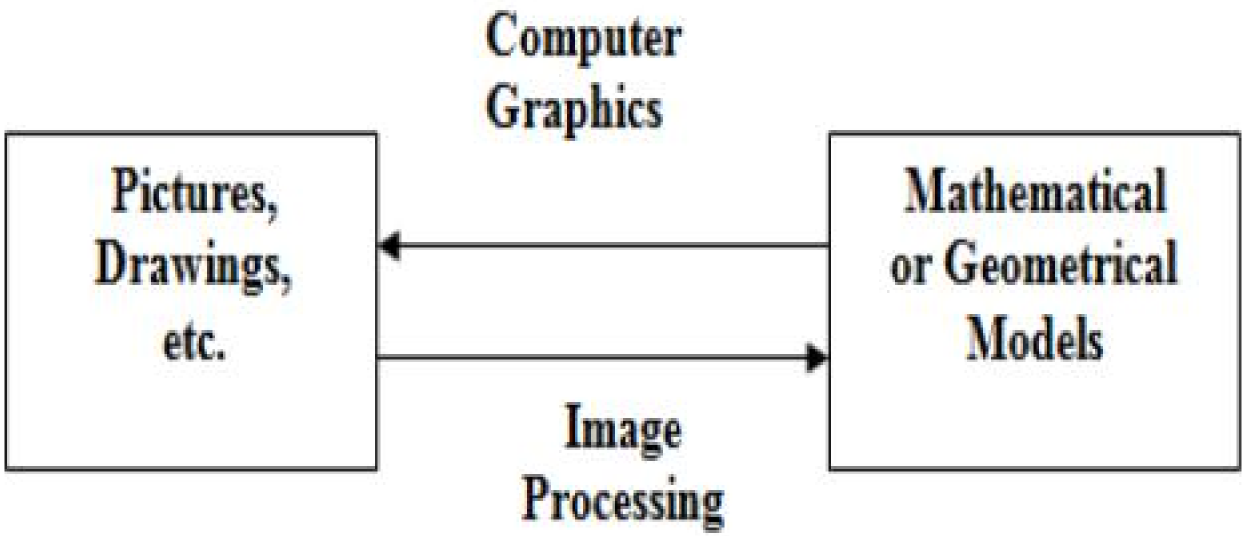
BCA
Fifth Semester
“ Computer Graphics“

Unit I
Introduction to Computer Graphics

- Computer graphics is the field of study which deals with pictures or images.

- Computer graphics is the pictorial representation and manipulation of data with the help of computer hardware and software.

Unit I



- *Computer Graphics*: Synthesize pictures from mathematical or geometrical models.
- *Image Processing*: analyze pictures to derive descriptions (often in mathematical or geometrical forms) of objects appeared in the pictures.

Application Areas of Computer Graphics

- Computer graphics is used today in many different areas of science, engineering, industry, business, education, entertainment, medicine, art and training, All of these are included in the following categories.

1) User interfaces

Most applications have user interfaces that rely on desktop windows systems to manage multiple simultaneous activities , and on point-and click facilities to allow users to select menu items, icons and objects on the screen. These activities fall under computer graphics. Typing is necessary only to input text to be stored and manipulated. For example, Word processing, spreadsheet, and desktop-publishing programs are the typical examples where user-interface techniques are implemented.

Application Areas of Computer Graphics

2) Plotting

Plotting 2D and 3D graphs of mathematical, physical, and economic functions use computer graphics extensively. The histograms, bar, and pie charts; the task-scheduling charts are the most commonly used plotting. These all are used to present meaningfully and concisely the trends and patterns of complex data.

3) Office automation and electronic publishing

Computer graphics has facilitated the office automation and electronic publishing which is also popularly known as desktop publishing, giving more power to the organizations to print the meaningful materials in-house. Office automation and electronic publishing can produce both traditional printed (Hardcopy) documents and electronic(softcopy) documents that contain text, tables, graphs, and other forms of drawn or scanned-in graphics.

Application Areas of Computer Graphics

4) Computer Aided Drafting and Design

One of the major uses of computer graphics is to design components and systems of mechanical, electrical, electrochemical, and electronic devices, including structures such as buildings, automobile bodies, airplane and ship hulls, very large scale integrated (VLSI) chips, optical systems,. and telephone and computer networks. These designs are more frequently used to test the structural, electrical, and thermal properties of the systems.

5) Scientific and business Visualization

Generating computer graphics for scientific, engineering, and medical data sets is termed as scientific visualization whereas business visualization is related with the non scientific data sets such as those obtained in economics. Visualization makes easier to understand the trends and patterns inherent in the huge amount of data sets. It would , otherwise , be almost impossible to analyze those data numerically.

Application Areas of Computer Graphics

6) Simulation and modeling

Simulation helps to learn or to feel the conditions one might have to face in near future without being in danger at the beginning of the course. For example, astronauts can exercise the feeling of weightlessness in a simulator; similarly a pilot training can be conducted in flight simulator. The military tank simulator, the naval simulator, driving simulator, air traffic control simulator, heavy-duty vehicle simulator, and so on are some of the mostly used simulator in practice. Simulators are also used to optimize the system, for example the vehicle, observing the reactions of the driver during the operation of the simulator.

7) Entertainment

Disney movies such as Lion Kings and The Beauty of Beast, and other scientific movies like Jurassic Park, The lost world etc are the best example of the application of computer graphics in the field of entertainment. Instead of drawing all necessary frames with slightly changing scenes for the production of cartoon-film, only the key frames are sufficient for such cartoon-film where the in between frames are interpolated by the graphics system dramatically decreasing the cost of production while maintaining the quality. Computer and video games such FIFA, Doom ,Pools are few to name where graphics is used extensively.

Unit I
Application Areas of Computer Graphics

8) Art and commerce

Here computer graphics is used to produce pictures that express a message and attract attention such as a new model of a car moving along the ring of the Saturn . These pictures are frequently seen at transportation terminals supermarkets , hotels etc. The slide production for commercial , scientific, or educational presentations is another cost effective use of computer graphics. One of such graphics packages is a PowerPoint.

9) Cartography

Cartography is a subject , which deals with the making of maps and charts. Computer graphics is used to produce both accurate and schematic representations of geographical and other natural phenomena from measurement data. Examples include geographic maps , oceanographic charts, weather maps, contour maps and population-density maps. Surfer is one of such graphics packages , which is extensively used for cartography.

Advantages of Computer Graphics

1) High quality graphics provides the best way to communicate with computer.
2) It is possible to produce animation
3) Can be used to control animation such as speed, total scene in view etc.
4) Provides facility of update dynamic which can be used to change shape, color and other properties of object in view.
5) Used to present data or information in the form of bar diagram, pie chart etc..which makes visualization better.

Display Technologies

1) Cathode Ray Tube (CRT)

- CRT are the most common display devices on computer today. A CRT is an evacuated glass tube, with a heating element on one end and a phosphor-coated screen on the other end.
- When a current flows through this heating element (filament) the conductivity of metal is reduced due to high temperature. These cause electrons to pile up on the filament.
- These electrons are attracted to a strong positive charge from the outer surface of the focusing anode cylinder.
- Due to the weaker negative charge inside the cylinder, the electrons head towards the anode forced into a beam and accelerated by the inner cylinder walls in just the way that water is speeds up when its flow though a small diameter pipe.
- The forwarding fast electron beam is called Cathode Ray. A cathode ray tube is shown in figure below.

Display Technologies

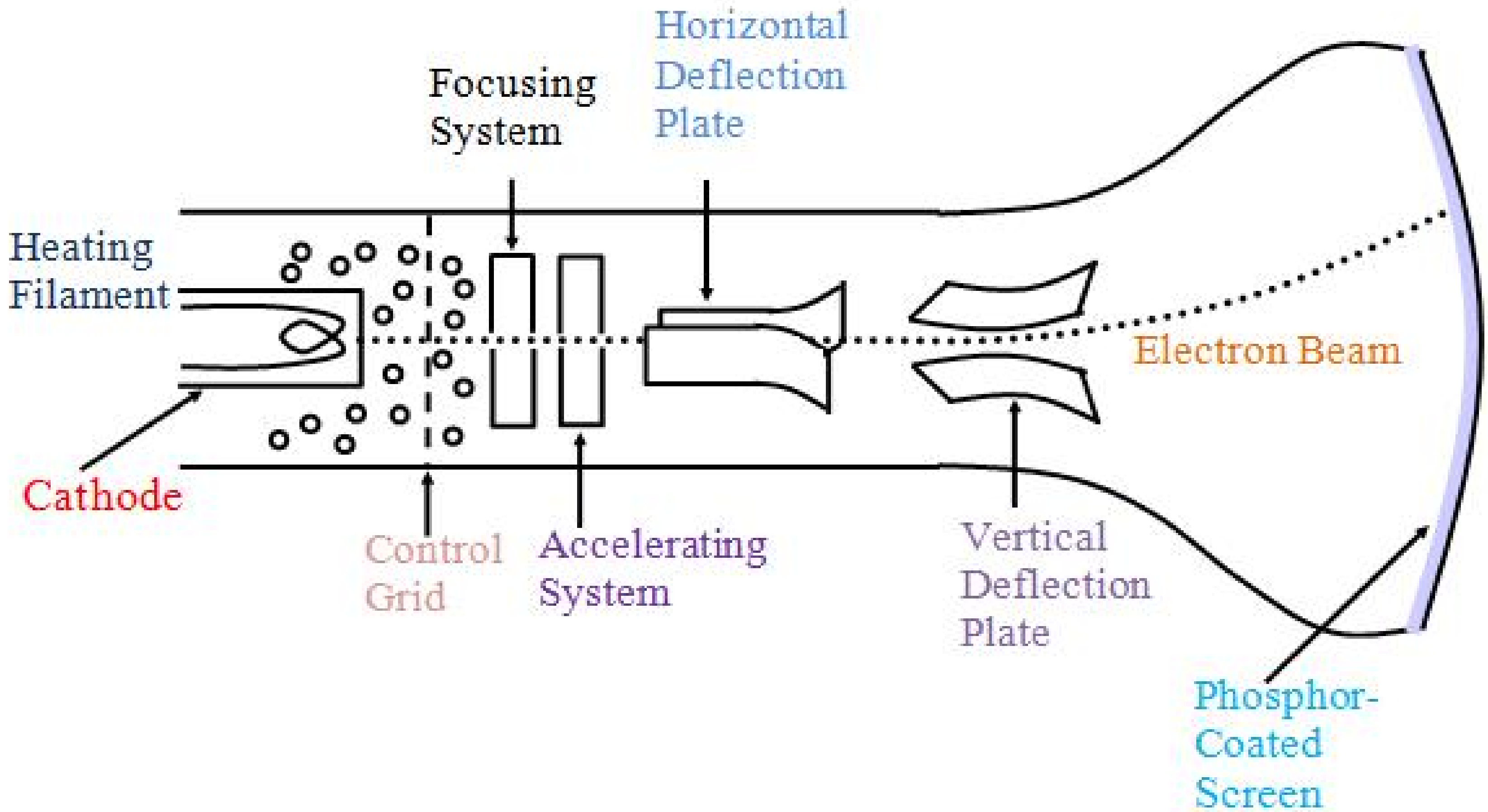


Figure: Cathode Ray Tube

Display Technologies

- There are two sets of weakly charged deflection plates with oppositely charged, one positive and another negative. The first set displaces the beam up and down and the second displaces the beam left and right.
- The electrons are sent flying out of the neck of bottle (tube) until the smash into the phosphor coating on the other end.
- When electrons strike on phosphor coating, the phosphor then emits a small spot of light at each position contacted by electron beam. The glowing positions are used to represent the picture in the screen.
- The amount of light emitted by the phosphor coating depends on the no of electrons striking the screen. The brightness of the display is controlled by varying the voltage on the control grid.

Display Technologies

<u>Persistence</u>:

-How long a phosphor continues to emit light after the electron beam is removed.

-Persistence of phosphor is defined as the time it takes for emitted light to decay to 1/10 (10%) of its original intensity. Range of persistence of different phosphors can react many seconds.

-Phosphors for graphical display have persistence of 10 to 60 microseconds. Phosphors with low persistence are useful for animation whereas high persistence phosphor is useful for highly complex, static pictures.

<u>Refresh Rate</u>:

Light emitted by phosphor fades very rapidly, so to keep the drawn picture glowing constantly, it is required to redraw the picture repeatedly and quickly directing the electron beam back over the some point. The no of times/sec the image is redrawn to give a feeling of non-flickering pictures is called refresh-rate. If Refresh rate decreases, flicker develops. Refresh rate above which flickering stops and steady it may be called as critical fusion frequency(CFF).

Display Technologies

<u>Resolution</u>:

Maximum number of points displayed horizontally and vertically without overlap on a display screen is called resolution. In other ways , resolution is referred as the no of points per inch(dpi/pixel per inch).

Color CRT

In color CRT, the phosphor on the face of CRT screen are laid(put) in to different fashion. Depending on the technology of CRT there are two methods for displaying the color pictures into the screen.

1. Beam penetration method
2. Shadow mask method

Beam Penetration method:

This method is commonly used for random scan display or vector display. In random scan display CRT, the two layers of phosphor usually red and green are coated on CRT screen. Display color depends upon how far electrons beam penetrate the phosphor layers.

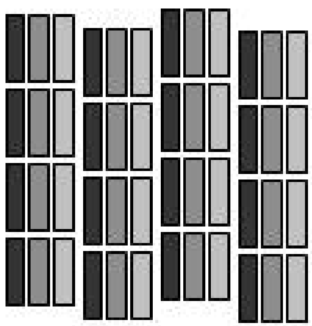
Slow electron excite only red layer so that we can see red color displayed on the screen pixel where the beam strikes. Fast electron beam excite green layer penetrating the red layer and we can see the green color displayed at the corresponding position. Intermediate is combination of red and green so two additional colors are possible – orange and yellow.

So only four colors are possible so no good quality picture in this type of display method.

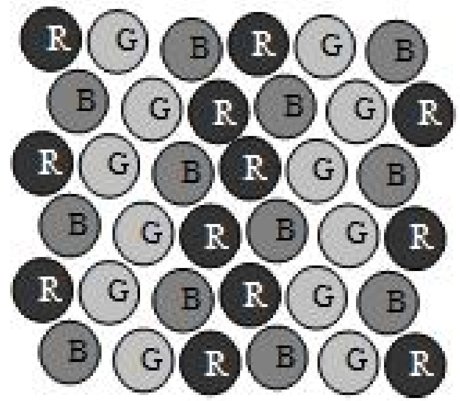
Shadow Mask Method:

Shadow mask method is used for raster scan system so they can produce wide range of colors. In shadow mask color CRT, the phosphor on the face of the screen are laid out in a precise geometric pattern. There are two primary variations.

1. The stripe pattern of inline tube
2. The delta pattern of delta tube



Stripe pattern



Delta Pattern

- In color CRT, the neck of tube, there are three electron guns, one for each red, green and blue colors. In phosphor coating there may be either strips one for each primary color, for a single pixel or there may be three dots one for each pixel in delta fashion.
- Special metal plate called a shadow mask is placed just behind the phosphor coating to cover front face.
- The mask is aligned so that it simultaneously allow each electron beam to see only the phosphor of its assigned color and block the phosphor of other two color.

1. Delta-Delta CRT:

- In delta-delta CRT, three electron beams one for each R,G,B colors are deflected and focused as a group onto shadow mask, which contains a series of holes aligned with the phosphor dots.
-

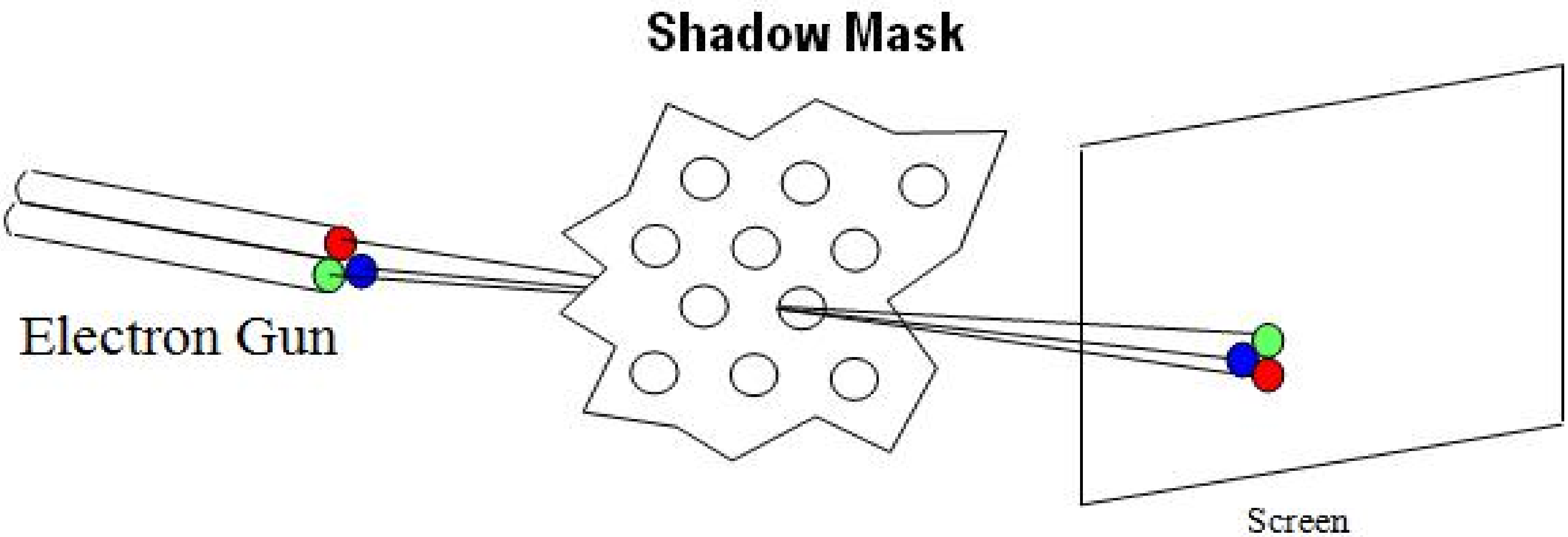


Figure: Shadow mask in Delta-Delta CRT

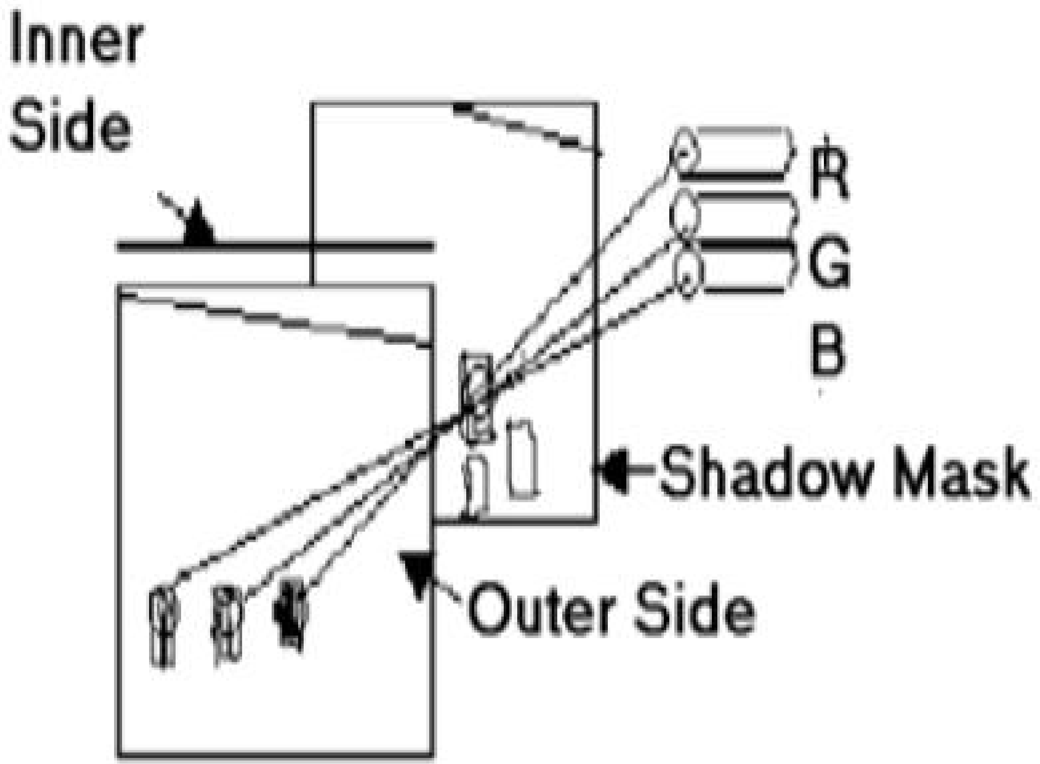
- Inner side of viewing has several groups of closely spaced red ,green and blue phosphor dot called triad in delta fashion.
- Thin metal plate adjusted with many holes near to inner surface called shadow mask which is mounted in such a way that each hole aligned with respective triad.

- Triad are so small that is perceived as a mixture of colors. When three beams pass through a hole in shadow mask, they activate the dot triangle to illuminate an small spot colored on the screen.
- The color variation in shadow mask CRT can be obtained by varying the intensity level of the three electron guns.

The main draw back of this CRT is due to difficulty for the alignment of shadow mask hole and respective triads.

A precision inline CRT:

This CRT uses strips pattern instead of delta pattern. Three strips one for each R, G, B color are used for a single pixel along a scan line so called inline. This eliminates the drawbacks of delta-delta CRT at the cost of slight reduction of image sharpness at the edge of the tube.

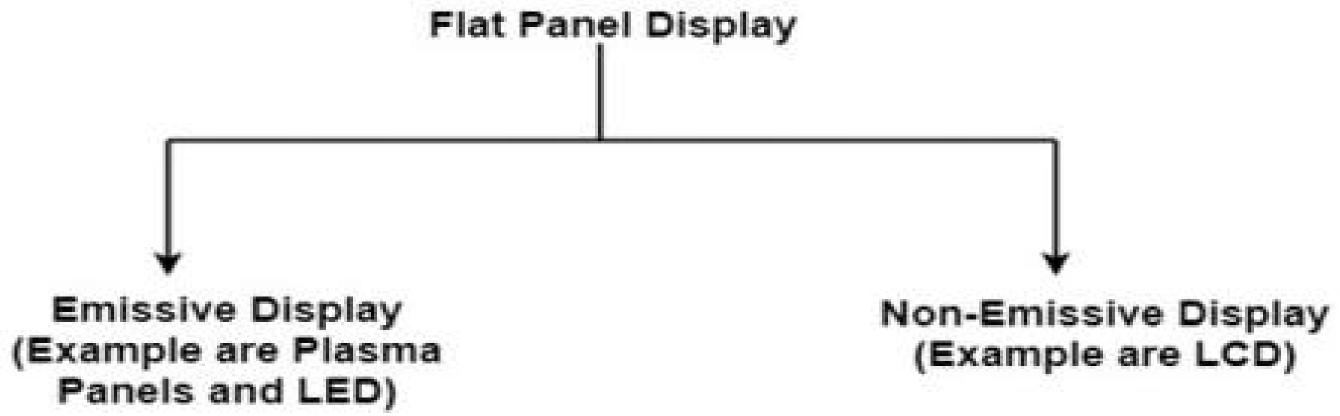


- Normally 1000 scan lines are displayed in this method. Three beams simultaneously expose three inline phosphor dots along scan line.

Flat Panel Display:

The Flat-Panel display refers to a class of video devices that have reduced volume, weight and power requirement compare to CRT.

Example: Small T.V. monitor, calculator, pocket video games, laptop computers, an advertisement board in elevator.



1. Emissive Display: The emissive displays are devices that convert electrical energy into light. Examples are Plasma Panel, thin film electroluminescent display and LED (Light Emitting Diodes).

2. Non-Emissive Display: The Non-Emissive displays use optical effects to convert sunlight or light from some other source into graphics patterns. Examples are LCD (Liquid Crystal Device).

Plasma Panel Display:

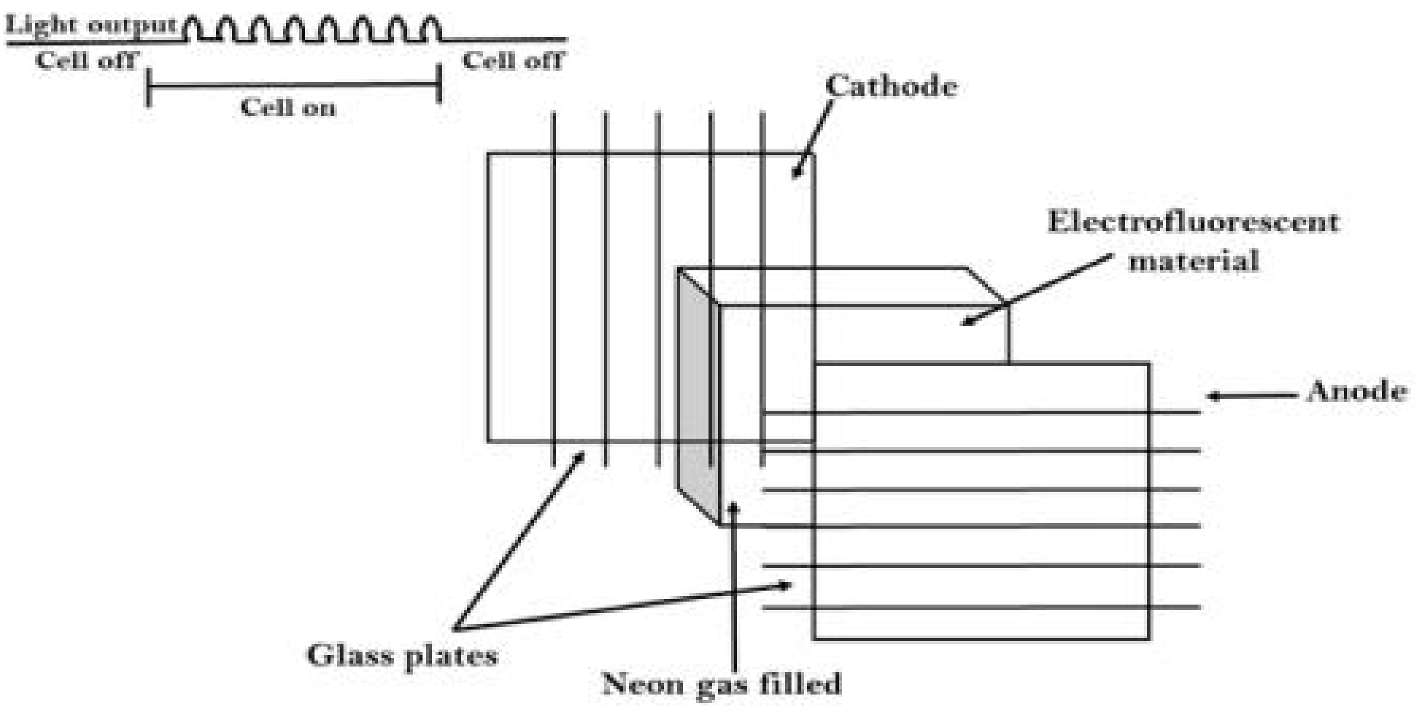
Plasma-Panels are also called as Gas-Discharge Display. It consists of an array of small lights. Lights are fluorescent in nature. The essential components of the plasma-panel display are:

1. **Cathode:** It consists of fine wires. It delivers negative voltage to gas cells. The voltage is released along with the negative axis.
2. **Anode:** It also consists of line wires. It delivers positive voltage. The voltage is supplied along positive axis.
3. **Fluorescent cells:** It consists of small pockets of gas liquids when the voltage is applied to this liquid (neon gas) it emits light.
4. **Glass Plates:** These plates act as capacitors. The voltage will be applied, the cell will glow continuously.

The gas will slow when there is a significant voltage difference between horizontal and vertical wires. The voltage level is kept between 90 volts to 120 volts. Plasma level does not require refreshing. Erasing is done by reducing the voltage to 90 volts.

Each cell of plasma has two states, so cell is said to be stable. Displayable point in plasma panel is made by the crossing of the horizontal and vertical grid. The resolution of the plasma panel can be up to 512 * 512 pixels.

Figure shows the state of cell in plasma panel display:



Advantage:

1. High Resolution
2. Large screen size is also possible.
3. Less Volume
4. Less weight
5. Flicker Free Display

Disadvantage:

1. Poor Resolution
2. Wiring requirement anode and the cathode is complex.
3. Its addressing is also complex.

LED (Light Emitting Diode):

In an LED, a matrix of diodes is organized to form the pixel positions in the display and picture definition is stored in a refresh buffer. Data is read from the refresh buffer and converted to voltage levels that are applied to the diodes to produce the light pattern in the display.

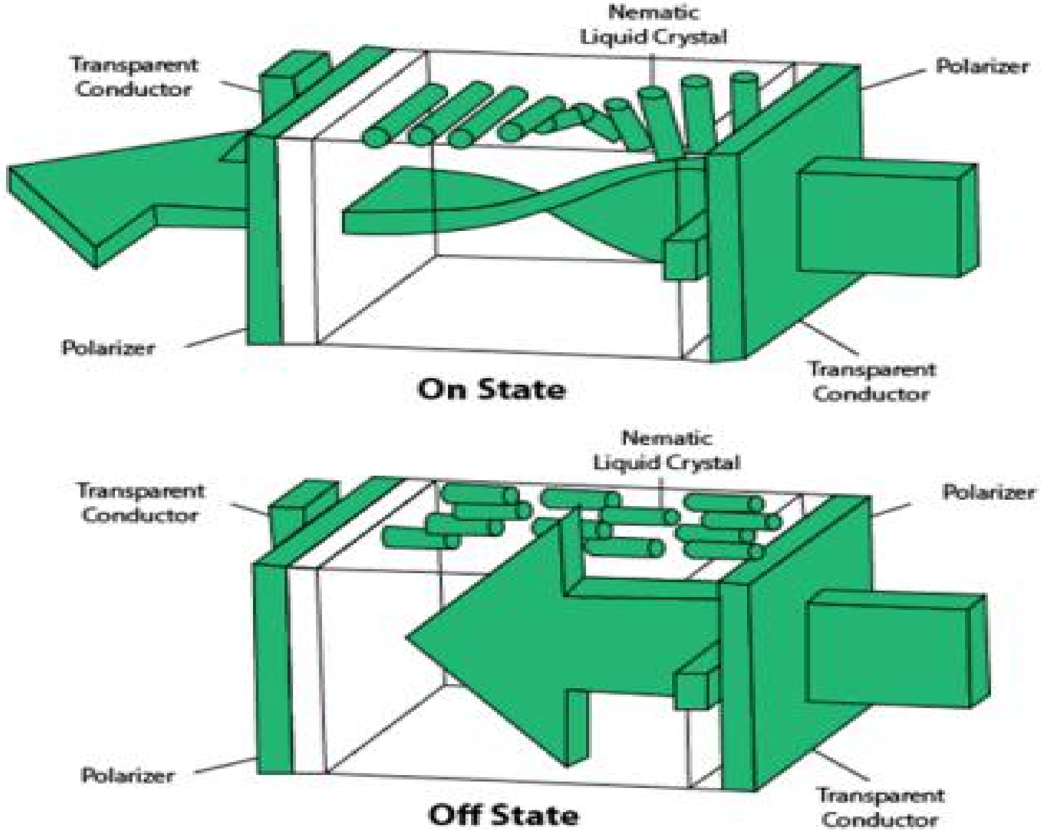
LCD (Liquid Crystal Display):

Liquid Crystal Displays are the devices that produce a picture by passing polarized light from the surroundings or from an internal light source through a liquid-crystal material that transmits the light.

LCD uses the liquid-crystal material between two glass plates; each plate is the right angle to each other between plates liquid is filled. One glass plate consists of rows of conductors arranged in vertical direction. Another glass plate is consisting of a row of conductors arranged in horizontal direction. The pixel position is determined by the intersection of the vertical & horizontal conductor. This position is an active part of the screen.

Liquid crystal display is temperature dependent. It is between zero to seventy degree Celsius. It is flat and requires very little power to operate.

LCD (Liquid Crystal Display):



Liquid Crystal Display

Advantage:

1. Low power consumption.
2. Small Size
3. Low Cost

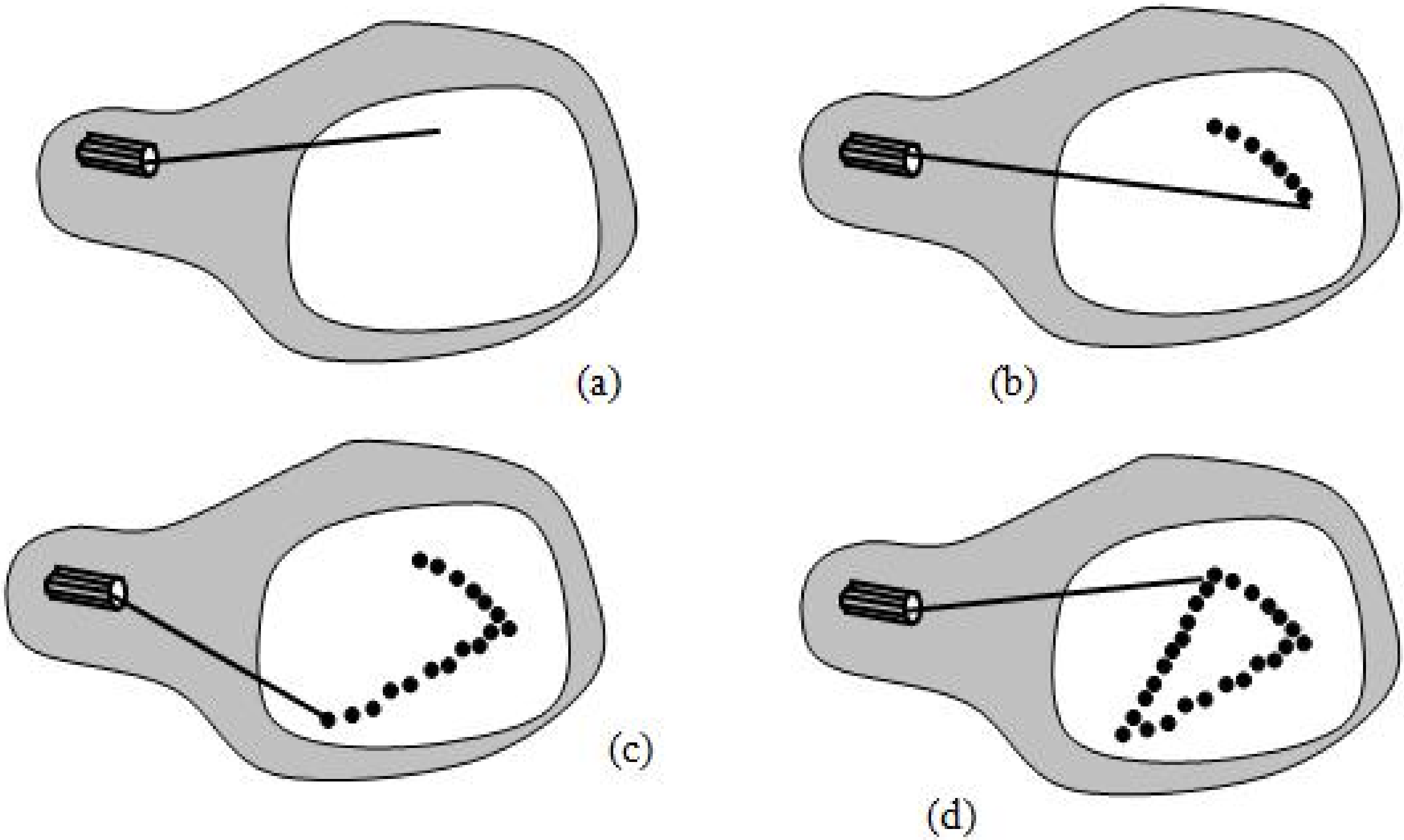
Disadvantage:

1. LCDs are temperature-dependent (0-70°C)
2. LCDs do not emit light; as a result, the image has very little contrast.
3. LCDs have no color capability.
4. The resolution is not as good as that of a CRT.

Basis For Comparison	LED	LCD
Definition	PN-Junction device which discharge visible lights when an electrical charge passes through it.	It is an optical device used for displaying the information in the form of text and images.
Stand For	Light Emitting Diode	Liquid Crystal Display
Backlight	No backlight	Cold cathode fluorescent lamp provides backlight.
Resolution	High	Low
Power Requirement	More	Less
Display Area	Small	Large
Cost	High	Low
Material	Gallium arsenide phosphide.	Liquid crystals and glass electrodes.
Switching Time	Fast	Slow
Direct Current	Do not effects.	Reduces Life Span

Random scan display: (Vector display)

In random scan system, the CRT has the electron beam that is directed only to the parts of the screen where the picture is to be drawn. It draws a picture one line at a time, so it is also called **vector display** (or stroke writing or calligraphic display). The component lines of a picture are drawn and refreshed by random scan system in any specified order.



(a) (b) (c) (d)

Figure: Random Scan Display

Random Scan Display Contd….

- The refresh rate of vector display depends upon the no of lines to be displayed for any image. Picture definition is stored as a set of line drawing instructions in an area of memory called the refresh display file (Display list or display file)
- To display a picture, the system cycles through the set of commands (line drawing) in the display file. After all commands have been processed, the system cycles back to the first line command in the list.
- Random scan systems are designed for drawing all component lines 30 to60 times per second. Such systems are designed for line-drawing applications and can not display realistic shaded scenes. Since CRT beam directly follows the line path, the vector display system produce smooth line.

The video controller:

The video controller is organized as in figure below. The raster-scan generator produces deflection signals that generate the raster scan and also controls the X and Y address registers, which in turn defines memory location to be accessed next. Assume that the frame buffer is addressed in X from 0 to X_{max} and in Y from 0 to Y_{max} then, at the start of each refresh cycle, X address register is set to 0 and Y register is set to 0 (top scan line).

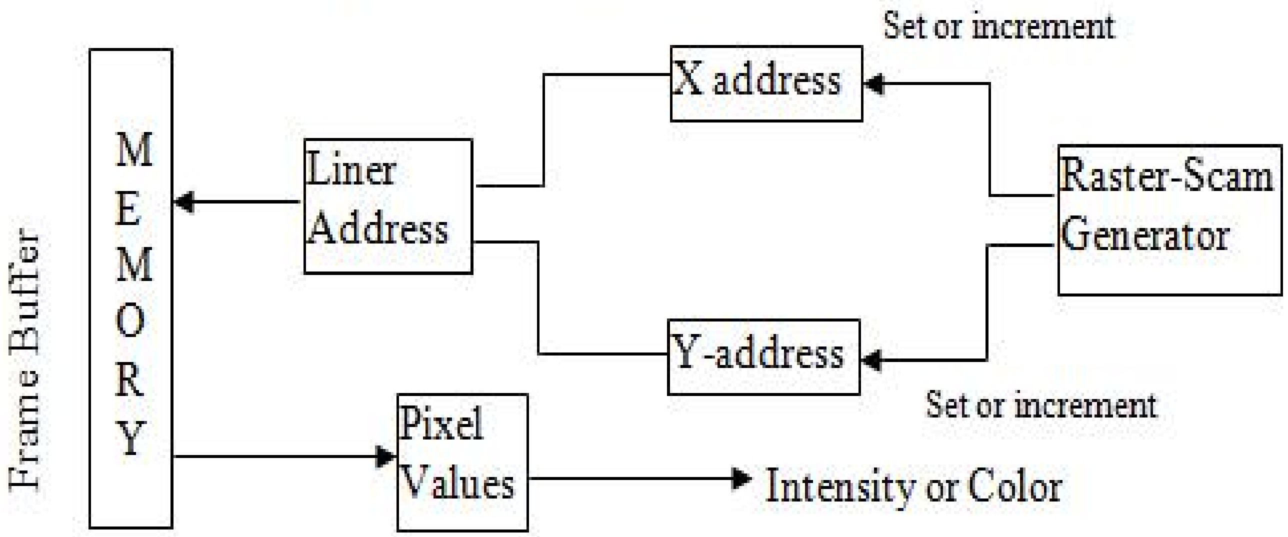


Figure: Organization of Video Controller.

As first scan line is generated, the X address is incremented up to X_{max}. Each pixel value is fetched and used to control the intensity of CRT beam. After first scan line X address is reset to 0 and Y address is incremented by 1. The process is continued until the last scan line ($Y=Y_{max}$) is generated.

. **Vector Display System.**

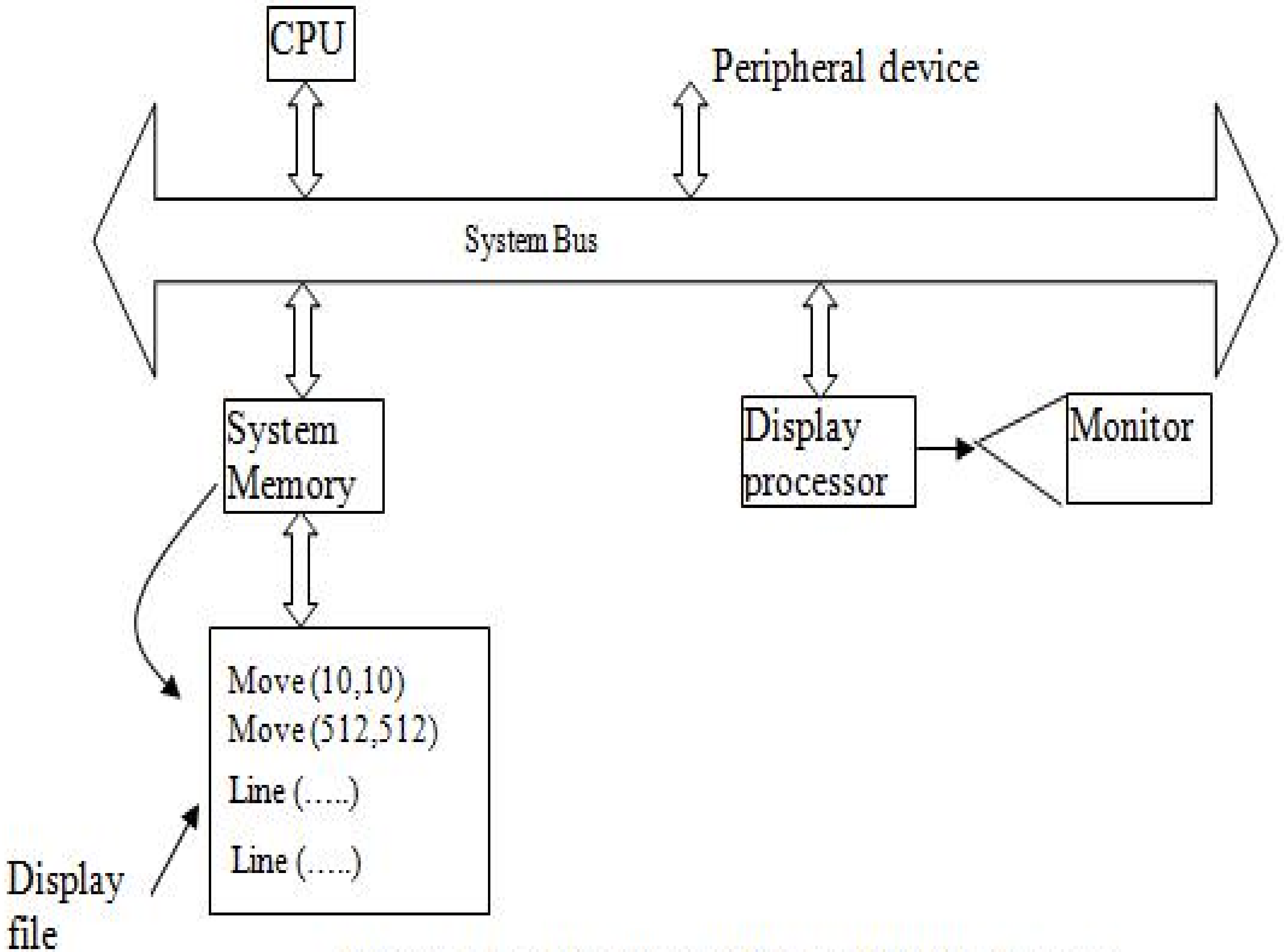


Figure : Architecture of Vector Display System

Vector Display System Contd…..

- Vector display system consists of several units along with peripheral devices. The display processor is also called as graphics controller.
- Graphics package creates a display list and stores in systems memory (consists of points and line drawing commands) called display list or display file.
- Refresh time around so cycle per second.
- Vector display technology is used in monochromatic or beam penetration color CRT.
- Graphics are drawn on a vector display system by directing the electron beam along component line.

Advantages:

- Can produce output with high resolutions.
- Better for animation than raster system since only end point information is needed.

Disadvantages:

- Cannot fill area with pattern and manipulate bits.
- Refreshing image depends upon its complexity.

Raster-Scan Display

- Raster Scan Display is based on television technology. In raster-scan the electron beam is swept across the screen, one row at a time from top to bottom. No of scan line per second is called horizontal scan rate.
- As 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 a memory called frame buffer or refresh buffer. Frame buffer holds all the intensity value for screen points.

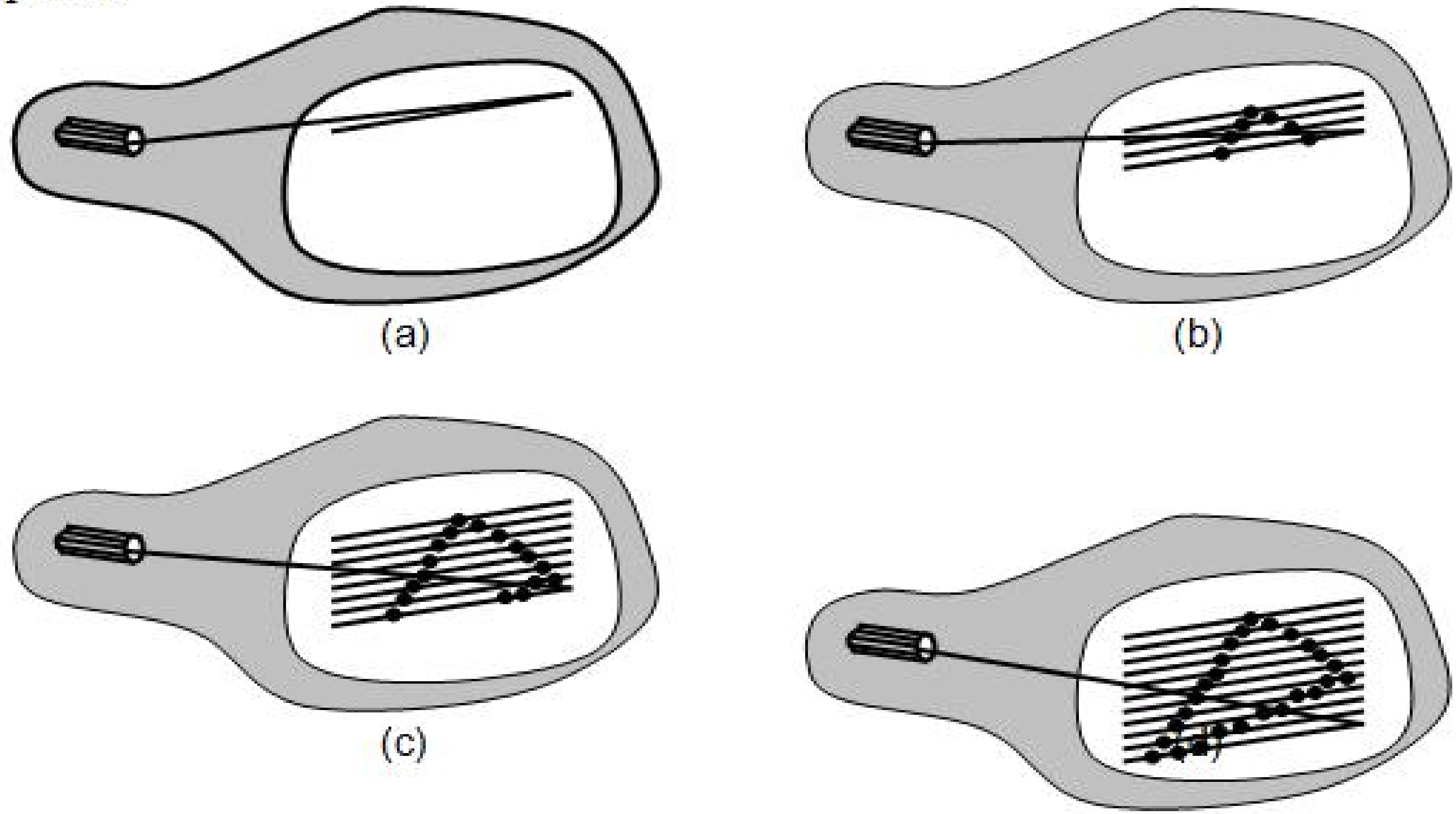


Figure: A raster-scan system displays an object as a set of points across each screen scan line

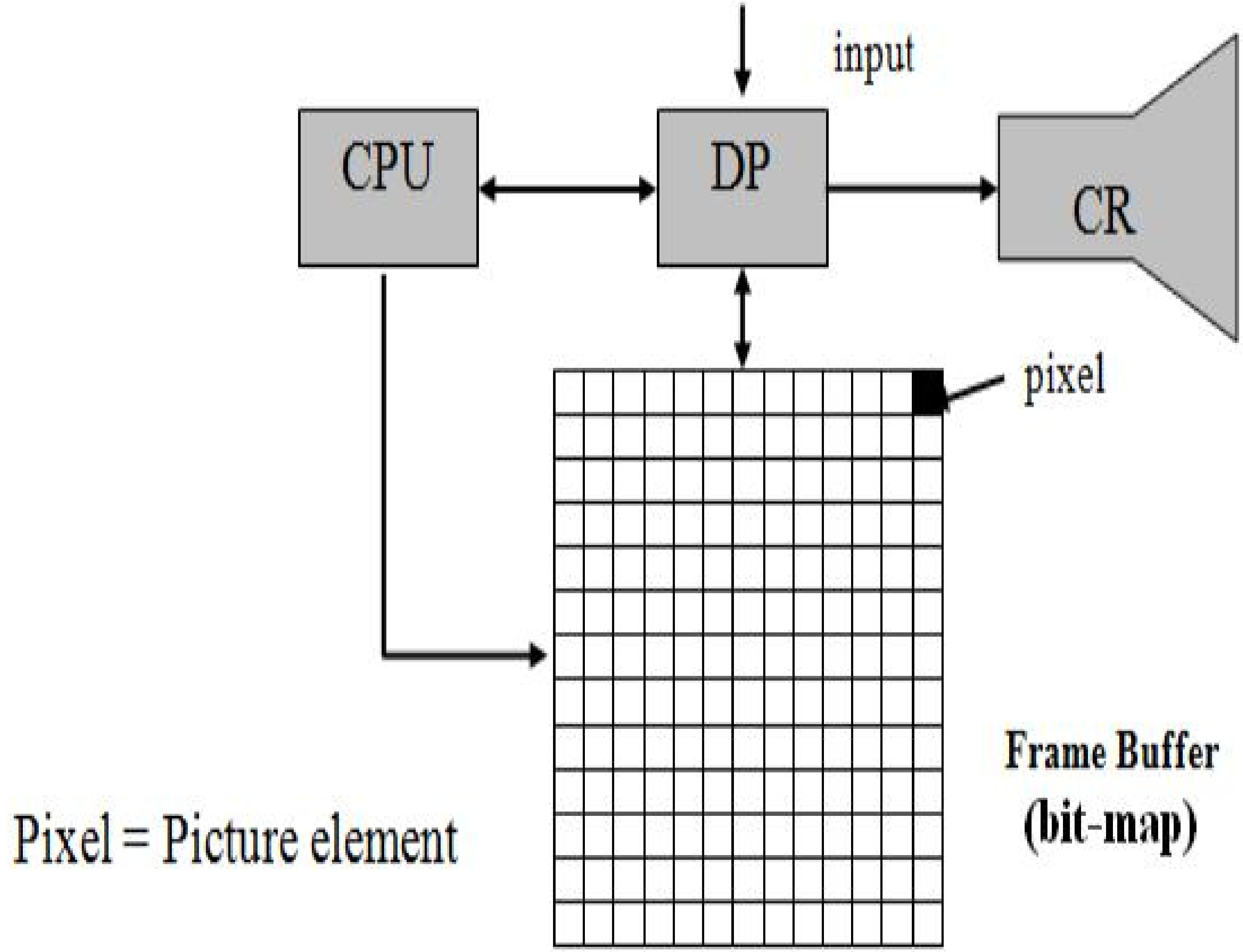


Figure: Raster Scan display system

- The stored intensity value is retrieved from frame buffer and painted on the scan line at a time. Home television are common examples using raster display
- Intensity range for pixel position depends on capability of raster system. For B/W system each point on screen are either on or off, so only one bit per pixel is needed to control the pixel intensity. To display color with varying intensity level, additional bits are needed. Up to 24 to 32 bit per pixel are included in high quality systems, which require more space of storage for the frame buffer, depending upon the resolution of the system.
- A system with 24 bit pixel and screen resolution 1024 × 1024 require 3 megabyte of storage in frame buffer.

  1024*1024 pixel = 1024*1024*24 bits = 3 MB

- The frame butter in B/W system stores a pixel with one bit per pixel so it is termed as bitmap. The frame buffer in multi bit per pixel storage, is called pixmap.
- Refreshing on Raster-Scan display is carried out at the rate of 60 or higher frames per second. 60 frames per second is also termed as 60 cycle per second usually used unit Hertz (HZ)
- Returning of electron beam from right end to deft end after refreshing each scan line is **horizontal retrace** . At the end of each frame, the electron beam returns to the top left corner to begin next frame called **vertical retrace**.

Interlaced: Display in two pass with interlacing.

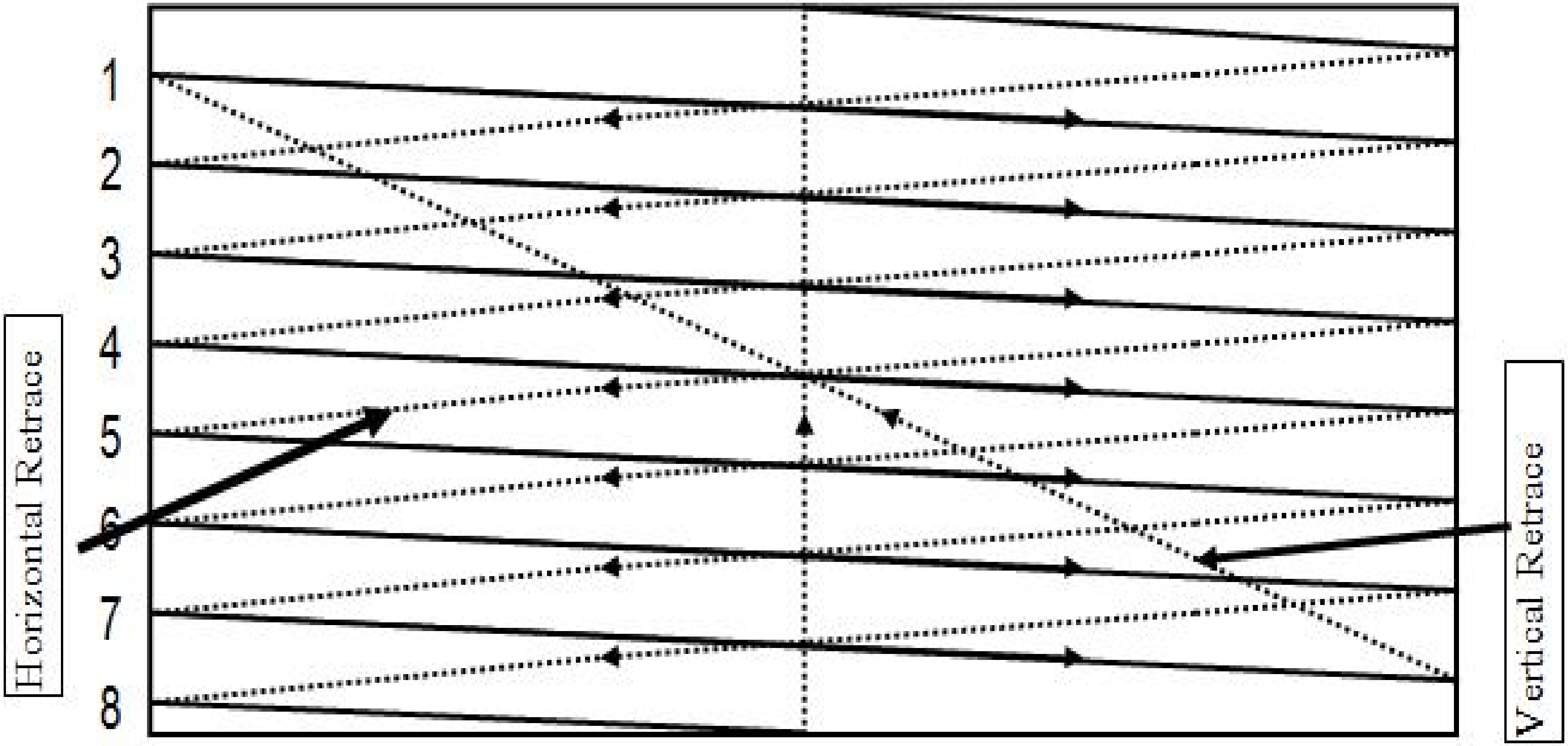


Figure: Horizontal retrace and Vertical retrace

***Question**: Consider a RGB raster system is to be designed using 8 inch by 10 inch screen with a resolution of 100 pixels per inch in each direction. If we want to store 8 bits per pixel in the frame buffer, How much storage(in bytes) do we need for the frame buffer?*

Solution: Size of screen = 8 inch × 10 inch.

Pixel per inch(Resolution) = 100.

Then, Total no of pixels = 8×100 × 10 × 100 pixels

Bit per pixel storage = 8

Therefore Total storage required in frame buffer = (800 ×1000×8) bits

= (800 ×1000×8)/8 Bytes

= 800000 Bytes.

Architecture of Raster Scan System:

The raster graphics systems typically consist of several processing units. CPU is the main processing unit of computer systems. Besides CPU, graphics system consists of a special purpose processor called video controller or display processor. The display processor controls the operation of the display device.

The organization of raster system is as shown below

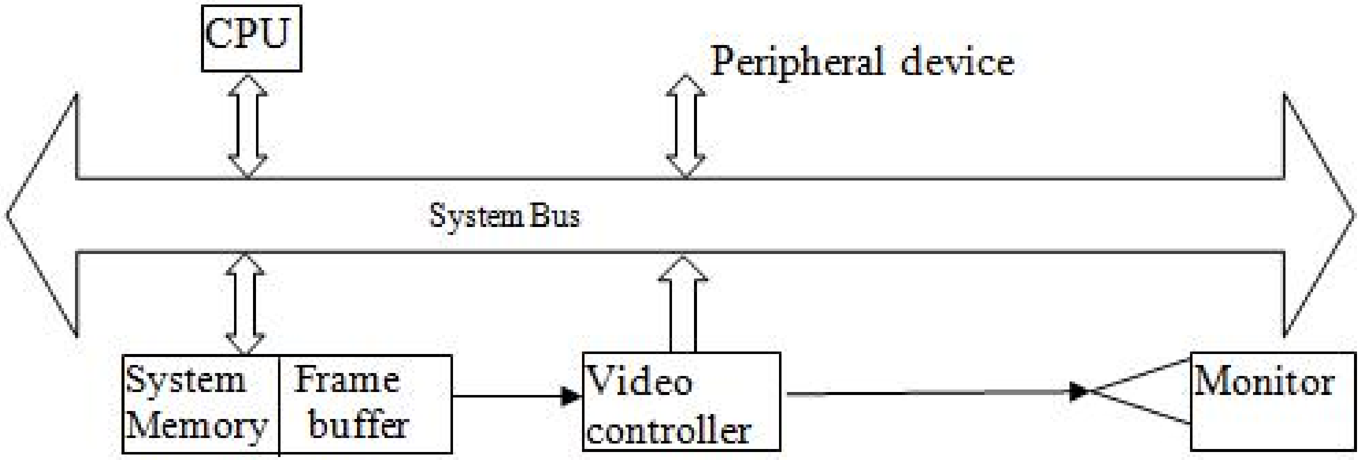


Figure: A simple Raster System.

- A fixed area of system memory is reserved for the frame buffer. The video controller has the direct access to the frame buffer for refreshing the screen.
- The video controller cycles through the frame buffer, one scan line at a time, typically at 60 times per second or higher. The contents of frame buffer are used to control the CRT beam's intensity or color.

Difference between random scan and raster scan

Base of Difference	Raster Scan System	Random Scan System
Electron Beam	The electron beam is swept across the screen, one row at a time, from top to bottom.	The electron beam is directed only to the parts of screen where a picture is to be drawn.
Resolution	Its resolution is poor because raster system in contrast produces zigzag lines that are plotted as discrete point sets.	Its resolution is good because this system produces smooth lines drawings because CRT beam directly follows the line path.
Picture Definition	Picture definition is stored as a set of intensity values for all screen points, called pixels in a refresh buffer area.	Picture definition is stored as a set of line drawing instructions in a display file.
Realistic Display	The capability of this system to store intensity values for pixel makes it well suited for the realistic display of scenes contain shadow and color pattern.	These systems are designed for line-drawing and can't display realistic shaded scenes.
Draw an Image	Screen points/pixels are used to draw an image.	Mathematical functions are used to draw an image.

Line Drawing Algorithms.

The slope-intercept equation of a straight line is:

$y = mx + b$ where m = slope of line and , b = y-intercept.

for any two given points (x_1, y_1) and (x_2, y_2)

$$\text{slope (m)} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\therefore b = y - \frac{y_2 - y_1}{x_2 - x_1} x \quad \text{from above equation i.e. } y = mx + b$$

At any point (x_k, y_k)

$$y_k = mx_k + b \ldots\ldots\ldots\ldots\ldots\ldots 1$$

At (x_{k+1}, y_{k+1}),

$$y_{k+1} = mx_{k+1} + b \ldots\ldots\ldots\ldots\ldots 2.$$

subtracting 1 from 2 we get-

$y_{k+1} - y_k = m(x_{k+1} - x_k)$

Here $(y_{k+1} - y_k)$ is increment in y as corresponding increment in x.

$\therefore \Delta y = m.\Delta x$

or $m = \dfrac{\Delta y}{\Delta x}$

For incremental algorithm in line drawing ,

- Increment x by 1
- Computer corresponding y and display pixel at position $(x_i, round(y_i))$

Problem: Floating point multiplication & addition

- The round function.

DDA line Algorithm:

The digital differential analyzer (DDA) is a scan conversion line drawing algorithm based on calculating either Δx or Δy form the equation,

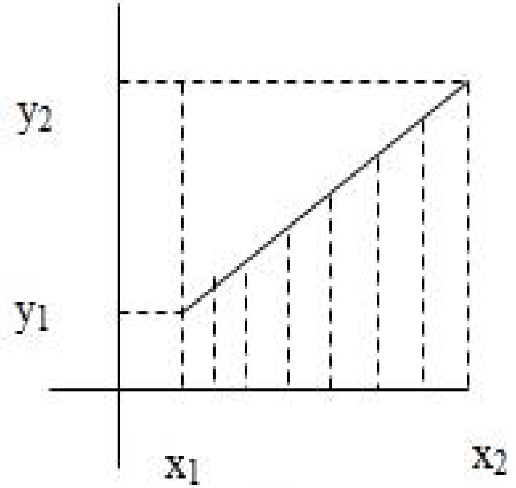
$$\Delta y = m.\Delta x.$$

We sample the line at unit intervals in one co-ordinate and determine the corresponding integer values nearest to the line path for the other co-ordinates.

Consider a line with positive slope.

If m<=1, we sample x co-ordinate. So $\Delta x = 1$ and compute each successive y value as:

$$y_{k+1} = y_k + m \qquad \because m = \frac{\Delta y}{\Delta x}, \Delta x = 1$$

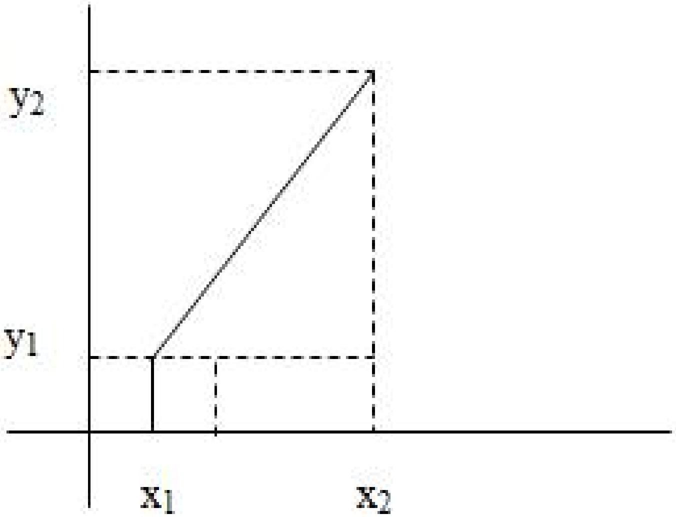


$m<=1$

Here k takes value from starting point and increase by 1 until final end point. m can be any real value between 0 and 1.

For line with positive slope greater than 1, we sample $\Delta y = 1$ and calculate corresponding x values as

$$x_{k+1} = x_k + \frac{1}{m} \quad \because m = \frac{\Delta y}{\Delta x}, \Delta y = 1$$



$m > 1$

The above equations are under the assumption that the lines are processed from left to right. i.e. left end point is starting. If the processing is from right to left, we can sample $\Delta y = -1$ for line $|m| < 1$

$$\therefore y_{k+1} = y_k - m,$$

If $|m|>1$, $\Delta y = -1$ and calculate

$$x_{k+1} = x_k - \frac{1}{m}.$$

DDA Algorithm

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

Step 1 – Get the input of two end points (X_0, Y_0) and (X_1, Y_1).

Step 2 – Calculate the difference between two end points.

```
dx = X1 - X0
dy = Y1 - Y0
```

Step 3 – Based on the calculated difference in step-2, 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.

```
if (dx > dy)
   Steps = absolute(dx);
else
   Steps = absolute(dy);
```

Step 4 – Calculate the increment in x coordinate and y coordinate.

```
Xincrement = dx / (float) steps;
Yincrement = dy / (float) steps;
```

Step 5 – Put the pixel by successfully incrementing x and y coordinates accordingly and complete the drawing of the line.

```
for(int v=0; v < Steps; v++)
{
   x = x + Xincrement;
   y = y + Yincrement;
   putpixel(x,y);
}
```

Drawbacks of DDA algorithm

- The DDA algorithm is faster method for calculating pixel position but it has problems:
- m is stored in floating point number.
- round of error
- error accumulates when we proceed line.
- so line will move away from actual line path for long line.

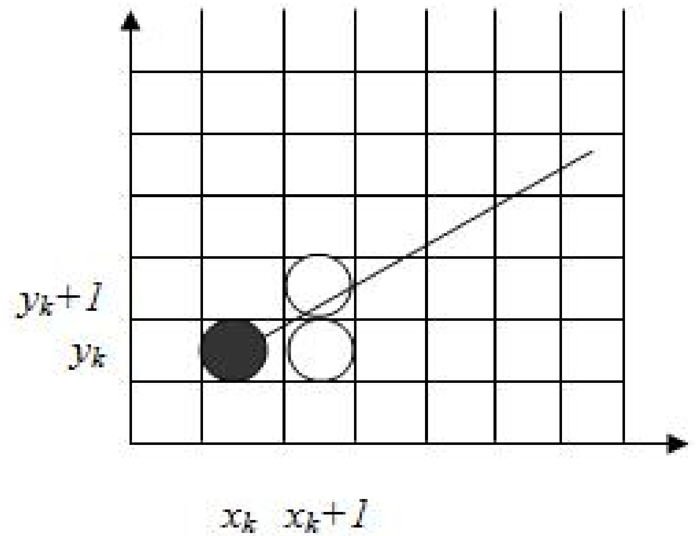
Bresenham's Line algorithm:

An accurate and efficient line generating algorithm, developed by Bresenham that scan converts lines only using integer calculation to find the next (x,y) position to plot. It avoids incremental error accumulation.

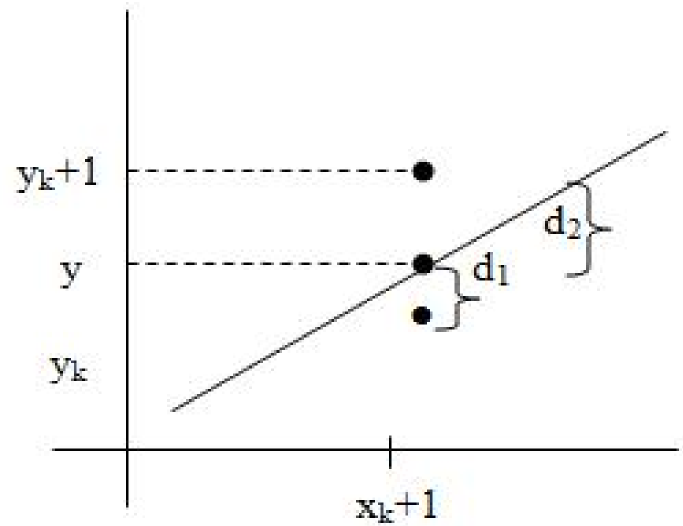
Line with positive slope less than 1 (0<m<1)

Pixel position along the line path are determined by sampling at unit x intervals. Starting from left end point, we step to each successive column and plot the pixel closest to line path.

Assume that (x_k,y_k) is pixel at k^{th} step then next point to plot may be either (x_k+1, y_k) or (x_k+1, y_k+1)



At sampling position x_k+1, we label vertical pixel separation from line path as d_1 & d_2 as in figure .
The y-coordinate on the mathematical line path at pixel column x_k+1 is $y=m(x_k+1)+b$



Then $d_1 = y - y_k$

$= m(x_k + 1) + b - y_k$

$d_2 = (y_k + 1) - y$

$= (y_k + 1) - m\,(x_k + 1) - b$

Now $d_1 - d_2 = \;\; 2m\,(x_k + 1) - (y_k + 1) - y_k + 2b$

$= \;\; 2m\,(x_k + 1) - 2y_k + 2b - 1$

A decision parameter p_k for the k[th] step in the line algorithm can be obtained by substituting $m = \dfrac{\Delta y}{\Delta x}$ in above eq[n] and defining

$$p_k = \Delta x(d_1 - d_2)$$

$$= \Delta x[2\frac{\Delta y}{\Delta x}(x_k + 1) - 2y_k + 2b - 1]$$

$$= 2\Delta y.x_k - 2\Delta x.y_k + 2\Delta y + \Delta x(2b - 1)$$

$$= 2\Delta y.x_k - 2\Delta x.y_k + c$$

Where the constant $c = 2\Delta y - \Delta x(2b - 1)$ which is independent.

If decision parameter p_k is negative i.e. $d_1 < d_2$, pixel at y_k is closer to the line path than pixel at y_k+1 . In this case we plot lower pixel. $(x_k+1\,,\,y_k)$. *other wise plot upper pixel* (x_k+1, y_k+1).

Co-ordinate change along the line occur in unit steps in either x, or y direction. Therefore we can obtain the values of successive decision parameters using incremental integer calculations.

At step $k+1$, p_{k+1} is evaluated as.

$$p_{k+1} = 2\Delta y.x_{k+1} - 2\Delta x y_{k+1} + c$$

$$\therefore p_{k+1} - p_k = 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k)$$

Since $x_{k+1} = x_k + 1$

$$\therefore p_{k+1} = p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

The term $y_{k+1} - y_k$ is either 0 or 1 depending upon the sign of p_k.

The first decision parameter p_0 is evaluated as.

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

and successively we can calculate decision parameter as

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

so if p_k is negative $y_{k+1} = y_k$ so $p_{k+1} = p_k + 2\Delta y$

otherwise $y_{k+1} = y_k + 1$, *then* $p_{k+1} = p_k + 2\Delta y - 2\Delta x$

Algorithm:

1. Input the two line endpoint and store the left endpoint at (x_o, y_o)
2. Load (x_o, y_o) in to frame buffer, i.e. Plot the first point.
3. Calculate constants $2\Delta x, 2\Delta y$ calculating $\Delta x, \Delta y$ and obtain first decision parameter value as

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

4. At each x_k along the line, starting at k=0, perform the following test,

   if $p_k < 0$, next point is $(x_k + 1, y_k)$

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

   otherwise

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

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

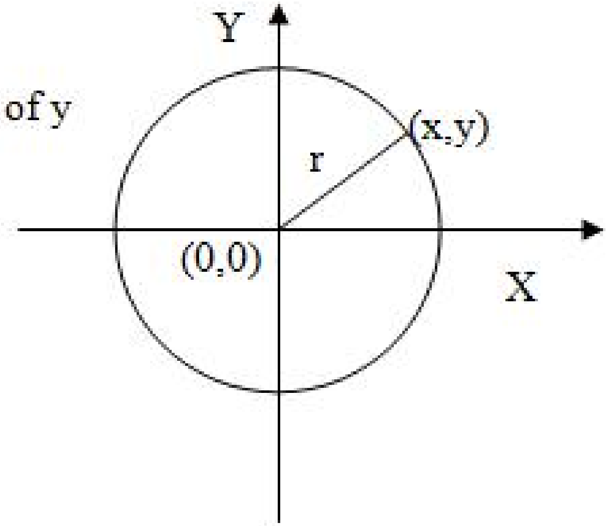
5. Repeat step 4 Δx times.

Algorithm for Circle

Simple Algorithm:

The equation of circle centered at origin and radius r is given by $x^2 + y^2 = r^2$

$$\Rightarrow y = \pm\sqrt{r^2 - x^2}$$



- Increment x in unit steps and determine corresponding value of y from the equation above. Then set pixel at position (x,y).
- The steps are taken from $-r$ to $+r$.
- In computer graphics, we take origin at upper left corner point on the display screen i.e. first pixel of the screen so any visible circle drawn would be centered at point other than (0,0). If center of circle is (xc,yc) then the calculated point from origin center should be moved to pixel position by $(x+xc, y+yc)$.

In general the equation of circle centered at (xc,yc) and radius r is

$$(x - xc)^2 + (y - yc)^2 = r^2$$

$$\Rightarrow y = yc \pm \sqrt{r^2 - (x - xc^2} \text{ ....(1)}$$

Use this equation to calculate the position of points on the circle. Take unit step from xc-r to $xc+r$ for x value and calculate the corresponding value of y- position for pixel position (x,y). This algorithm is simple but,

- Time consuming – square root and squares computations
- Non – uniform spacing , due to changing slope of curve. If non-uniform spacing is avoided by interchanging x and y for slope |m|>y, this leads to more computation.

Drawing circle using polar equations

If (x,y) be any point on the circle boundary with center (0,0) and radius r, then

$x = r\cos\theta$

$x = r\sin\theta$

i.e. $(x, y) = (r\cos\theta, r\sin\theta)$

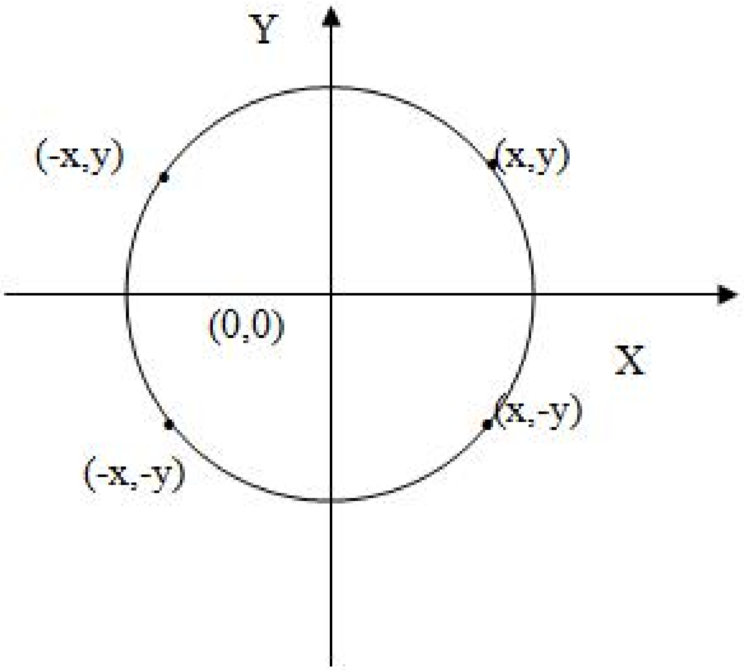
To draw circle using these co-ordinates approach, just increment angle starting from 0 to 360. Compute (x,y) position corresponding to increment angle. Which draws circle centered at origin, but the circle centered at origin is not visible completely on the screen since (0,0) is the starting pixel of the screen. If center of circle is given by (xc,yc) then the pixel position (x,y) on the circle path will be computed as

$x = xc + r\cos\theta$

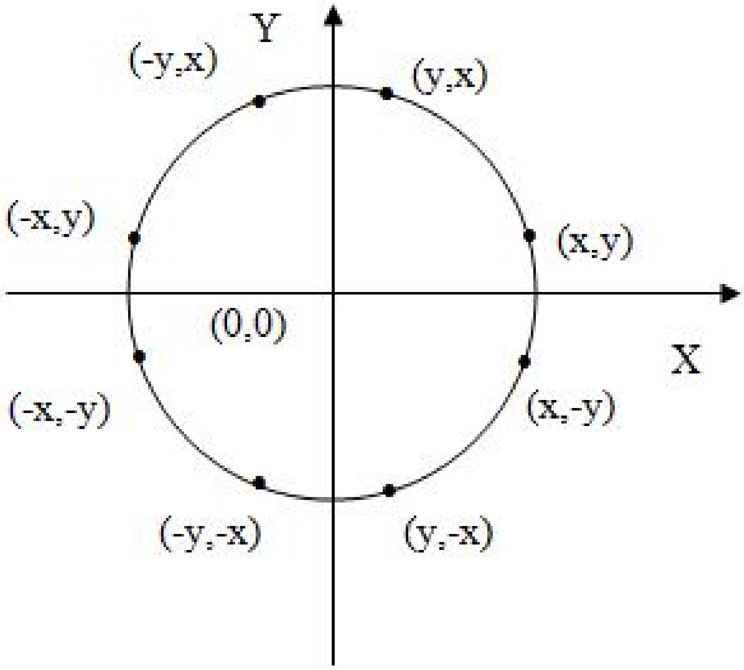
$y = yc + r\sin\theta$

Symmetry in circle scan conversion:

We can reduce the time required for circle generation by using the symmetries in a circle e.g. 4-way or 8-way symmetry. So we only need to generate the points for one quadrant or octants and then use the symmetry to determine all the other points.



4-way symmetry



8-way symmetry

Problem of computation still persists using symmetry since there are square roots, trigonometric functions are still not eliminated in above algorithms.

Mid point circle Algorithm:

In mid point circle algorithm, we sample at unit intervals and determine the closest pixel position to the specified circle path at each step.

For a given radius r, and screen center position (xc, yc), we can set up our algorithm to calculate pixel positions around a circle path centered at (0,0) and then each calculated pixel position (x, y) is moved to its proper position by adding xc to x and yc to y.

$i.e.\ x = x + xc,\ y = y + yc.$

To apply the mid point method, we define a circle function as:

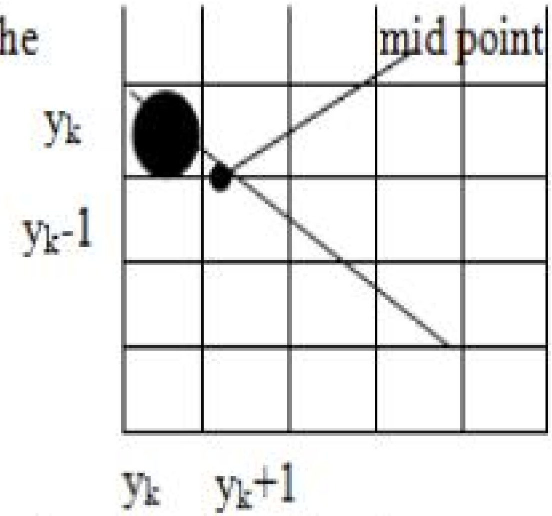
$$f_{circle} = x^2 + y^2 - r^2$$

To summarize the relative position of point (x, y) by checking sign of f_{circle} function,

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

The circle function tests are performed for the mid positions between pixels near the circle path at each sampling step. Thus the circle function is decision parameter in mid point algorithm.

The figure, shows the mid point between the two candidate pixel at sampling position $x_k + 1$. Assuming we have just plotted the pixel (x_k, y_k), we next need to determine whether the pixel at position $(x_k + 1, y_k) or (x_k + 1, y_k - 1)$ is closer to the circle.



Our decision parameter is circle function evaluated at the mid point

$$p_k = f_{circle}(x_k + 1, y_k - \frac{1}{2})$$

$$= (x_k + 1)^2 + (y_k - 1/2)^2 - r^2 = (x_k + 1)^2 + y_k^2 - y_k + \frac{1}{4} - r^2$$

If $p_k < 0$, then mid-point lies inside the circle, so point at y_k is closer to boundary otherwise, $y_k - 1$ closer to choose next pixel position.

Successive decision parameters are obtained by incremental calculation. The decision parameter for next position is calculated by evaluating circle function at sampling position $x_{k+1}+1$ i.e. x_k+2 as

$$p_{k+1} = f_{circle}(x_{k+1}+1, y_{k+1}-\frac{1}{2})$$

$$= \{(x_{k+1}+1)\}^2 + (y_{k+1}-\frac{1}{2})^2 - r^2$$

$$= (x_{k+1})^2 + 2x_{k+1} + 1 + (y_{k+1})^2 - (y_{k+1}) + \frac{1}{4} - r^2$$

Now, $p_{k+1} - p_k = 2x_{k+1} + (y^2{}_{k+1} - y_k{}^2) - (y_{k+1} - y_k) + 1$

i.e. $p_{k+1} = p_k + 2x_{k+1} + (y^2{}_{k+1} - y_k{}^2) - (y_{k+1} - y_k) + 1$

Where y_{k+1} is either y_k or $y_k - 1$ depending upon sign of p_k. and $x_{k+1} = x_k + 1$

If p_k is negative, $y_{k+1} = y_k$ so we get,

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

If p_k is positive, $y_{k+1} = y_k - 1$ so we get,

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

Where $2x_{k+1} = 2x_k + 2$

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

At the start position, (0,r), these two terms have the values 0 an 2r, respectively. Each successive values are obtained by adding 2 to the previous value of 2x and subtracting 2 from previous value of 2y.

The initial decision parameter is obtained by evaluating the circle function at starting position $(x_0, y_0) = (0, r)$.

$$p_0 = f_{circle}(1, r - 1/2)$$

$$= 1 + (r - 1/2)^2 - r^2$$

$$= 1 + r^2 - r + \frac{1}{4} - r^2$$

$$= \frac{5}{4} - r$$

If p_0 is specified in integer,

$$p_0 = 1 - r.$$

The Algorithm:

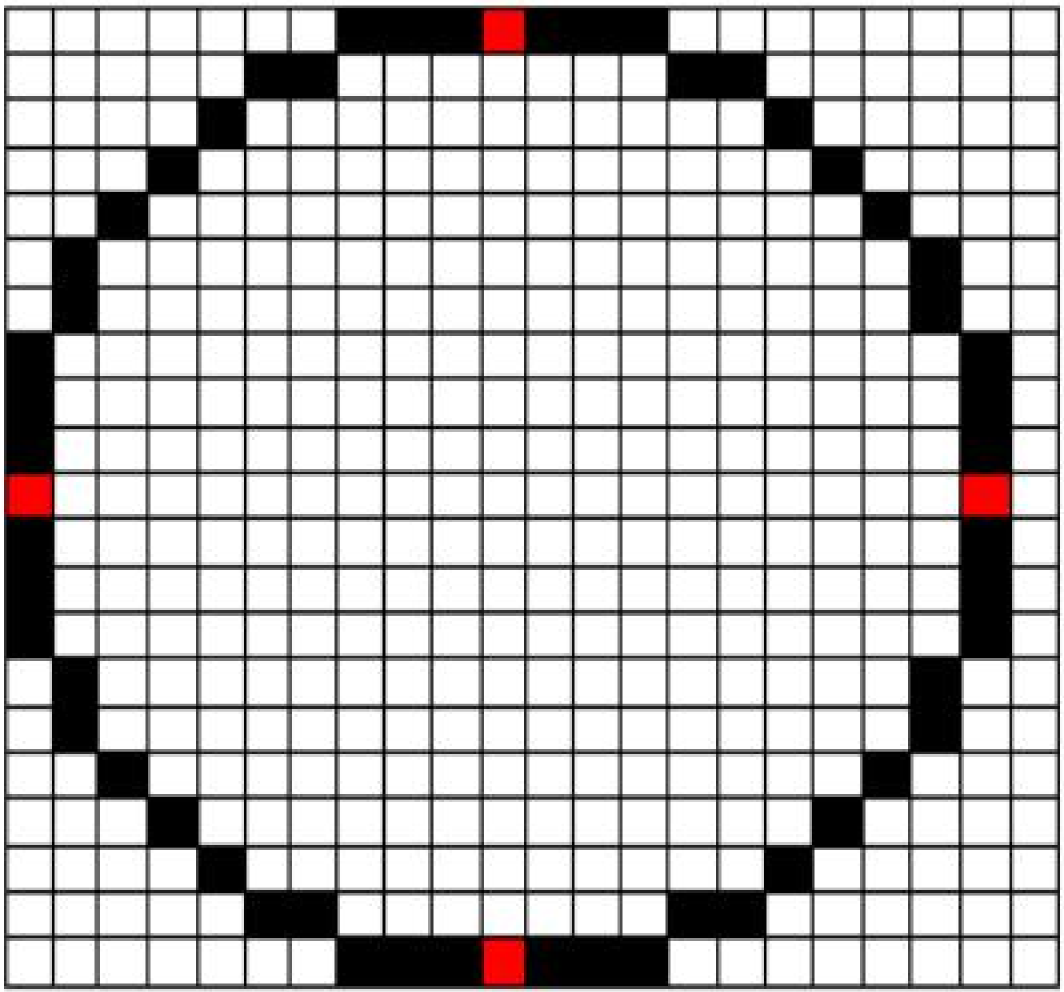
1. Input radius r and circle centre (x_c, y_c), and obtain the first point on circle centered at origin as.
$$(x_0, y_0) = (0, r).$$
2. Calculate initial decision parameter
$$p_0 = \frac{5}{4} - r$$
3. At each x_k position, starting at $k = 0$, perform the tests:
If $p_k < 0$ next point along the circle centre at (0,0) is $(x_k + 1, y_k)$
$$p_{k+1} = p_k + 2x_{k+1} + 1)$$
Otherwise, the next point along circle is $(x_k + 1, y_k - 1)$
$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$
Where $2x_{k+1} = 2x_k + 2.$ and $2y_{k+1} = 2y_k - 2.$
4. Determine symmetry point on the other seven octants.
5. Move each calculated pixels positions (x, y) in to circle path centered at (x_c, y_c) as
$$x = x + x_c, y = y + y_c$$
6. Repeat 3 through 5 until $x \geq y$.

- Calculate the required points to plot the following circle using the midpoint circle algorithm.

Radius (r) = 10, Centre = (3, 4)

$P_0=1-r=-9$, Initial Point $(x_0, y_0) = (0,10)$

k	P_k	(X_{k+1}, Y_{k+1})	Other Quadrants	Other Octants	All Points after adding the centres
0	-9	(1,10)	(-1, 10)(1, -10)(-1, -10)	(1,10)(10,1)(-1,10)(10,-1)(1,-10)(-10,1)(-1,-10)(-10,-1)	(4,14)(13,5)(2,14)(13,3)(4,-6)(-7,5)(2,-6)(-7,3)
1	-6	(2,10)	(-2, 10)(2, -10)(-2, -10)	(2,10)(10,2)(-2,10)(10,-2)(2,-10)(-10,2)(-2,-10)(-10,-2)	...
2	-1	(3,10)	(-3, 10)(3, -10)(-3, -10)	(3,10)(10,3)(-3,10)(10,-3)(3,-10)(-10,3)(-3,-10)(-10,-3)	...
3	6	(4,9)	(-4, 9)(4, -9)(-4, -9)	(4,9)(9,4)(-4,9)(9,-4)(4,-9)(-9,4)(-4,-9)(-9,-4)	...
4	-3	(5,9)	(-5, 9)(5, -9)(-5, -9)	(5,9)(9,5)(-5,9)(9,-5)(5,-9)(-9,5)(-5,-9)(-9,-5)	...
5	8	(6,8)	(-6, 8)(6, -8)(-6, -8)	(6,8)(8,6)(-6,8)(8,-6)(6,-8)(-8,6)(-6,-8)(-8,-6)	...
6	5	(7,7)	(-7, 7)(7, -7)(-7, -7)	(7,7)(-7,7)(7,-7)(-7,-7)	...



Ellipse Algorithm

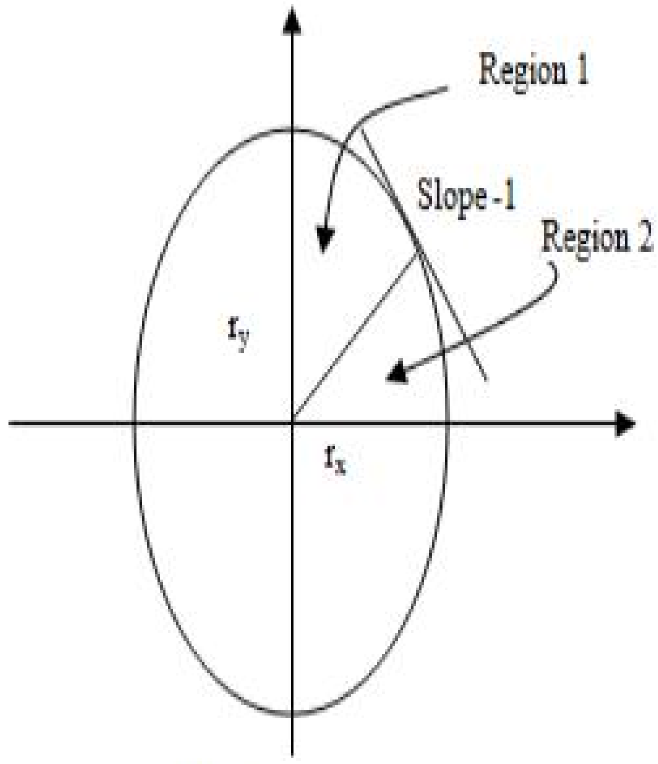
The basic algorithm for drawing ellipse is same as circle computing x and y position at the boundary of the ellipse from the equation of ellipse directly.

We have equation of ellipse centered at origin (0,0) is

$$\frac{x^2}{r_x^{\ 2}} + \frac{y^2}{r_y^{\ 2}} = 1 \text{ which gives}$$

$$y = \pm \frac{r_y}{r_x} \sqrt{(r_x^{\ 2} - x^2)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Stepping unit interval in x direction from $-r_x$ to r_x we can get corresponding y value at each x position which gives the ellipse boundary co-ordinates. Plotting these computed points we can get the ellipse.



Figure

If center of ellipse is any arbitrary point *(xc,yc)* then the equation of ellipse can be written as

$$\frac{(x-xc)^2}{r_x^{\ 2}}+\frac{(y-yc)^2}{r_y^{\ 2}}=1$$

i.e. $y = yc \pm \frac{r_y}{r_x}\sqrt{r_x^{\ 2}-(x-xc)^2}$ ------------------(2)

for any point *(x,y)*on the boundary of the ellipse If major axis of ellipse with major axis along X-axis the algorithm based on the direct computation of ellipse boundary points can be summarized as

1. Input the center of ellipse *(xc,yc)* x-radius *xr* and y-radius *yr*.
2. For each x position starting from xc-r and stepping unit interval along x-direction, compute corresponding y positions as

   $$y = yc \pm \frac{r_y}{r_x}\sqrt{r_x^{\ 2}-(x-xc)^2}$$

3. Plot the point (x , y).
4. Repeat step 2 to 3 until x >= xc+xr

Computation of ellipse using polar co-ordinates:

Using the polar co-ordinates for ellipse, we can compute the (x,y) position of the ellipse boundary using the following parametric equations

$$x = xc + r\cos\theta$$
$$y = yc + r\sin\theta$$

The algorithm based on these parametric equation on polar co-ordinates can be summarized as below.

1. Input center of ellipse *(xc,yc)* and radii xr and yr.
2. Starting θ from angle 0° step minimum increments and compute boundary point of ellipse as

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

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

3. Plot the point at position (round(x), round(y))
4. Repeat until θ is greater or equal to 360°.

```
void drawellipse(int xc,int yc,int rx,int ry)
{
        int x,y;
        float theta;
        const float PI=3.14;
        for(theta=0.0;theta<=360;theta+=1)
        {
                x= xc+rx*cos(theta*PI/180.0);
                y= yc+ry*sin(theta*PI/180.0);
                putpixel(x,y,1);

        }
}
```

The methods of drawing ellipses explained above are not efficient. The method based on direct equation of ellipse must perform the square and square root operations due to which there may be floating point number computation which cause rounding off to plot the pixels. Also the square root causes the domain error. Due to the changing slope of curve along the path of ellipse, there may be un-uniform separation of pixel when slope changes. To eliminate this problem , extra computation is needed. Although , the method based on polar co-ordinate parametric equation gives the uniform spacing of pixel due to uniform increment of angle but it also take extra computation to evaluate the trigonometric functions. So these algorithms are not efficient to construct the ellipse. We have another algorithm called mid- point ellipse algorithm similar to mid-point circle algorithm which is efficient algorithm for computing ellipse.

Mid-Point Ellipse Algorithm:

The mid-point ellipse algorithm decides which point near the boundary (i.e. path of the ellipse) is closer to the actual ellipse path described by the ellipse equation. That point is taken as next point .

It is applied to the first quadrant in two parts as in figure. Region 1 and Region 2.We process by taking unit steps in x-coordinates direction and finding the closest value for y for each x-steps in region 1.

In first quadrant at region 1, we start at position *(0,ry)* and incrementing x and calculating y closer to the path along clockwise direction . When slope becomes -1 then shift unit step in x to y and compute corresponding x closest to ellipse path at Region 2 in same direction.

Alternatively , we can start at position *(rx,0)* and select point in counterclockwise order shifting unit steps in y to unit step in x when slope becomes greater than -1.

Here, to implement mid-point ellipse algorithm, we take start position at *(0,ry)* and step along the ellipse path in clockwise position throughout the first quadrant.

We define ellipse function center at origin i.e, *(xc,yc)= (0,0)* as

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

$$f_{ellipse}(x,y) = \begin{cases} <0, \text{ if (x,y) lies inside boundary of ellipse.} \\ =0 \text{ if (x,y lies on the boundary of ellipse.} \\ >0 \text{ if (x,y) lies outside the boundary of ellipse.} \end{cases}$$

So $f_{ellipse}$ function serves as decision parameter in ellipse algorithm at each sampling position. We select the next pixel position according to the sign of decision parameter.

Starting at $(0,r_y)$, we take unit step in x-direction until we reach the boundary between the region 1 and region 2. Then we switch unit steps in y over the remainder of the curve in first quadrant. At each step, we need to test the slope of curve. The slope of curve is calculated as;

$$\frac{dy}{dx} = -\frac{2r_y^2x}{2r_x^2y}$$

At the boundary between region 1 and region 2,

$\frac{dy}{dx} = -1$ and $2r_y^2 = 2r_x^2$ Therefore , we move out of region 1 when $2r_y^2x \geq 2r_x^2$

Assuming the position (x_k, y_k) is filled, we move x_{k+1} to determine next pixel. The corresponding y value for x_{k+1} position will be either y_k or $y_k - 1$ depending upon the sign of decision parameter. So the decision parameter for region 1 is tested at mid point of (x_k+1, y_k) and (x_k+1, y_k-1) i.e.

$$p_{1k} = f_{ellipse}(x_{k+1}, y_k - \frac{1}{2})$$

$$\text{or } p_{1k} = r_y^2 (x_{k+1})^2 + r_x^2 (y_k - \frac{1}{2})^2 - r_x^2 r_y^2$$

$$\text{or } p_{1k} = r_y^2 (x_{k+1})^2 + r_x^2 y_k^2 - r_x^2 y_k + \frac{r_x^2}{4} - r_x^2 r_y^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

if $p_{1k} < 0$, the mid point lies inside boundary, so next point to plot is (x_k+1, y_k)

otherwise, next point to plot will be (x_k+1, y_k-1)

The successive decision parameter is computed as

$$p_{1k+1} = f_{ellipse}(x_{k+1}+1, y_{k+1} - \frac{1}{2})$$

$$= r_y^{\ 2}(x_{k+1}+1)^2 + r_x^{\ 2}(y_{k+1}-\frac{1}{2})^2 - r_x^{\ 2}r_y^{\ 2}$$

or $$p_{1k+1} = r_y^{\ 2}(x_{k+1}^2 + 2x_{k+1} + 1) + r_x^{\ 2}(y_{k+1}^2 - y_{k+1} + \frac{1}{4}) - r_x^{\ 2}r_y^{\ 2}$$

or $$p_{1k+1} = r_y^{\ 2}x_{k+1}^2 + 2r_y^{\ 2}x_{k+1} + r_y^2 + r_x^{\ 2}y_{k+1}^2 - r_x^{\ 2}y_{k+1} + \frac{r_x^{\ 2}}{4} - r_x^{\ 2}r_y^{\ 2} \text{ ....} (2)$$

Subtracting (2) - (1)

$$p_{1k+1} - p_{1k} = 2r_y^{\ 2}x_{k+1} + r_y^2 + r_x^{\ 2}(y_{k+1}^2 - y_k^2) - r_x^2(y_{k+1} - y_k)$$

if $p_{1k} < 0$, $y_{k+1} = y_k$ then,

$$\therefore p_{1k+1} = p_{1k} + 2r_y^{\ 2}x_{k+1} + r_y^{\ 2}$$

Otherwise $y_{k+1} = y_k - 1$ then we get,

$$p_{1k+1} = p_{1k} + 2r_y^{\ 2}x_{k+1} + r_y^{\ 2} - 2r_x^{\ 2}y_{k+1}$$

At the initial position, $(0,r_y)$ $2r_y^{\ 2}x = 0$ and $2r_x^{\ 2}y = 2r_x^2 r_y$

In region 1, initial decision parameter is obtained by evaluating ellipse function at $(0,r_y)$ as

$$p_{10} = f_{ellipse}(1, r_y - \frac{1}{2})$$

or $$p_{10} = f_{ellipse}(1, r_y - \frac{1}{2})$$

$$= r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2$$

Similarly, over the region 2, the decision parameter is tested at mid point of $(x_k, y_k - 1)$ and $(x_k + 1, y_k - 1)$ i.e.

$$p_{2k} = f_{ellipse}(x_k + \frac{1}{2}, y_k - 1)$$

$$= r_y^2 (x_k + \frac{1}{2})^2 + r_x^2 (y_k - 1)^2 - r_x^2 r_y^2$$

$$\therefore p_{2k} = r_y^2 x_k^2 + r_y^2 x_k + \frac{r_y^2}{4} + r_x^2 (y_k - 1)^2 - r_x^2 r_y^2 \ldots\ldots\ldots\ldots\ldots(3)$$

if $p_{2k} > 0$, the mid point lies outside the boundary, so next point to plot is $(x_k, y_k - 1)$ otherwise, next point to plot will be $(x_k + 1, y_k - 1)$

The successive decision parameter is computed as evaluating ellipse function at mid point of

$$p_{2k+1} = f_{ellipse}(x_{k+1} + \frac{1}{2}, y_{k+1} - 1) \text{ with } y_{k+1} = y_k - 1$$

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

or $p_{2k+1} = r_y^2 x_{k+1}^2 + r_y^2 x_{k+1} + \frac{r_y^2}{4} + r_x^2 (y_k - 1)^2 - 2r_x^2 (y_k - 1) + r_x^2 - r_x^2 r_y^2$ ....(4)

subtracting (4)-(3)

$$p_{2k+1} - p_{2k} = r_y^2 (x_{k+1}^2 - x_k^2) + r_y^2 (x_{k+1} - x_k) - 2r_x^2 (y_k - 1) + r_x^2$$

or $p_{2k+1} = p_{2k} + r_y^2 (x_{k+1}^2 - x_k^2) + r_y^2 (x_{k+1} - x_k) - 2r_x^2 (y_k - 1) + r_x^2$

if $p_{2k} > 0$, $x_{k+1} = x_k$ then

$$p_{2k+1} = p_{2k} - 2r_x^2 (y_k - 1) + r_x^2$$

otherwise $x_{k+1} = x_k + 1$ then

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

or $p_{2k+1} = p_{2k} + r_y^2 (2x_k + 1) + r_y^2 - 2r_x^2 (y_k - 1) + r_x^2$

or $p_{2k+1} = p_{2k} + r_y^2(2x_k + 2) - 2r_x^2(y_k - 1) + r_x^2$

or $p_{2k+1} = p_{2k} + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_x^2$ where $x_{k+1} = x_k + 1$ and $y_{k+1} = y_k - 1$

The initial position for region 2 is taken as last position selected in region 1 say which is (x_0, y_0) then initial decision parameter in region 2 is obtained by evaluating ellipse function at mid point of (x_0, y_0-1) and (x_0+1, y_0-1) as

$$p_{20} = f_{ellipse}(x_0 + \frac{1}{2}, y_0 - 1)$$

$$= r_y^2(x_0 + \frac{1}{2})^2 + r_x^2(y_0 - 1)^2 - r_x^2 r_y^2$$

Mid-point ellipse Drawing algorithm

1. Input center (xc,yc) and r_x and r_y for the ellipse and obtain the first point as

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

2. Calculate initial decision parameter value in Region 1 as

   $$P_{10} = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^{\ 2}$$

3. At each x_k position, in Region 1, starting at $k = 0$, compute
   $x_{k+1} = x_k + 1$
   If $p_{1k} < 0$, then the next point to plot is
   $p_{1k+1} = p_{1k} + 2r_y^{\ 2} x_{k+1} + r_y^{\ 2}$
   $y_{k+1} = y_k$
   Otherwise next point to plot is
   $, y_{k+1} = y_k - 1$
   $p_{1k+1} = p_{1k} + 2r_y^{\ 2} x_{k+1} + r_y^{\ 2} - 2r_x^{\ 2} y_{k+1}$ with $x_{k+1} = x_k + 1$ and $y_{k+1} = y_k - 1$

4. Calculate the initial value of decision parameter at region 2 using last calculated point say (x_0, y_0) in region 1 as

$$p_{20} = r_y^2 (x_0 + \frac{1}{2})^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$$

5. At each y_k position in Region 2 starting at k = 0, perform computation

$y_{k+1} = y-1;$
if $p_{2k} > 0$, then
$x_{k+1} = x_k$
$p_{2k+1} = p_{2k} - 2r_x^2 (y_k - 1) + r_x^2$
Otherwise
$x_{k+1} = x_k + 1$
$p_{2k+1} = p_{2k} + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_x^2$ where $x_{k+1} = x_k + 1$ and $y_{k+1} = y_k - 1$

6. Determine the symmetry points in other 3 quadrants.

7. Move each calculated point (x_k, y_k) on to the centered (xc,yc) ellipse path as

$x_k = x_k + xc;$

$y_k = y_k + yc$

8. Repeat the process for region 1 until $2r_y^2 x_k \geq 2r_x^2 y_k$ and region until $(x_k, y_k) = (r_x, 0)$

End