media, such as slide or transparency.

Solution: Check each pixel to see whether it has already been written to prior to writing a new point.

Aliasing

- The effect created when rasterization is performed over a discrete series of pixels.
- In particular, when lines or edges do not necessarily align directly with arrow or column of pixels, that line may appear unsmooth and have a stair-step edge appearance.
- Jagged appearance of curves or diagonal lines on a display screen, which is caused by low screen resolution.
- Refers to the plotting of a point in a location other than its true location in order to fit the point into the raster.

- Consider equation Y=mX+b, For m=0.5, b=1, X=3 ------ Y would be 2.5. So the point (3, 2.5) is plotted at alias location (3,3).

Anti-Aliasing

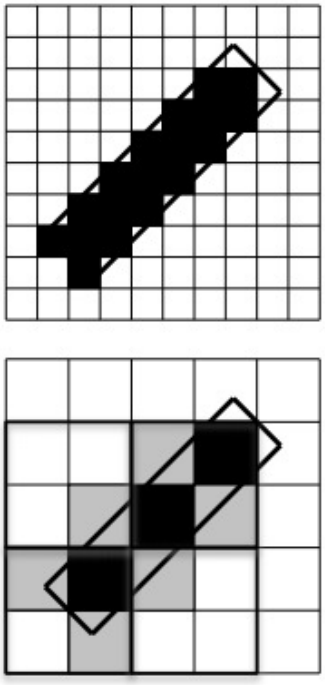
The image on the right shows the result of anti-aliasing through the use of higher resolution.

Anti-Aliasing

- Antialiasing utilizes blending techniques to blur the edges of the lines and provide the viewer with the illusion of a smoother line.
- Two general approaches:
- **Super-sampling**
  - samples at higher resolution, then filters down the resulting image
  - Sometimes called post-filtering
  - The prevalent form of anti-aliasing in hardware
- **Area sampling**
  - sample primitives with a box (or Gaussian, or whatever) rather than spikes
  - Requires primitives that have area (lines with width)
  - Sometimes referred to as pre-filtering

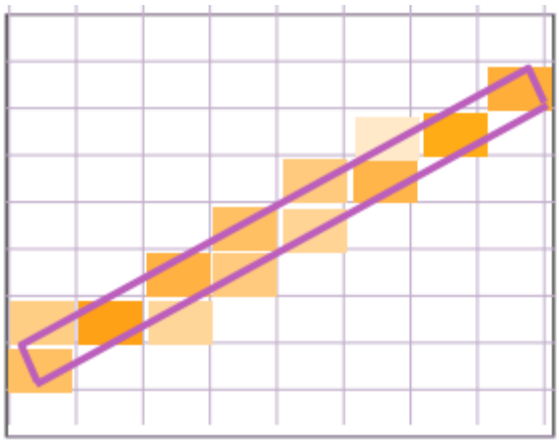
Super-sampling

- Sample at a higher resolution than required for display, and filter image down
- 4 to 16 samples per pixel is typical
- Samples might be on a uniform grid, or randomly positioned, or other variants
- Divide each pixel into sub-pixels.
- The number of intensities are the max number of sub-pixels selected on the line segment within a pixel.
- The intensity level for each pixel is proportional to the number of sub-pixels inside the polygon representing the line area.
- Line intensity is distributed over more pixels.



Area Sampling

- Determine the percentage of area coverage for a screen pixel, then set the pixel intensity proportional to this percentage.
- Consider a line as having thickness.
- Consider pixels as little squares.
- Unweighted area sampling: Fill pixels according to the proportion of their square covered by the line.
- Weighted area sampling: Weight the contribution according to where in the square the primitives falls



Area Sampling

UnweightedArea Sampling

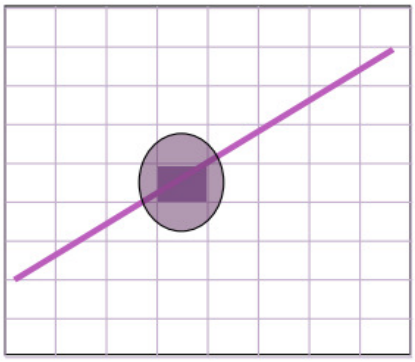
- Primitive cannot affect intensity of pixel if it does not intersect the pixel.
- Equal areas cause equal intensity, regardless of distance from pixel center to area.
- Unweighted sampling colors two pixels identically when the primitive cuts the same area through the two pixels.
- Intuitively, pixel cut through the center should be more heavily weighted than one cut along corner.

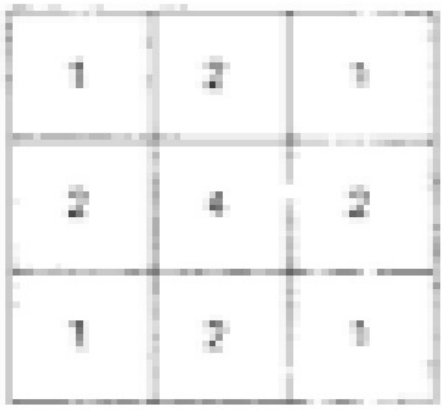
0	0	0	1/8	0
0	0	1/4	.914	1/8
0	1/4	.914	1/4	0
1/8	.914	1/4	0	0
0	1/8	0	0	0

Unweighted sampling

Weighted Area Sampling

- weight the subpixel contributions according to position, giving higher weights to the central subpixels.
- weighting function, W(x,y)
  - specifies the contribution of primitive passing through the point (x, y) from pixel center

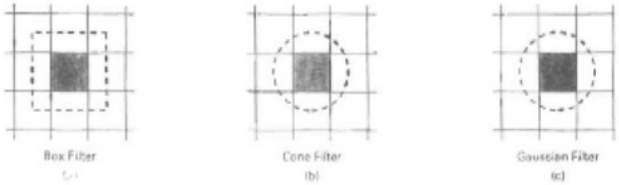




Filtering Techniques

- A continuous weighting surface (or filter function) covering the pixel.
- Applying the filter function by integrating over the pixel surface to obtain the weighted average intensity
- Weighted (Filter) Function
  - Determines the influence on the intensity of a pixel of a given small area dA of a primitive.
  - This function is constant for unweighted and decreases with increasing distance for weighted.
  - Total intensity is the integral of the weighting (filter) function over the area of overlap.
  - Ws is the volume (always between 0 and 1)
  - I=Imax •Ws

- Box, Cone and Gaussiean



http://www.graphics.cornell.edu/online/tutorial/

https://www.cs.utexas.edu/~fussell/courses/cs324e2003/hear50265_ch03.pdf

http://www.slideshare.net/rajeshkamboj/computer-graphics-notes-btech-kuk-mdu

http://www.slideshare.net/rhspcte/computer-graphics-ebookhearn-baker