paudelmuku@gmail.com

COMPUTER GRAPHICS

Disclaimer

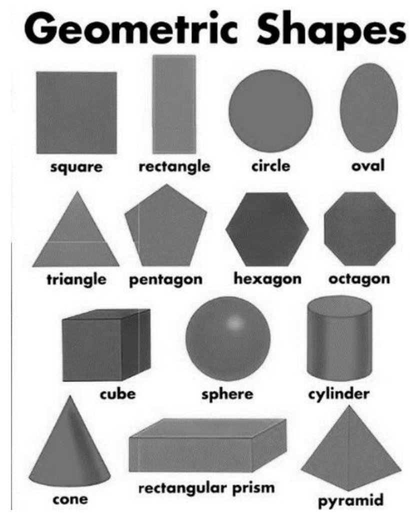
*This document is part of teaching materials for COMPUTER GRAPHICS under the Tribhuvan University syllabus for Bachelors of Arts in Computer Application (**BCA**). This document does not cover all aspect of learning COMPUTER GRAPHICS AND ANIMATION, nor are these be taken as primary source of information. As the core textbooks and reference books for learning the subject has already been specified and provided to the students, students are encouraged to learn from the original sources because this document cannot be used as a substitute for prescribed textbooks.*

*Various text books as well as freely available material from internet were consulted for preparing this document. Contents in This document are **copyrighted** to the instructor and authors of original texts where applicable.*

Prepared *By: Er. Mukunda Paudel*
paudelmuku@gmail.com

Chapter-1 Introduction

1.1 Advantages of Computer Graphics and Area of Application



Computer Graphics

Computer

Raw Data → process → Information

Graphics → Graph + pics

Graph → Geometrical shapes / mathematical figures

Pics→ image → how image behaves in different co-ordinates? OR how the image fits in screen of different display device?

- Firstly, provide raw data to the computer to make geometrical / mathematical figures.

- Provide raw data to the computer and by processing that raw data with the help of algorithms or programs we will get a beautiful / meaningful information i.e. Graph, bar, image etc. that is call computer graphics

Definition:

- ✓ CG is an art of drawing pictures, lines, charts or diagrams using computer with the help of programs (algorithms) and software.
- ✓ Technology that deals with designs and pictures on computers. So, computer graphics are visual representations of data displayed on a monitor made on a computer.

- ❖ Computer Graphics is a field related to the generation of graphics using computers.

- It includes the creation, storage, and manipulation of images of objects.

- These objects come from diverse fields such as physical, mathematical, engineering, architectural, abstract structures and natural phenomenon. Computer graphics today is largely interactive, i.e. the user controls the contents, structure, and appearance of images of the objects by using input devices, such as keyboard, mouse, or touch-sensitive panel on the screen.

In short, Computer graphics refer different things in different contexts:

- ✓ **Pictures**, scenes that are generated by a computer.
- ✓ **Tools** used to make such pictures, software and hardware, input/output devices.
- ✓ The **whole field of study** that involves these tools and the pictures they produce

Types of Computer Graphics:

Basically 2-Types

1. **Interactive Computer Graphics:**

    - Involves two way communication between computer and user.
    - User has full control over the content
    - In the Interactive Graphics, data / information shown in the display unit can be interact by one or more input device.
    - I/O Device→ request to system/ computer → O/P as a Graphical content.
    - Example: Simulators, User Interface etc.
    - Interactive computer graphics affects our lives in a number of indirect ways.

2. **Non-interactive Computer Graphics**

    - Involves one-way communication between user and computer /system.
    - User has control over some parts of the contents and totally controlled by program.
    - Also called **passive computer graphics**
    - Example: Videos, Images etc.

Computer Graphics vs Image Processing.

CG	IP
1. It is Field related to generation of pictures using computers (various algorithms for the generation of images)	1. It applies the techniques to modify or interpret existing pictures. (various techniques are used to manipulate pictures)
2. It includes creation storage and manipulation of images of objects	3. It is part of CG that handles image manipulation or interpretation
4. Synthesize pictures from mathematical or geometrical models.	5. Analyze pictures to drive descriptions (often in mathematical or geometrical forms) of objects appeared in the pictures
6. E.g. Drawing pictures	7. Making blurred image visible

History of Computer Graphics

At first only textual data was present → 1963 Ivan Sutherland developed a sketch pad in which images can be drawn on screen using light pen→ then US army → later he developed V-R Equipment → and he developed flight simulator → he was student of computer science.

Evaluation of Computer Graphics Can be summarized as follows:

In

1950 → First Graphics Images were created

1951 → CRT monitors on Main Frame computer were introduced

1959 → CAD was used to design cars

1961 → First video game named “Space War “developed

1963 → First Hidden Line and Hidden surface removal algorithms developed.

1965 → DDA algorithm developed by Jack Bresenham

1973 → First use of 2D animations

1982 →AutoCAD was released.

2001 → First digital film name "The Spirits Within "with digital actors

2006 → Google acquired sketch-up

2015 → "Big Data" used for constructing animation

2016 → With enough preparation, Real Time source can be animated

2020→ most video animation are CGI now, availability of highly advanced graphics.

Areas of Application of Computer Graphics *imp*

Widely used in numerous field in various way

1. **User Interface (UI)**

- CG is used to design : Menus, icons , cursors, dialog boxes, scrollbars, grids, 3D

- The interaction of users interacts with electronic devices such as computers mobiles and other devices. This interface uses icons, menus and other visuals, graphics by which user easily interacts.

- For example, Word processing, spreadsheet, and desktop-publishing programs are the typical examples where user-interface techniques are implemented.

2. **Plotting and presentation**

- Plotting bar charts histograms pie charts task scheduling charts are most commonly used plotting.
- Plotting is used to present data meaningfully and concisely
- Computer graphics are used for creating charts, bar diagrams and other visuals for the presentation purpose, with this user, can easily understand the points.
- Extensively used to plot and present 2D and 3D graphs of physical data, mathematical reports, scientific reports, economic and financial data/ functions and behavior.

3. **Office Automation and electronics publishing.**

- Provides a full set of word-processing features as well as fine control over placement of text and graphics, so that you can create newsletters, advertisements, books, and other types of documents.
- It means by using a personal computer or workstation high-quality printed documents can be produced.
- Electronic publishing / desktop publishing system allows you more power to print the meaningful materials in-house.
- The most powerful desktop publishing systems enable you to create illustrations. While less powerful systems let you insert illustrations created by other programs.
- 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
- A particularly important feature of desktop publishing systems is that they enable you to see on the display screen exactly how the document will appear when printed. Systems that support this feature are called WYSIWYGs (what you see is what you get).
- Desktop publishing systems have become increasingly popular for producing newsletters, brochures, books, and other documents that formerly required a typesetter.

4. **Computer aided Design**

- Computer-aided design (CAD) is the use of computer technology for the design of objects, real or virtual.
- 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.

- The design of geometric models for object shapes, in particular, is often called computer-aided geometric design (CAGD).
- CAD may be used to design curves and figures in two-dimensional ("2D") space; or curves, surfaces, or solids in three-dimensional ("3D") objects.
- CAD is also widely used to produce computer animation for special effects in movies, advertising, technical manuals.

5. **Digital Art**:

- Digital art most commonly refers to art created on a computer in digital form.
- Computer Graphics provide a new way of creating designs. Now Artists and designers use illustrator, coral draw, Photoshop, adobe muse and other different types of applications for creating new designs.
- The impact of digital technology has transformed traditional activities such as painting, drawing and sculpture, while new forms, such as net art, digital installation art, and virtual reality, have been recognized artistic practices.

6. **Web Design**:

- Web design is the skill of designing presentations of content usually hypertext or hypermedia that is delivered to an end-user through the World Wide Web, by way of a Web browser.
- The process of designing Web pages, Web sites, Web applications or multimedia for the Web may utilize multiple disciplines, such as animation, authoring, communication design, corporate identity, graphic design, human-computer interaction, information architecture, interaction design, marketing, photography, search engine optimization and typography.

7. **Education and Training**

- For the educational or training purpose – A computer simulation, a computer model or a computational model can be used. Which is a computer program, or network of computers, that attempts to simulate an abstract model of a particular system.

- Computer simulations have become a useful part of mathematical modeling of many
  a) natural systems in physics (computational physics), chemistry and biology,
  b) Human systems in economics, psychology, and
  c) Social science and in the process of engineering new technology, to gain insight into the operation of those systems, or to observe their behavior.

i. Computational Biology:

- Computational biology is an interdisciplinary field that applies the techniques of computer science, applied mathematics and statistics to address biological problems.
- The main focus lies on developing mathematical modeling and computational simulation techniques in biological sector.

ii. Computational Physics:

- Computational physics is the study and implementation of numerical algorithm to solve problems in physics for which a quantitative theory already exists.
- It is often regarded as a sub discipline of theoretical physics but some consider it an intermediate branch between theoretical and experimental physics.

iii. Information of Graphics:

- Information graphics or information graphics are visual representations of information, data or knowledge.
- These graphics are used where complex information needs to be explained quickly and clearly, such as in signs, maps, journalism, technical writing, and education.
- They are also used extensively as tools by computer scientists, mathematicians, and statisticians to ease the process of developing and communicating conceptual information.

iv. Scientific and business Visualization:

- Scientific visualization is a branch of science, concerned with the visualization of three dimensional phenomena, such as architectural, meteorological, medical, biological systems.
- 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

- Scientific visualization focuses on the use of computer graphics to create visual images which aid in understanding of complex, often massive numerical representation of scientific concepts or results.

8. Simulation and modeling

- Simulation is the imitation of the conditions like those, which is encountered in real life. Simulation thus 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

9. Entertainment

- Disney movies such as Lion Kings and The Beauty of Beast, and other scientific movies like Jurassic Park, Avatar, 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, PUBG, Doom, Pools are few to name where graphics is used extensively.

10. 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 / Disadvantages of Computer Graphics

- ❖ Computer graphics provides the tool to create not only real-world objects but also objects that are abstract in nature such as 4D mathematical models.
- ❖ There are many applications and advantages of computer graphics in real world. Some of the main advantages of computer graphics are mentioned below

S.N.	Advantages	Disadvantages
1.	Enhancing Visual Communication	Requiring specialized skill and software i.e. complexity
2.	Facilitating the creation of complex design	Potentially leading to unrealistic or exaggerated representations
3.	Improving the realism of visual content	Taking time and high initial investment to create
4.	Enabling the creation of 3D models and animations	Potentially promoting a reliance on technology
5.	Improving the accessibility of visual content	Being vulnerable to manipulation

COMPONENTS OF COMPUTER GRAPHICS

- Interactive computer Graphics consist of three components
    - ✓ Digital buffer,
    - ✓ TV monitor and
    - ✓ Display controller.
- Using these components, we are able to see the output on the screen in form of pixels (picture elements).
- Following is the explanation of these components

1. **Digital Memory Buffer**

    - This is a place where images and pictures are stored as an array (matrix of 0 & 1, 0 represents darkness and 1 represents image or picture).

    - Digital memory buffer also called **frame buffer**.

    - In today's term frame buffer is called V-RAM (video RAM) and it helps to store the image in bit form. It helps to increase the speed of graphics (sometimes we watch movies on our computer system and movie run slowly. System engineer is then called for. He/She comes and fits in V-RAM (in Megabytes) into our system and movie runs perfectly.

2. **TV Monitor**

    - Monitor helps us to view the display and they make use of CRT technology (Cathode ray Tube).

3. **Display Controller**
    - It is an interface between Memory Buffer and TV Monitor.

    - Its job is to pass the contents of frame buffer to the monitor. This passing has to be fast for steady display on the monitor (depending upon the material of the system). The image must be passed repeatedly to the monitor to maintain a steady picture on the screen.

- The display controller reads each successive byte of data from FB Memory and converts 0's and 1's into corresponding video signals. This signal is then feed to the TV monitor to produce a black and white picture on screen.

- In today's term, display controller is recognized as a display card and one of our choices can be VGA card with a resolution of 640x480. (Display Controller is also capable of displaying image in colors).

1.2 Hardware and Software for Computer Graphics

Must know CG Terminologies?

1. **Pixel**

   - Pix + el = picture element
   - Fundamental building block of any image or computer graphics
   - Defined as addressable tiny objects or color spot (dots) displayed on screen, which forms an image or
   - Image is a collection of pixels.

2. **Resolution**

   - Resolution is defined as total number of pixels in our digital screen
   - I.e. Resolution = no. of pixels in X direction * no. of pixels in Y direction.

3. **Aspect ratio**

   - Screen's aspect ratio refers to the ratio of the no. of X (horizontal) pixel to the no of Y (vertical) pixels i.e. Aspect Ratio = total no of X (horizontal) pixels / total no of Y (vertical) pixel
   - Example : 800*600 Resolution = 480000 No. of pixels = 4:3 Aspect ratio

4. **Persistence**

- Basically persistence means how much time is taken by the emitted light to reduce to one-tenth ($1/10^{th}$ or 10%) of its original intensity.
- Different kinds of phosphors are used in a CRT. The difference is based upon the time for how long the phosphor continues to emit light after the CRT beam has been removed. This property is referred to as **Persistence.**
- Now, phosphors with lower persistence require higher refresh rates to maintain a picture on the screen without any flicker.
- Graphics monitor are usually constructed with persistence in the range from 10 to 60 micro second.

5. Refresh Rate

- How many times per second a screen is refreshed by electron beam is known as its refresh rate.
- It's measured in Hertz (Hz), the unit of frequency.
- For a good quality of display device it is suggested that it should have higher resolution & refresh rate.
- Alternatively referred to as **frame rate**, **horizontal scan rate**, **vertical frequency**, or **frequency**. **Refresh rate** is a CRT monitor measurement in Hz that indicates how many times per second a monitor screen image is renewed. For example, a monitor with a refresh rate to 75 Hz means the screen is going to redraw 75 times per second.
- Usually 50/sec
- Refresh rate above which flickering stops is called Critical Fusion Frequency (CFF)
- The factor affective CFF are persistence, image intensity, ambient room light, Wave length of emitted light, observer.
- An older refresh rate standard, developed by the Video Electronics Standards Association (VESA Local Bus), was only 60 Hz. This refresh rate caused the display's image to flicker, causing eye fatigue and headaches in users. A new standard set the refresh rate to 75 Hz. **It is believed that 70 Hz or higher eliminates the flicker**. When purchasing a monitor, look for a refresh rate of 75 to 85 Hz.
- Finally, an LCD does not have a refresh rate

6. Aliasing

- Aliasing is distortion that appear in any display system when the sampling the continuous object to discrete integer pixel position.
- This is because, lines, polygon, circle etc. are continuous but a raster device is discrete.

7. Bit Map and Pixel Map

- If a pixel has only two-color values (i.e. black and white), it can be encoded by a 1 bit of information. On a black and white system with one bit per pixel, the frame buffer is called bitmap.
- An image of more than two colors is called pix map.

8. **Bit Depth** (**Or Color Depth**)
   - It is defined as number of bits assigned to each pixel in the image.

9. **Phosphorescence**
   - It is a process in which energy absorbed by a substance is released relatively slowly in the form of light.

Display Devices and Hard Copy Devices (video display devices)

- Computer Graphics has become a common element in today's modern world.
- Be it in user interfaces, or data visualization, motion pictures etc, computer graphics plays an important role.
- The primary output device in a graphics system is a video monitor. Although many technologies exist, but the operation of most video monitors is based on the standard Cathode Ray Tube (CRT) design.

Monitors, commonly called as **Visual Display Unit** (VDU), are the main output device of a computer. It forms images from tiny dots, called pixels that are arranged in a rectangular form. The sharpness of the image depends upon the number of pixels.

There are two kinds of viewing screen used for monitors.

1. Cathode-Ray Tube (CRT)
2. Flat-Panel Display (FPD)

Cathode Ray Tube (CRT)

- A cathode ray tube (CRT) is a specialized vacuum tube in which images are produced when an electron beam strikes a phosphorescent surface.
- It modulates, accelerates, and deflects electron beam(s) onto the screen to create the images.
- Most desktop computer displays make use of CRT for image displaying purposes.

The primary components are

A) **heated metal**
B) **Cathode** and
C) **Control grid**.

Working

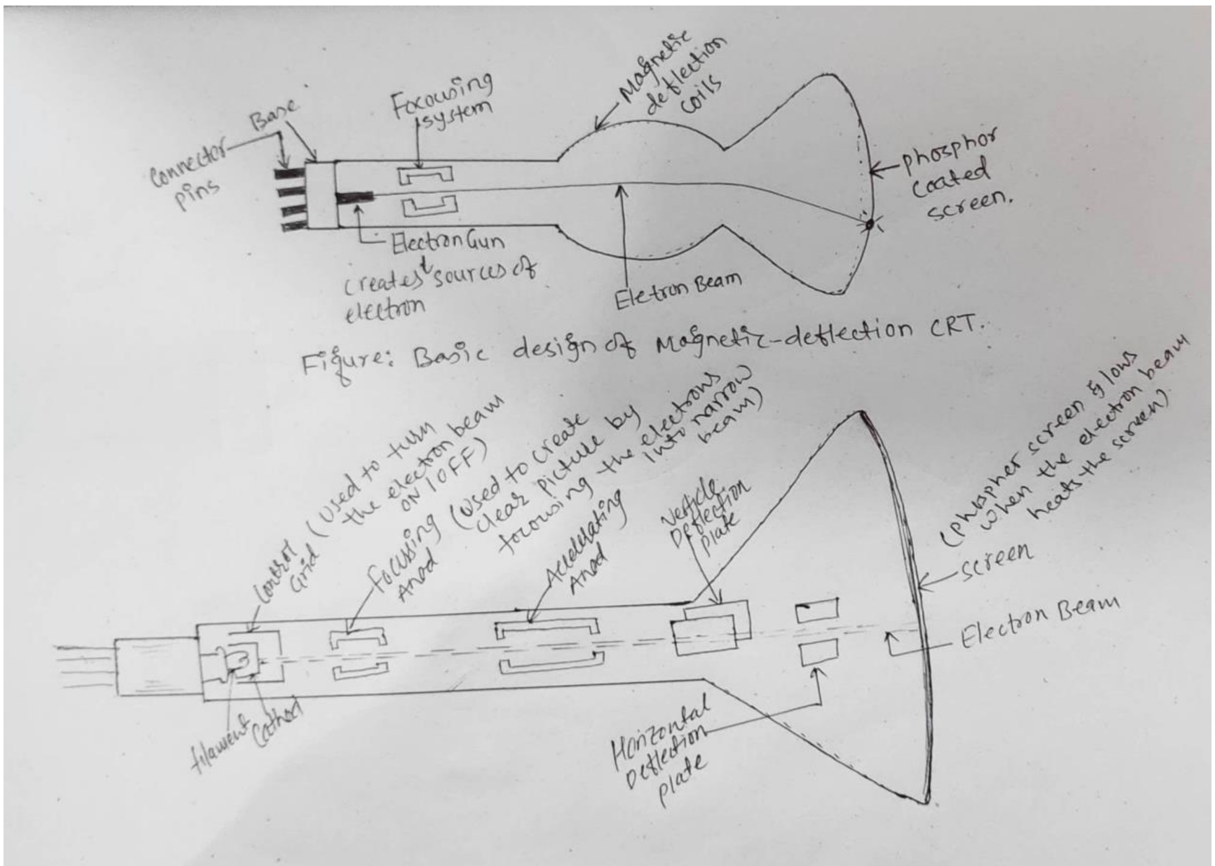


Figure: Basic design of Magnetic-deflection CRT.

1. The heat is supplied to the cathode (by passing current through the filament). This way the electrons get heated up and start getting ejected out of the cathode filament.
2. This stream of negatively charged electrons is accelerated towards the phosphor screen by supplying a high positive voltage.
3. This acceleration is generally produced by means of an accelerating anode.
4. Next component is the **Focusing System**, which is used to force the electron beam to converge to small spot on the screen.

5. If there will not be any focusing system, the electrons will be scattered because of their own repulsions and hence we won't get a sharp image of the object.
6. This focusing can be either by means of electrostatic fields or magnetic fields.

The deflecting system is in charge of deflecting an electron beam in response to input signals to produce a pattern. When an electron beam is accelerated, it passes through a deflection system, allowing it to be positioned anywhere on the screen. The cathode-ray-tube deflection system is made up of two pairs of parallel plates known as the vertical and horizontal deflection plates

Flat Panel Display

- **Flat-Panel Devices** are the devices that have less volume, weight, and power consumption compared to Cathode Ray Tube (CRT).
- Due to the advantages of the Flat-Panel Display, use of CRT decreased.
- As Flat Panel Devices are light in weights that's why they can be hang on walls and wear them on our wrist as a watch.
- Flat Panel Display (FPD) allow users to view data, graphics, text and images.
- **Example:** Small T.V. monitor, calculator, pocket video games, laptop computers, an advertisement board in elevator.

Types of Flat panel display:

1. **Emissive Display:**

   The Emissive Display or Emitters are the devices that convert electrical energy into light energy.

   Examples: Plasma Panel, LED (Light Emitting Diode), Flat CRT.

2. **Non-Emissive Display:**

   Non-Emissive Display or Non-Emitters are the devices that use optical effects to convert sunlight or some other source into graphic patterns.

Examples: LCD (Liquid Crystal Display)

Advantages of Flat Panel Devices:

- Flat Panel Devices like LCD produces high quality digital images.
- Flat Panel monitor are stylish and have very space saving design.
- Flat Panel Devices consumes less power and give maximum image size in minimum space.
- Flat Panel Devices use its full color display capability.
- Full motion video can be viewed on Flat Panel Devices without artifacts or contrast loss.

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

1. **Raster / Refresh Scan Display**

2. **Random / Vector Scan Display**

Raster and Random Systems and Architectures

- Raster and Vector Graphics A raster image is made of up pixels, each a different color, arranged to display an image
- Vector image is made up of paths, each with a mathematical formula (vector) that tells the path how it is shaped and what color it is bordered with or filled by.

Raster Graphics	**Vector Graphics**
Raster Graphics are composed of pixels	Vector graphics are composed of paths.
Raster image pixels do not keep on their appearance as size increases - when you blow a photograph up, it becomes blurry for this reason	Vector images keep on appearance regardless of size, since the mathematical formulas dictate how the image is rendered
Normally they have the file extension of .gif, .jpg	They have the extension of .eps

Refresh or Raster Scan Display System

- A raster consists of pixels organized into rows and columns (or a grid) where each cell contains a value of information. In this display system, raster points are used as basic drawing primitives.
- Home television (CRT) and printers are example of systems using raster scan method

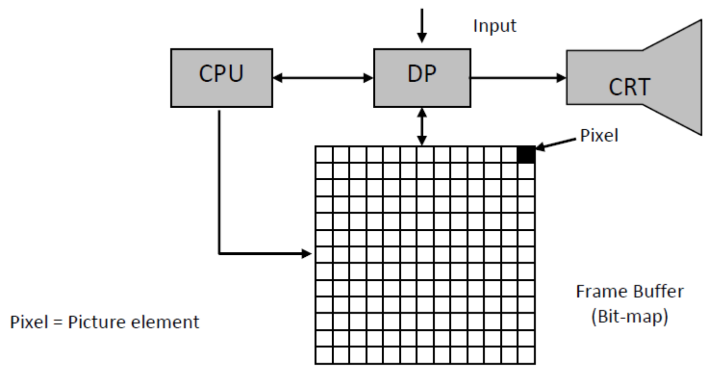


Figure: Raster Scan display system

- 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.

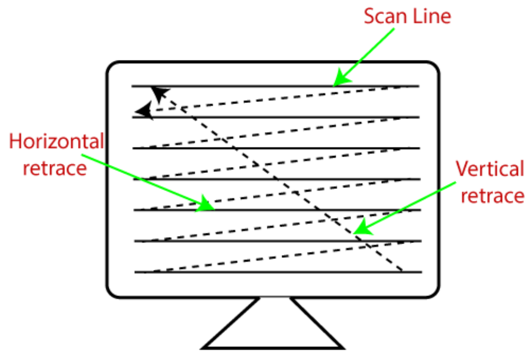
- Refreshing on raster-scan displays is carried out at the rate of 60 to 80 frames per second. Refreshing must be done because light emitted by phosphor fades very rapidly, so to keep the drawn picture glowing constantly, it is required to redraw the picture repeatedly and quickly directing the electron beam back over some point.
- The no of times/sec the image is redrawn to give a feeling of non-flickering pictures is called refresh-rate

Each scan line in display device consists of two retrace

Prepared By: Er. Mukunda Paudel
paudelmuku@gmail.com

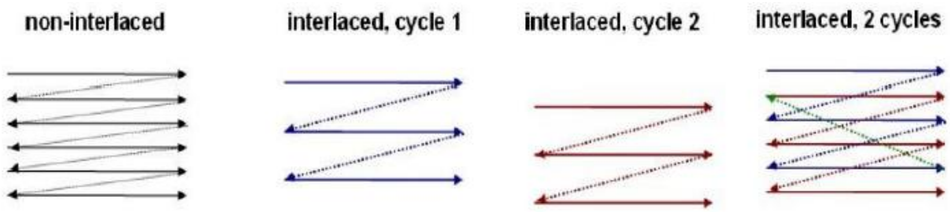
I) Retrace procedure

- At the end of each scan line in raster scan display, the electron beam returns to the left side of the screen to begin displaying the next scan line.
- The return to the left of the screen, after refreshing each scan line is called the **horizontal retrace** of the electron beam. And at the end of each frame the electron beam returns to the top left corner of the screen to begin the next frame which is called **vertical retrace**.



II) Interlaced refresh procedure

- On some raster scan systems each frame is displayed in two passes using an interlaced refresh procedure so that the whole picture should displaced in half time.
- Here, the first scan does the even lines 0, 2, 4...
- Then the second scan does the odd lines 1, 3, 5,



Architecture of Raster graphics System

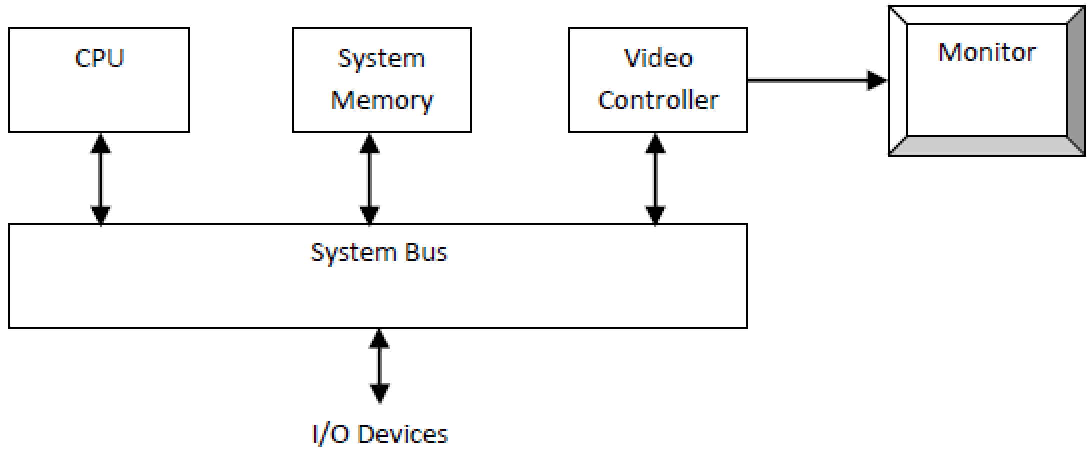


Figure: Architecture of simple raster graphics system.

- ❖ Raster graphics systems having additional processing unit like video controller or display controller.
- ❖ Here frame buffer can be anywhere in the system memory and video controller access this for refresh the screen.
- ❖ In addition to video controller more processors are used as co-processors to accelerate the system in sophisticated raster system.

<u>Raster graphics system with a fixed portion of the system memory reserved for the frame buffer</u>

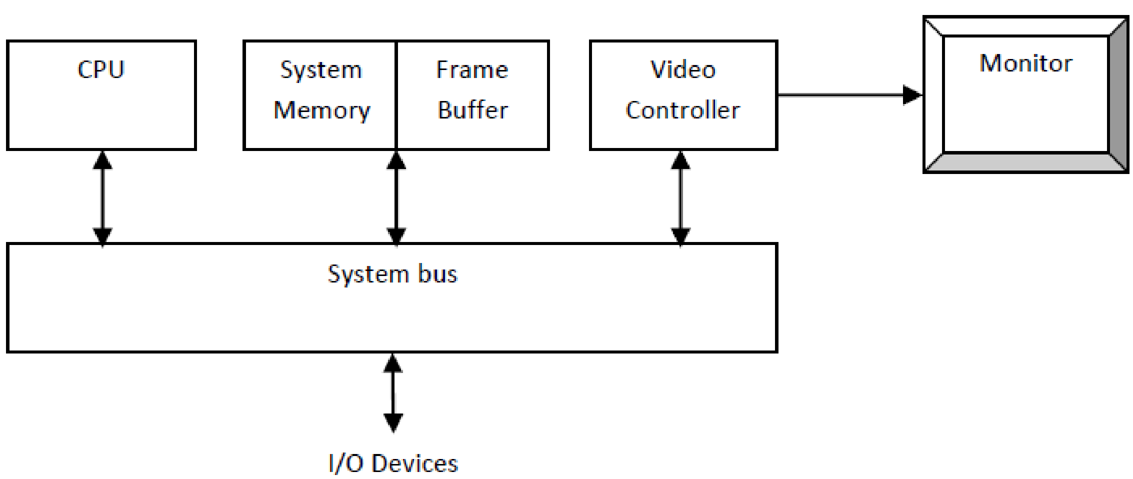


Figure: Architecture of a raster graphics system with a fixed portion of the system memory reserved for the frame buffer

- A fixed area of the system memory is reserved for the frame buffer and the video controller can directly access that frame buffer memory.

- Frame buffer location and the screen position are referred in Cartesian coordinates.

- For many graphics monitors the coordinate origin is defined at the lower left screen corner.

- Screen surface is then represented as the first quadrant of the two dimensional systems with positive X_{value} increases as left to right and positive Y_{value} increases bottom to top.

Random / Vector Scan Display System

- 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.

Architecture of Random Scan Display system

paudelmuku@gmail.com

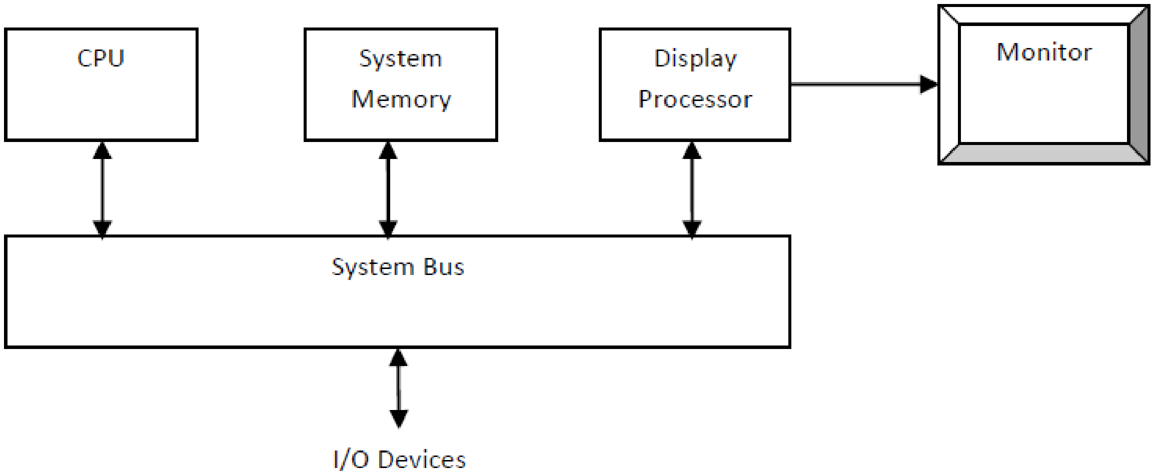


Figure: Architecture of a simple random-scan system

- An application program is input & stored in the system memory along with a graphics package.
- Graphics commands in the application program are translated by the graphics package into a display file stored in the system memory.
- This display file is used by display processor to refresh the screen.
- Display process goes through each command in display file. Once during every refresh cycle.
- Sometimes the display processor in random scan system is also known as display processing unit or a graphics controller.
- In this system graphics platform are drawn on random scan system by directing the electron beam along the component times of the picture.
- Lines are defined by coordinate end points.
- This input coordinate values are converts to X and Y deflection voltages.
- A scene is then drawn one line at a time

2.4 Video Controller

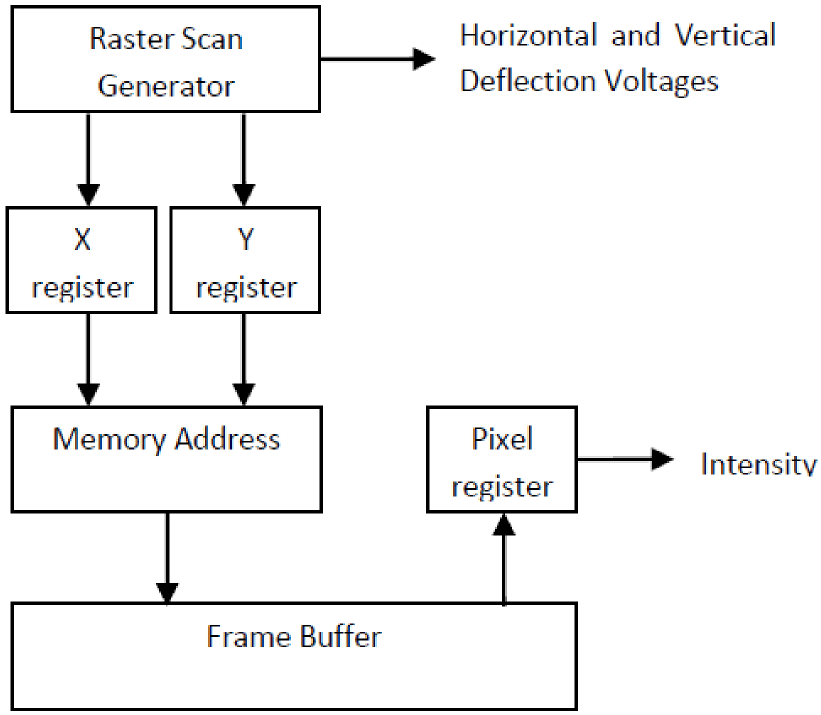


Figure: basic video controller refresh operation.

- ❖ Two registers are used to store the coordinates of the screen pixels which are X and Y as shown in above figure.
- ❖ 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 Xmax and in Y from 0 to Ymax 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).
- ❖ As first scan line is generated, the X address is incremented up to Xmax. 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=Ymax) is generated
- ❖ Since screen must be refreshed at the rate of 60 frames per second the simple procedure illustrated in figure cannot be accommodated by typical RAM chips.

- To speed up pixel processing video controller retrieves multiple values at a time using more numbers of registers and simultaneously refresh block of pixel.
- Such a way it can speed up and accommodate refresh rate more than 60 frames per second.

Raster (Refresh) scan vs Vector (Random) Scan display system

Raster Scan	Random Scan	Remarks
The electron beam is swept across the screen one row at a time from top to bottom	The electron beam is swept to the parts of the screen where a picture is to be drawn.	*Electron Beam*
It has lower or poor resolution because picture definition is stored as an intensity value.	It has high resolution because it stores picture definition as a set of line commands	*Resolution*
Picture definition is stored as a set of intensity values for all screen points (pixels) in a refresh buffer.	Picture definition is stored as a set of line in a display list or file.	*Picture definition*
The capacity of the system to store intensity values for pixels make it well suited for realistic display with shadow and color pattern.	These system are designed for line-drawing and can't display realistic shaded scenes.	*Realistic display*
Screen points or pixels are used to draw an image.	Mathematical functions are used to draw an image	*Image drawing*
They are cheaper than random display.	It is more expensive than raster-scan display.	*cost*
Refresh rate Refresh rate is 60-80 fps.	All components are drawn—30 to 60 times per second.	*Refresh Rate*

Interlacing It uses interlacing.	It doesn't use interlacing	*Interlacing*
Editing Editing is difficult	Editing is easy	*Editing*
Refresh area Refresh area is independent of picture complexity.	Refresh area depends on complexity of picture	*Refresh area*
Smoothness Produce jagged line.	Produce smooth line.	*Smoothness*
CRT, TV, Printer	Pen Plotter	*Example*

Use of Digital to Analog Converter and Frame Buffer Organization

- ❖ A frame buffer is a large, contiguous piece of computer memory into which the intensive values for all pixels are placed.
- ❖ At a minimum there is one memory bit for each pixel, this amount of memory is called a bit plane.
- ❖ The picture is built up in the frame buffer one bit at a time.
- ❖ We know that a memory bit has only two states, therefore a single bit plane yields a black-and white display.
- ❖ We know that a frame buffer is a digital device and the CRT is an analog device. Therefore, a conversion from a digital representation to an analog signal must take place when information is read from the frame buffer and displayed on the raster CRT graphics device.
- ❖ For this conversion, digital to analog converter (DAC) can be used.
- ❖ Each pixel in the frame buffer must be accessed and converted before it is visible on the raster CRT.

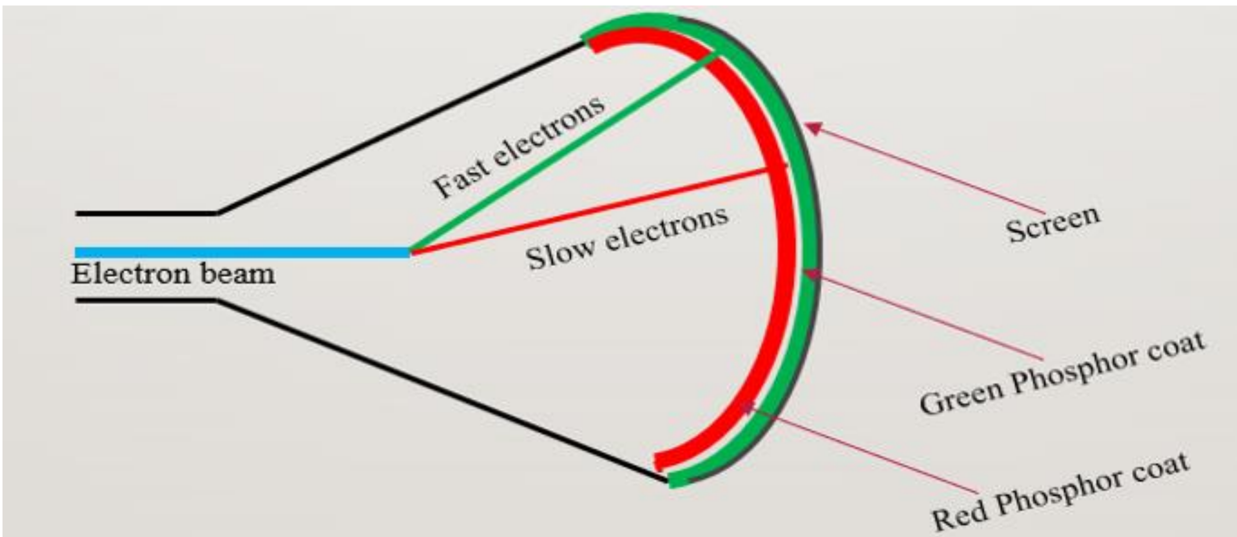
Color Monitors

The basic principle behind colored displays is that combining the 3 basic colors – Red, Blue and Green, can produce every color. By choosing different ratios of these three colors we can produce different colors – millions of them in-fact. We also have basic phosphors, which can produce these basic colors. So, one should have a technology to combine them in different combinations.

There are two popular techniques for producing color displays with a CRT are:

1. **Beam Penetration Method:**

- ❖ This method is for random scan monitor display where two different layers of phosphor coating are used
- ❖ Red (outer) and Green (Inner) coated on the CRT screen.
- ❖ In this method only four colors are possible and hence-the poor picture quality.



Working:

- ❖ A beam of slow electrons excites only the outer red layer, then a beam of very fast electrons penetrates through the red phosphor and excites the inner green layer.

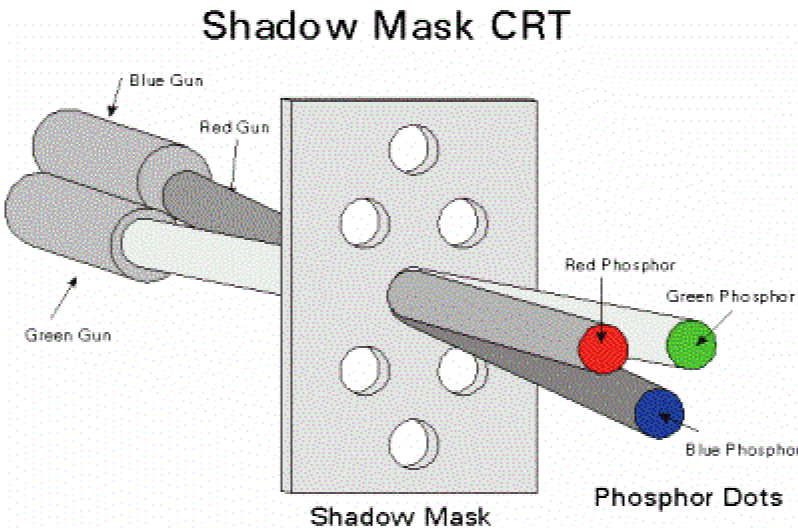
- ❖ Intermediate is a combination of ted and green so, two additional colors arrange and yellow color

- ❖ When quantity of red is more than green then orange color appears

- ❖ When quantity of green is more than red color then yellow color appears

2. Shadow Mask Method:

paudelmuku@gmail.com



- The Shadow mask CRT, instead of using one electron gun, uses 3 different guns placed one by the side of other to form a triangle or delta structure.

- Each pixel point on the screen is also made up of 3 type of phosphors to produce red, blue and green colors

- The phosphor screen is a metal screen called shadow mask. This plate has holes placed strategically (in such a technical way), so that when the beams from the three electron guns are focused on a particular pixel, they get focused on particular color producing pixel only.

Scan Conversion Algorithms (Line, Circle, Ellipse)

Numerical Session:

1. **There is a system with 24 bits per pixel and resolution of 1024 by 1024. Calculate the size of frame buffer (in Megabytes)**

   Solution:
   Resolution = 1024 * 1024
   Total number of pixel = 1024 * 1024 = 1048576 pixels
   Bits per pixels storage = 24 bits

Therefore, total storage required in frame buffer = 1048576 * 24 = 25165824bits

=25165824 / 8 Byte
= 25165824 / (8 * 1024) Kb
= 25165824 / (8 * 1024 * 1024) Mb
= 3 Mb

2. **How Many k bytes does a frame buffer needs in a 600 x 400 pixel?**

Suppose, n bits are required to store 1 pixel.
Then, the size of frame buffer = Resolution * bits per pixel = (600 * 400) * n bits

= 240000 n bits
= 240000 n kb / (8 * 1024)
= 29.30 n k bytes.

3. **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
Pixels Per inch (Resolution) = 100
Then total number of pixels = 8 * 100 * 10 * 100 = 800000 pixels
Bits per pixels' storage =8
Therefore, total storage required in frame buffer = 800000 * 8 bits

= 6400000bits
= 6400000/8 bytes
= 800000Bytes

4. **Find out the aspect ratio of the raster system using 8 x 10 inches' screen and 100 pixels/inch.**

Solution:
We know that,

Aspect ratio = Width / Height
= (8 x 100) / (10 * 100)
= 4 / 5
So, aspect ratio = 4: 5 5.

5. **What is the time required to display a pixel on the monitor of size 1024 * 768 with refresh rate of 60 Hz?**

Solution:
Refresh Rate = 60Hz → 60 frames per second
Total number of pixel in one frame = 1024 * 768 = 786432 pixels.
60 frames need 1 second
1 frame need 1 / 60 second → 786432 pixels need 1/ 60 second
→ 1 pixels need 1 / (60 * 786432) second
→ 10^9 / (60 * 786432) ns
→ 21.19 ns

6. **If the total intensity achievable for a pixel is 256 and the screen resolution is 640 * 480. What will be the size of frame buffer?**

Solution:
1 pixel = 256 different intensity level
Let,
x be the number of bits required to represent 256 different intensity level
Then, $2^X = 256$
Therefore, x= 8 bits
Resolution = 640 * 480
Hence, number of bits required for the screen = 640 * 480 * 8
= 2457600 bits.

7. **If a pixel is accessed from the frame buffer with an average access time of 300ns then will this rate produce an un-flickering effect for the screen size of 640 * 480?**

Solution: Size of screen = 640 * 480
Total Number of pixels = 640 * 480 = 307200
Average access time of one pixel = 300ns
Therefore,
Total time required to access entire pixels of image in the screen = 307200 * 300 = 92160000 ns
= $92160000/10^9$
= 0.09216 seconds

i.e. 1 cycle take 0.09216 second

Now, Number of cycles per second i.e. Refresh Rate =?

0.09216 seconds = 1 cycle

1 second = 1/ 0.09216 = 10.86

Therefore, Refresh Rate= 10.86 cycles per second

Since the minimum refresh rate for unflicker image is 60 frames per second, hence we can say the monitor produces flickering effect.

8. **Consider a raster scan system having 12-inch by 12-inch screen with a resolution of 100 pixels per inch in each direction. If display controller of this system refreshes the screen at the rate of 50 frames per second, how many pixels could be accessed per second and what is the access time per pixel of the system.**

   Solution: Size of Screen = 12 inch * 12 inches
   Resolution = 100 pixels per inch
   Therefore, total number of pixels in one frame = 12 * 100 * 12 * 100
   Refresh Rate = 50 frames per second → 50 frames can be accessed in 1 second
   Therefore,
   Total number of pixels accessed in 1 second = 50 * 12 * 100 * 12 * 100 = 72000000 pixels.
   Again,
   Since, 50 frames can be accessed in 1 second
   1 frame can be accessed in 1/50 second
   (12 * 100 * 12 * 100) pixels can be accessed in 1/50 second
   Then 1 pixel can be accessed in 1 / (50 * 12 * 100 * 12 * 100) second
   = 10^9 / (50 * 12 * 100 * 12 * 100) ns
   =13.88 ns

Hence, Access time per pixel = 13.88ns/pixel.

9. **How much time is spent scanning across each row of pixels during screen refresh on a raster system with resolution 1280 * 1024 and refresh rate of 60 frames per second?**

   Solution:
   Resolution = 1280 * 1024
   i.e. One frame contains 1024 scan line and each scan line consists of 1280 pixels.
   Refresh Rate = 60 frames per second
   i.e. 60 frame take 1 second
   1 frame take 1/60 second
   i.e. 1024 scan line take 1/60 second i.e. 0.0166 second
   1 scan line take 0.0166/1024
   = 0.016 Ms

paudelmuku@gmail.com

1.5 Scan Conversion Algorithms (Line, Circle, Ellipse)

Output Primitives:

Initial phase of computer science→ computing and textual form printing

Later on → graphics evolved→ drawing of different objects like line, circle, ellipse, rectangle, spline curve etc.

Hence, these graphical components are called output primitives.

Output primitives are the geometric structure such as straight-line segments and polygon areas used to describe the shapes and color of the objects.

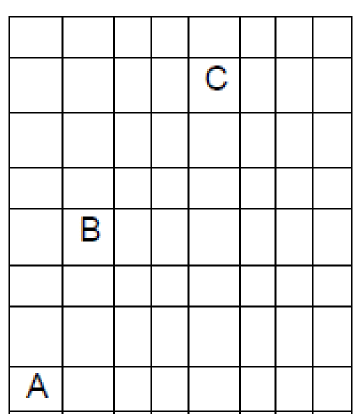
In case of the two-dimensional algorithm, we mostly deal with the line, circle, ellipse etc. as they are the basic output primitives.

Scan Converting a Point and a Straight Line

Point:

- ❖ Pixel is a unit square area identified by the coordinate of its lower left corner.
- ❖ Each pixel on the display surface has a finite size depending on the screen resolution & hence a pixel can't represent a single mathematical number.
- ❖ Origin of the reference coordinate system being located of the lower left corner of the display surface.
- ❖ The each pixel is accused by non-negative integer coordinate pair(x, y).

The x values start at the origin &increase from left to right along a scan line & y values start at the bottom & increase upwards.



- ❖ In the above diagram the coordinate of pixel A:0,0 ,B:1,4 , C:4,7.C:4,7.

- A coding position (4. 2, 7. 2) is represented by C whereas (1.5, 4.2) is represented by B.
- In order to half a pixel on the screen we need to round off the coordinate to a nearest integer.

Line:

- Line drawing is accomplished by calculating intermediate point coordinates along the line path between two given end points.
- Screen pixel are referred with integer values, plotted positions may only approximate the calculate coordinates, what is pixel which are intensified are those which lie very close to the line path.
- In a high resolution system the adjacent pixels are so closely spread that the approximated line pixels lie very close to actual line path and hence the plotted lines appear to be much smooth-almost like straight line drown on paper.
- In low resolution system the same approximation technique causes to display with stair step appearance that is not smooth.

Line Drawing Algorithm

The equation of a straight line is Y=mX + b

Where **m** representing slope of the line and **b** as the y intercept

.

Consider the two end points of a line segment are (x_1, y_1) & (x_2, y_2) then the straight line can be written as

$$y_1 = \frac{y_2 - y_1}{x_2 - x_1} x_1 + b \quad \text{………………….. (1)}$$

We can determine the volume for the slope 'm' & y intercept 'b' with the following calculation.

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$b = y_1 - mx_1$$

$$\Rightarrow y_1 = mx_1 + b$$

For any given x interval Δx along a line, we can compute the corresponding y interval Δy from the above equation

$$m = \frac{\Delta y}{\Delta x} \Rightarrow \Delta y = m\Delta x$$

Similarly, we can obtained the x interval Δx corresponding to a specified Δy as

$$\Delta x = \frac{\Delta y}{m}$$

For a line with slope magnitude |m|<1,

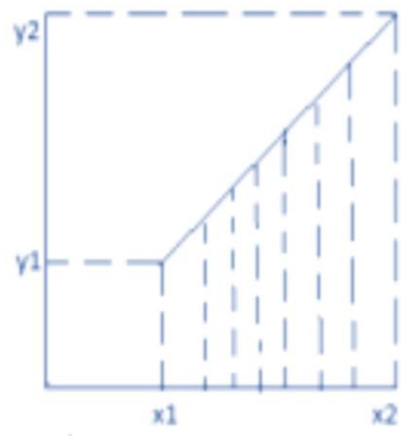
Δx can be set proportional to a small horizontal deflection voltage & the corresponding vertical deflection is then set proportional to Δy as calculate from the equation

$$\Delta y = m\,\Delta x$$

For a line whose slopes have magnitudes |m|>1,

Δy can be set proportional to a small vertical deflection voltage with the corresponding horizontal deflection voltage set proportional to Δx calculate from the equation

$$\Delta x = \frac{\Delta y}{m}$$



For a line with m=1,

Then $\Delta x = \Delta y$ and vertical & horizontal deflection voltages are equal.

In each case a smooth line with slope 'm' is generated between specified end point.

DDA Algorithm (Digital Differential Analyzer)

- DDA is a scan-conversion line algorithm base on either Δx or Δy.

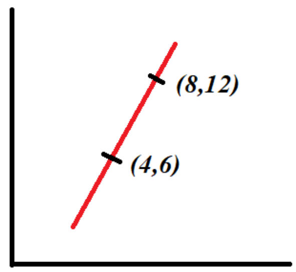
The process of representing continuous graphics objects as a collection of discrete pixels is called ***scan conversion***. The graphics objects are continuous. The pixels used are discrete.

- We sample the line at unit interval in one direction (x if △x is greater than △y, otherwise in y direction) and determine the corresponding integer values nearest the line path for the other coordinate

Algorithm:

Consider the line having positive slope

paudelmuku@gmail.com



The equation of the line is given,

$$y = mx+b \text{ ....(i)}$$

$$m = (y2-y1)/(x2-x1) \text{ ....(ii)}$$

For any interval $\triangle x$, corresponding interval is given by $\triangle y = m.\triangle x$.

Case I: If $|m| \leq 1$, we sample at unit x interval

i.e $\triangle x = 1$.

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

Then we compute each successive y-values, by setting $\triangle y = m$

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

The calculated y value must be rounded to the nearest integer

Case II: If $|m| > 1$, we sample at unit y-interval

i.e $\triangle y = 1$ and compute each successive x-values.

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

Therefore, $1 = m.\triangle x$, and $\triangle x = 1/m$ (since, $m=\triangle y/\triangle x$ and $\triangle y=1$).

$$x_{k+1} = x_k + 1/m$$

The above equations are under the assumption that lines are to be processed from left endpoints (x_k, y_k) to right endpoints (x_{k+1}, y_{k+1}).

Advantages

- DDA algorithm is a faster method for calculating pixel position then the equation of a pixel position. Y = mx + b
- Simplex and fast method

Disadvantages

- Accumulation of round of error is successive addition of the floating point increments is used to find the pixel position but it take lot of time to compute the pixel position.
- Poor end point accuracy

Problem: 01: Digitized the line with end points (0, 0) and (4, 5) using DDA.

Solution:
Given,
Starting Point: $P(x_1, y_1) = (0, 0)$
Ending Point: $Q (x_2, y_2) = (4, 5)$

Now Slope m= $\frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_{2-} x_1}$ = (5-0) / (5-0) = (5/4) = 1.25 i.e. Slope |m|>1,

From DDA algorithm we have

$$Y_{k+1} = y_k + 1$$

$$x_{k+1} = x_k + (1/m)$$

Hence,

X	Y	X-Plot	Y-Plot	(X,Y)
0	0	0	0	(0,0)
0.8	1	1	1	(1,1)
1.6	2	2	2	(2,2)
2.4	3	2	3	(2,3)
3.2	4	3	4	(3,4)
4	5	4	5	(4,5)

Problem: 02: Consider a line from M (2, 1) to P (8, 3). Using DDA algorithm, rasterize this line.

Solution:

Given,

Starting Point: M (x_1, y_1) = (2, 1)
Ending Point: P (x_2, y_2) = (8, 3)

Now,

Slope m= = $\frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_{2} - x_1}$ = (3-1) / (8-2) = (1/3) = 0.333 i.e. slope |m|<1

From DDA algorithm we have

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

Hence,

X	Y	X-Plot	Y-Plot	(X,Y)
2	1	2	1	(2,1)
3 (2+1)	1.33 (1+0.33)	3	1	(3,1)
4 (3+1)	1.66 (1.33+0.33)	4	2	(4,2)
5	1.993	5	2	(5,2)
6	2,326	6	2	(6,2)
7	2.659	7	3	(7,3)
8	2.999	8	3	(8,3)

Practice:

Q1. Consider a line from A (0, 0) to B (5, 5). Using DDA algorithm, rasterize this line.

Q2. Digitized the line with end points M (3, 7) and N (8, 3) using DDA**.**

paudelmuku@gmail.com

Bresenham's Line Drawing Algorithm

We have to different condition for Bresenhams line drawing algorithm for positive slope.

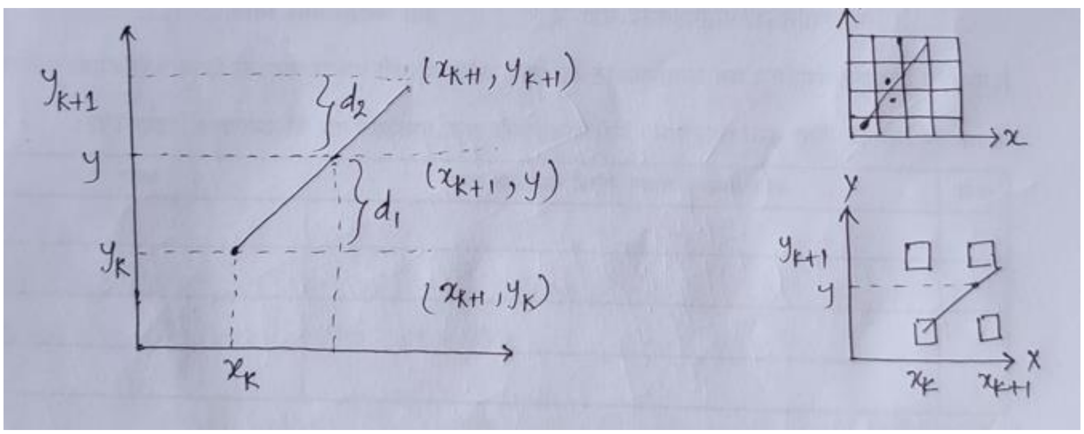
1. **<u>For +Ve Slope less than 1 (i.e. m<1)</u>**

Objective:

To decide which of two possible pixel position is closer to the line path ar each sample step.

If we consider the +ve slope < 1 then pixel position along line path are determined by sampling at X- interval

Let us start from left end point (x_0, Y_0) of the given line we step to each successive column (X position) and plot the pixel whose scan line y-value is closest to the line path.



Let, (x_k, y_k) be the pixel position of starting point of line the next pixel to be plotted is 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 mathematical line path as d_1 and d_2.

The y coordinate on the mathematical line at pixel column position x_{k+1} is calculated as:

y= mx+b => m (x_k+1)+b --------------------------------- (1)

From above figure,

$$d_1 = y- y_k \text{ and } d_2 = (y_{k+1}) - y$$

difference between these two separations is $d_1 - d_2$ then

$$d_1 - d_2 = (y- y_k) – [(y_k +1)-y] => y- y_k -y_k -1 +y => 2y -2y_k -1$$

from equation (1)

$d_1 - d_2 = 2[m(x_k +1) + b] – 2y_k -1$

$= 2m (x_{k +1})+2b – 2y_{k -1}$ ……………………….(2)

If $d_1 – d_2 < 0$ i.e. Distance d_1 is small (d1 <d2) so pixel x_{k+1} , y_k is plotted

If $d_1 – d_2 > 0$ i.e. Distance d_2 is small (d1 >d2) so pixel x_{k+1} , y_{k+1} is plotted

In Equation (2), Slope (m) is present and 'm' gives float value so we need to remove it.

Put m = dy / dx and multiply both side by dx, we get

$$dx(d_1- d_2)= 2 . dx . dy/dx (x_k +1) + 2b\ dx -2 y_k\ dx -dx$$

$$\Rightarrow 2 dy. x_k+2dy +2b . dx – 2y_k\ dx -dx$$

from this expression we can decide the nearest distance the pixel. Hence consider the decision parameter as P^k for the k^{th} step / position.

Therefore, $P_k = dx(d1 -d2) = 2 dy. x_k+2 dy+2b. dx-2y_k dx-dx$

Hence, $P_k = 2dy. X_k - 2 dx . y_k + c$

where, c = 2dy+2b.dx-dx is constant in above expression which doesn't affect the decision.

For the k^{th} position P_k is decision parameter and it changes towards every position.

Now,

To decide the next nearest pixel to be plotted, we have to find next P_k i.e. P_{k+1}

$P_{k+1} = 2dy. x_{k+1} – 2dx. y_{k+1}$ or, $P_{next}=2dy.x_{next} – 2dx. y_{next}$

To find how decision Parameter changes,

$$P_{k + 1} - P_k = 2dy. x_{k+1} – 2dx. y_{k+1} – [2dy. x_k - 2 dx . y_k]$$

$$= 2dy. x_{k+1} – 2dx. y_{k+1} –2dy. x_k + 2 dx . y_k$$

Case-I: If P_{k+1}- P_k <0 , i.e. P_k < 0 => point to be plotted (x_{k+1}, y_k)

case-II: If P_{k+1}- P_k >=0 , i.e. P_k >=0 => point to be plotted (x_{k+1}, y_{k+1})

From Case I:

$P_{k+1}-P_k = 2dy.\ x_{k+1} - 2dx.\ y_{k+1} - 2dy.\ x_k + 2\ dx\ .\ y_k$

$= (2\ dy\ (x_k + 1)) - 2dx.y_k - 2dy\ .\ x_k + 2dx.\ y_k$

$= 2dy\ x_k + 2dy - 2dx.y_k - 2dy\ .\ x_k + 2dx.\ y_k$

$\mathbf{P_{k+1} = P_k + 2dy}$

Similarly, from Case II:

$P_{k+1}-P_k = [2dy.\ x_{k+1} - 2dx.\ y_{k+1}] - [2dy.\ x_k - 2\ dx\ .\ y_k]$

On Solving,

$P_{k+1}-P_k = 2dy - 2dx$

$\mathbf{P_{k+1} = P_k + 2dy - 2dx}$

Now, **<u>Calculate the decision parameter for initial condition</u>**

We know, for k^{th} position

$P_k = 2\ dy.\ x_k - 2\ dx\ y_k + 2\ dy + 2b.\ dx - dx$

Let, P_0 is the initial decision parameter,

$P_0 = 2\ dy\ x_0 - 2\ dx\ y_0 + 2\ dy - 2\ dx\ b - dx$

$\Rightarrow 2\ dy\ x_0 - 2\ dx\ y_0 + 2\ dy - 2\ dx\ [y_0 - (dy/dx).x_0] - dx$

As we know, $y=mx+b$

$y_o = mx_0 + b$

$b = Y_o - m.x_o$

$b = Y_o - (dy/dx)*x_0$

- ⇨ $2\ dy\ x_0 - 2\ dx\ y_0 + 2dy - 2\ dx\ y_0 - 2dy\ x_0 - dx$
- ⇨ $\mathbf{P_0 = 2dy - dx}$ Which is required initial decision parameter

Algorithm

Step 1 : Input the two line end points &store left end point in (x_0 ,y_0)

Step 2 : load (x_0 ,y_0) into the from buffer plot the first point.

Step 3 : Calculate constants Δx , Δy, 2Δy and 2Δy - 2Δx and obtain the starting value for the decision parameter as $P_0 = 2\Delta y - \Delta x$

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

If pk <0 the next point to plot is ($x_k + 1$, y_k) & $p_{k+1} = p_k + 2\Delta y$

Otherwise the next point to plot is ($x_k + 1$, $y_k + 1$) & $p_{k+1} = p_k + 2\Delta y - 2\Delta x$

Step 5 : End

Bresenham's Line Drawing Algorithm for slope (m) >1

Algorithm

Step 1 : Input the two line end points &store left end point in (x_0 ,y_0)

Step 2 : load (x_0 ,y_0) into the from buffer plot the first point.

Step 3 : Calculate constants Δx , Δy, 2Δy and 2Δx - Δy and obtain the starting value for the decision parameter as $P_0 = 2\Delta x - \Delta y$

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

If pk <0 the next point to plot is (x_k , $y_k + 1$) & $p_{k+1} = p_k + 2\Delta x$

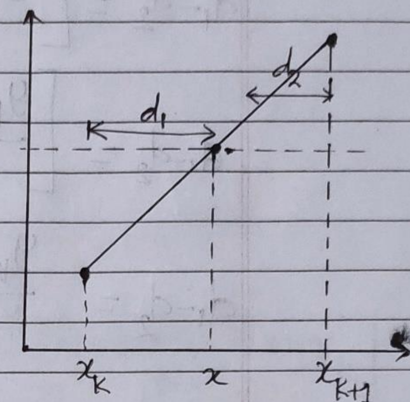
Otherwise the next point to plot is ($x_k + 1$, $y_k + 1$) & $p_{k+1} = p_k + 2\Delta x - 2\Delta y$

Step 5 : End

Derivation:

Bresenham's Line Drawing Algorithm

for $|m| > 1$



⇒ Let (x_k, y_k) be the pixel position determined then the next to ~~the~~ be plotted is either (x_{k+1}, y_{k+1}) or (x_k, y_{k+1})

Let, d_1 and d_2 be the seperation of pixel positions (x_k, y_{k+1}) and (x_{k+1}, y_{k+1}) for the actual line path.

We know $y = mx + b$ then

$$x = \frac{y-b}{m} \quad \text{— (i)}$$

Now, sampling position at y_{k+1}

$$x = \frac{(y_k + 1 - b)}{m} \quad \text{— (ii)}$$

from above figure,

$$d_1 = x - x_k$$
$$d_2 = x_k + 1 - x$$

Now,

To find closes pixel,

$$d_1 - d_2 = (x - x_k) - (x_k + 1) - x$$

Date
Page

put x from (11) we get,

$$d_1 - d_2 = \left[\frac{(y_k + 1 - b)}{m} - x_k\right] - \left[x_k + 1 - \frac{y_k + 1 - b}{m}\right]$$

$$d_1 - d_2 = \left[\frac{y_k + 1 - b - m x_k}{m}\right] - \left[\frac{m x_k + m - y_k - 1 + b}{m}\right]$$

$$d_1 - d_2 = \frac{y_k + 1 - b - m x_k - m x_k - m + y_k + 1 - b}{m}$$

$$m(d_1 - d_2) = 2y_k - 2m x_k - 2b + 2 - m$$

To remove m put $m = \frac{\Delta y}{\Delta x}$ and multiply both side by Δx then,

$$\Delta x \cdot \frac{\Delta y}{\Delta x}(d_1 - d_2) = \Delta x \, 2y_k - 2\cancel{\Delta x} \cdot \frac{\Delta y}{\cancel{\Delta x}} x_k - 2b\Delta x + 2\Delta x - \cancel{\Delta x} \cdot \frac{\Delta y}{\cancel{\Delta x}}$$

$$\Delta y(d_1 - d_2) = 2y_k \Delta x - 2x_k \Delta y - 2b\Delta x + 2\Delta x - \Delta y$$

Let, P_k is the decision parameter, then above expression contains the decision variable x_k and y_k then,

$$P_k = \Delta y(d_1 - d_2) = 2y_k \Delta x - 2x_k \Delta y + C$$

where, $C = 2\Delta x - 2b\Delta x - \Delta y$

then the next decision parameter will be P_{k+1}.

$$P_{k+1} = 2\Delta x y_{k+1} - 2\Delta y x_{k+1}$$

$$P_{next} = 2\Delta x y_{next} - 2\Delta y x_{next}$$

Therefore,

$$P_{k+1} - P_k = [2\Delta x y_{k+1} - 2\Delta y x_{k+1}] - [2\Delta x y_k - 2\Delta y x_k]$$

$$= 2\Delta x y_{k+1} - 2\Delta y x_{k+1} - 2\Delta x y_k + 2\Delta y x_k$$

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

$$\boxed{P_{k+1} = P_k + 2\Delta x (y_{k+1} - y_k) - 2\Delta y (x_{k+1} - x_k)}$$

OR

$$P_{next} = P_{current} + 2\Delta x (y_{next} - y_{current}) - 2\Delta y (x_{next} - x_{current})$$

Case-I : if $P_k \geq 0$ then

$$x_{k+1} = x_k + 1$$
$$y_{k+1} = y_k + 1 \quad \text{and}$$

$$P_{k+1} = P_k + 2\Delta x - 2\Delta y$$

Case-II: if $P_k < 0$ then

$$x_{k+1} = x_k \text{ and } y_{k+1} = y_k + 1$$

$$P_{k+1} = P_k + 2\Delta x$$

To find initial value of decision parameter

As we know

$$P_k = 2\Delta x y_k - 2\Delta y x_k + C$$

$$P_0 = 2\Delta x y_0 - 2\Delta y x_0 + 2\Delta x - 2b\Delta x - \Delta y$$

$$= 2\Delta x y_0 - 2\Delta y x_0 + 2\Delta x - 2(y_0 - m x_0)\Delta x - \Delta y$$

$$= 2\Delta x y_0 - 2\Delta y x_0 + 2\Delta x - 2\Delta x y_0 + 2\Delta x \cdot m \cdot x_0 - \Delta y$$

$$= -2\Delta y x_0 + 2\Delta x + 2\Delta x \cdot \frac{\Delta y}{\Delta x} \cdot x_0 - \Delta y$$

$$= \cancel{-2\Delta y x_0} + 2\Delta x + \cancel{2\Delta y x_0} - \Delta y$$

$$\boxed{P_0 = 2\Delta x - \Delta y}$$

Which is Required Expression.

Note: for BLA:

1. if $|m| \leq 1$, Sample at x-direction i.e $x_{k+1} = x_k + 1$
2. if $|m| > 1$, Sample at y-direction i.e. $y_{k+1} = y_k + 1$.

Findings for $|m| > 1$.

1. Initial decision parameter $(P_0) = 2\Delta x - \Delta y$
2. if $P_k \geq 0$, then plot (x_k+1, y_k+1) and
$$P_{k+1} = P_k + 2\Delta x - 2\Delta y$$
3. if $P_k < 0$, then plot (x_k, y_k+1) and
$$P_{k+1} = P_k + 2\Delta x$$

Mid- Point Circle Generation Algorithm

Circle: A Circle is defined as a S\set of points that are all at a given distance 'r' from the center position (Xc, Yc).

This distance relationship is expressed by the $(x\text{-}x1)^2 + (y\text{-}y1)^2 = r^2$

OR

A Circle is a Set of all points lie at an equal distance (called radios) from a fixed point called Centre.

This is also a scan converting algorithm.

equation of the Circle with center (0,0) is $\mathbf{x^2 + y^2 = r^2}$

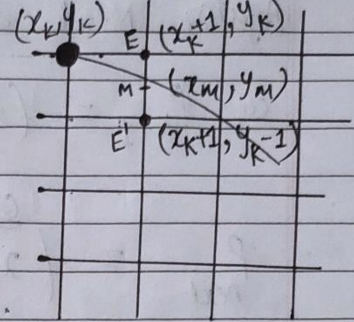
The equation of Circle having center (h, k) is $\mathbf{(x-h)^2 + (y-k)^2 = r^2}$

<u>In the circle drawing</u>

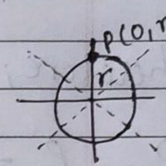
We use symmetry of Circle

For Symmetry, if we draw only one octant, we can easily draw points on other seven octants using reflection procedure.

Here we have to find points for the first octants, when $x \geq y$ then it indicates the end of octant 1^{st}.



The first Co-ordinate point is $(0, r)$. to draw the circle, The X-value increasing as unit interval and y-value decreasing along the value of X.



Now, The next pixel to draw is either (x_k+1, y_k) or (x_k+1, y_k-1). so the mid point of these two pixel is (x_m, y_m).

then,
$$x_m = \frac{x_k+1+x_k+1}{2} = x_k+1$$

and
$$y_m = \frac{y_k+y_k-1}{2} = y_k - \frac{1}{2}$$

$\because$ mid-point formulae
$$x_m = \frac{x_1+x_2}{2}$$
$$y_m = \frac{y_1+y_2}{2}$$

We know
The eq^n of circle is $x^2+y^2=r^2$ —— (I)

putting (x_m, y_m) in eq^n (I) we get

$$\Rightarrow (x_k+1)^2 + \left(y_k - \frac{1}{2}\right)^2 = r^2$$

Let P_K is the decision parameter then

$$\boxed{P_K = (x_k+1)^2 + \left(y_k - \frac{1}{2}\right)^2 - r^2} \quad \text{—— (II)}$$

Again, Next decision parameter $P_k = P_{k+1}$

So, $P_{k+1} = (x_{k+1} + 1)^2 + (y_{k+1} - 1/2)^2 - r^2$

Or, $P_{next} = (x_{next} + 1)^2 + (y_{next} - 1/2)^2 - r^2$

Therefore,

$$p_{k+1} - p_k = (x_k + 2)^2 + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2 - (x_k + 1)^2 - \left(y_k - \frac{1}{2}\right)^2 + r^2$$

$$p_{k+1} = p_k + x_k^2 + 4x_k + 4 + y_{k+1}^2 - y_{k+1} + \frac{1}{4} - x_k^2 - 2x_k - 1 - y_k^2 + y_k - \frac{1}{4}$$

$$p_{k+1} = p_k + 2x_k + 3 + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k)$$

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

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

If $p_k < 0$;

$$y_{k+1} = y_k \Rightarrow p_{k+1} = p_k + 2x_{k+1} + 1$$

If $p_k \geq 0$;

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

For initial decision parameter;

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

$$p_0 = f\left(1, r - \frac{1}{2}\right) = 1 + \left(r - \frac{1}{2}\right)^2 - r^2 = \frac{5}{4} - r$$

All increments are integer, rounding $\frac{5}{4}$ will give 1 so,

$$p_0 = 1 - r$$

Algorithm:

1. The input radios ‘r’ and center (Xc, Yc) to obtain the first octant point in the circumference of a circle is centered on the origin as (X0, Y0) = (0,r)
2. calculate the initial decision parameter P0= 1- r
3. at each x k position starting k=0 perform following operation
   3.1. if P k <0 then plotting point will be (x_{k+1}, y_k) and $P_{k+1} = P_k + 2 * x_{k+1} + 1$
   3.2. else plot (x_{k+1}, y_{k-1}) and $P_{k+1} = P_k + 2 * x_{k+1} - 2 * y_{k+1} + 1$
4. Determine the symmetry points in the other octants
5. move at each point by given center i.e. (X+ Xc and Y+Yc) if necessary
6. Repeat step 3 to 5 until the X>=Y.

Page

Ellipse Generation Algorithms

→ Ellipse: is an elongated circle, therefore elliptical curves can be generated by modifying circle drawing algorithm's procedures.

→ Circle → has one radious in all direction
ellipse → has two radii

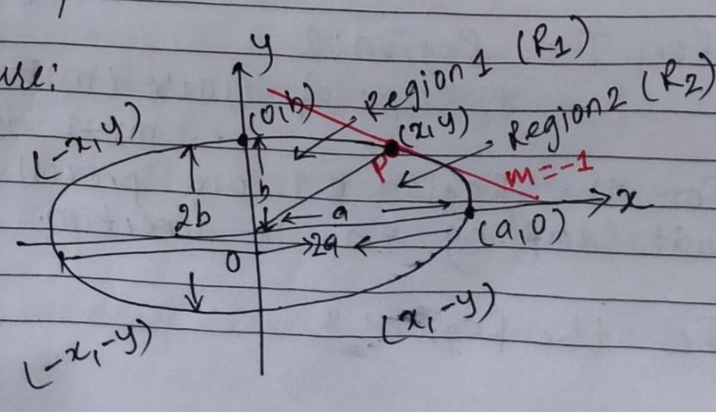
- one in x-axis (called major axis) (2a)
- next in y-axis (called minor axis) (2b)

→ Due to these two different radii circle and ellipse has different symmetry.

→ In circle 8 point symmetry
In ellipse 4 point symmetry

→ In Circle, we need to plot only one octant of any quardant but in ellipse we need to plot two octants (i.e. one complete quardant to plot entire ellipse).

→ See figure:



→ Ellipse generating algorithm is applied through 1^{st} quadrant according to the slope of ellipse.

→ slope is obtained by differentiating the ellipse function.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

$$x^2b^2 + y^2a^2 = a^2b^2$$

$$x^2b^2 + y^2a^2 - a^2b^2 = 0 \quad \text{— (1)}$$

differentiating (1) w.r.to x

$$2xb^2 + 2y\,a^2\frac{dy}{dx} = 0$$

$$\frac{dy}{dx} = -\frac{2xb^2}{2ya^2}$$

→ At the boundary point between Region 1 and Region 2 of ellipse slope is -1.

i.e. at point P, $\frac{dy}{dx} = -1$. } see figure for point P.

→ For The Region: 1

* slope of curve (m) is greater than -1.

i.e. $m > -1$ for R_1.

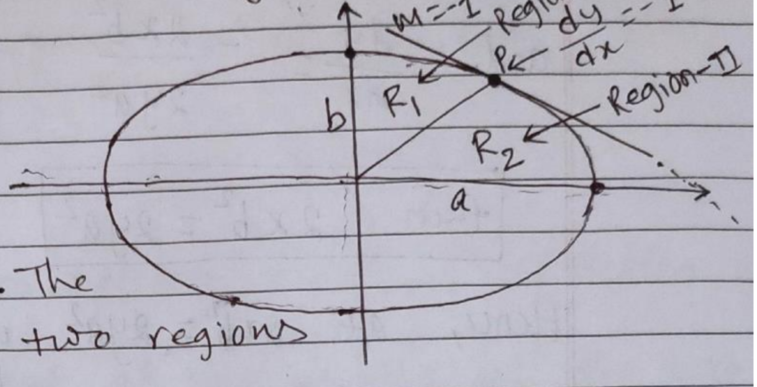
So, For the Region: 1 we process by taking unit sampling in x-direction.

→ For the Region: 2

Page

* slope is less than 1. i.e. $m < 1$ for R_2.
* sample in y-direction by unit interval

Derivation:



Let us consider one quarter of an ellipse. The curve is divided into two regions R_1 and R_2.

In the region-I, the slope on the curve is greater than -1 while in region-II slope is less than -1.

Also, consider the general equation of an ellipse,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

$$\Rightarrow \quad x^2b^2 + y^2a^2 - a^2b^2 = 0$$

Where, a is horizontal radious and b is vertical radious

Now,

Lets start from the point (0,b) in region-I and take unit step sampling towards x-dir^n untill the boundary between R_1 and R_2.

and, In R_2 we sample at y-direction to test the slope at each point of curve.

At the boundary region between R_1 and R_2

Prepared By: Er. Mukunda Paudel
paudelmuku@gmail.com

$\dfrac{dy}{dx} = -1$

and, $\dfrac{dy}{dx} = -\dfrac{2xb^2}{2ya^2}$

$\because \dfrac{x^2}{a^2} + \dfrac{y^2}{b^2} = 1$

diff.

$2xb^2 - 2ya^2\dfrac{dy}{dx} = 0$

at boundary $\dfrac{dy}{dx} = -1$

so, $2xb^2 = 2ya^2$

$\boxed{\text{then } 2xb^2 = 2ya^2}$

Hence, at $2xb^2 = 2ya^2$ we are in Region-I & Region-II boundary point.

if $2xb^2 > 2ya^2$ we move out of Region-I

Therefore

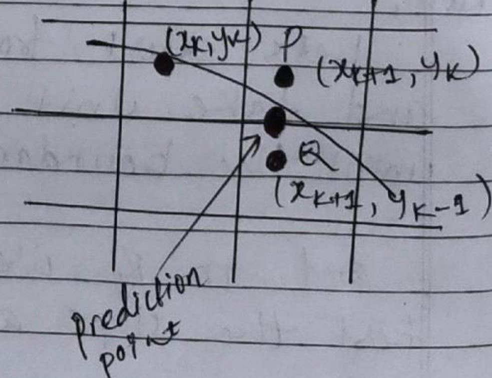
for R_1 $m < 1$ and $2xb^2 < 2ya^2$

for R_2 $m > 1$ and $2xb^2 > 2ya^2$

In Region-I

We know $\dfrac{dy}{dx} > -1$ and x is always incremented in each step.

also, we know that at R_1, $2xb^2 < 2ya^2$



When x is always incremented in each step i.e. $x_{k+1} = x_k + 1$

and $y_{k+1} = y_k$ if P is selected

$= y_k - 1$ if Q is selected

i.e (y_{k+1} is either y_k or $y_k - 1$)

To make the decision between P & Q, a prediction $(x_k+1, y_k-\frac{1}{2})$ is set at the middle bt^n the two candidate pixels. This is calculated by $\begin{cases} x_m = \frac{x_1+x_2}{2} \\ y_m = \frac{y_1+y_m}{2} \end{cases}$

put $(x_k+1, y_k-\frac{1}{2})$ in the eq^n of ellipse then

$$E_{function} = b^2(x_k+1)^2 + a^2(y_k-\tfrac{1}{2})^2 - a^2b^2$$

Let, P_k be the decision parameter as recent expression contains values satisfying standard equation of ellipse.

$$P_k = b^2(x_k+1)^2 + a^2(y_k-\tfrac{1}{2})^2 - a^2b^2$$

for Region-1 denote P_k as P_{1K}

$$\boxed{P_{1K} = b^2(x_k+1)^2 + a^2(y_k-\tfrac{1}{2})^2 - a^2b^2}$$

$$P_{1(k+1)} = b^2(x_{k+1}+1)^2 + a^2(y_{k+1}-\tfrac{1}{2})^2 - a^2b^2$$

i.e. $\boxed{P_{1next} = b^2(x_{next}+1)^2 + a^2(y_{next}-\tfrac{1}{2})^2 - a^2b^2}$

Now we are sampling at x direction so

$$x_{k+1} = x_k + 1 \quad \text{the}$$

$$P_{1(k+1)} = b^2\{(x_k+1)+1\}^2 + a^2\left(y_{k+1}-\frac{1}{2}\right)^2 - a^2b^2$$

$$= b^2\{(x_k+1)^2 + 2(x_k+1)*1+1\} + a^2\left(y_{k+1}-\frac{1}{2}\right)^2 - a^2b^2$$

Now,

$$P_{1(k+1)} - P_k = \left[\{b^2(x_k+1)^2 + 2(x_k+1)+1\} + a^2\left(y_{k+1}-\frac{1}{2}\right)^2 - a^2b^2\right]$$
$$- \left[b^2(x_k+1)^2 + a^2\left(y_k-\frac{1}{2}\right)^2 - a^2b^2\right]$$

$$\Rightarrow \left[b^2(x_k+1)^2 + 2b^2(x_k+1) + b^2 + a^2(y_k+1)^2 - a^2*y_{k+1} + a^2*\frac{1}{4}\right.$$
$$\left.- a^2b^2\right] - \left[b^2(x_k+1)^2 + a^2y_k^2 - a^2y_k + a^2*\frac{1}{4} - a^2b^2\right]$$

$$\Rightarrow 2b^2(x_k+1) + b^2 + a^2\{(y_{k+1}^2 - y_k) - (y_{k+1} - y_k)\}$$

Case-I If $P_{1k} < 0$ then midpoint is inside the ellipse boundary and the pixel on scan line y_k is closer to boundary of ellipse

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

$y_{k+1} = y_k$

and, ~~P_{k+1} for region-1 (Lets denote by $P_{1(k+1)}$)~~

$$P_{1(k+1)} = P_{1k} + 2b^2(x_{k+1}) + b^2$$

Page

case-II if $P_{1K} \geq 0$ then pridected midpoint is outside or on the ellipse boundary then we select pixel y_{K-1} in same scan line.

Hence, $x_{K+1} = x_K + 1$

$y_{K+1} = y_K - 1$

and

$$P1_{K+1} = P_{1K} + 2b^2 x_{K+1} - 2a^2 y_{K+1} + b^2$$

To Find Initial Decision Parameter for R1.

We have the starting point for the ellipse is (0,b).

i.e. $(x_0, y_0) = (0, b)$

Then, The next point will be either (1,b) or (1,b-1).

from (1,b) and (1, b-1) we can calculate mid point and i.e $(1, b-\frac{1}{2})$

let, $P1_0$ is the initial decision parameter for R1.

then, $P1_0 = b^2 + a^2(b-\frac{1}{2})^2 - a^2b^2$ {∵ putting $(1, b-\frac{1}{2})$ in eqn of ellipse

$$\boxed{P1_0 = b^2 - a^2b + \frac{a^2}{4}}$$

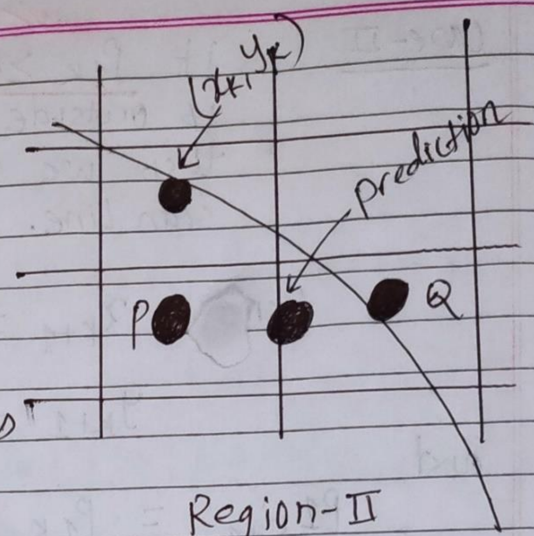
For Region-2

In R2 $\Rightarrow$ $2xb^2 \geq 2ya^2$

In R2, y is always decremented in each step i.e.

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

then x_k has two possibilities



if P is selected

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

if Q is selected $x_{k+1} = x_k + 1$.

Therefore, we have to decide by taking mid point of (x_k, x_k+1) as prediction pixel.

mid point of P & Q is $\left(x_k + \frac{1}{2}, y_k - 1\right)$

putting this point in standered equation of Ellipse, we get,

$$E_{function} = \left(x_k + \tfrac{1}{2}, y_k - 1\right)$$

$$= b^2\left(x_k + \tfrac{1}{2}\right)^2 + a^2(y_k - 1)^2 - a^2b^2$$

Let, $P2_K$ is the decision parameter for Region-II then

$$\boxed{P2_K = b^2\left(x_k + \tfrac{1}{2}\right)^2 + a^2(y_k - 1)^2 - a^2b^2}$$

In Region-2 we are sampling at y-direction then $y_{k+1} = y_k - 1$

$$P2_{k+1} = b^2 \left(x_{k+1} + \frac{1}{2}\right)^2 + a^2 \left(y_{k+1} - 1\right)^2 - a^2 b^2$$

i.e. $$P2_{next} = b^2 \left(x_{next} + \frac{1}{2}\right)^2 + a^2 \left(y_{next} - 1\right) - a^2 b^2$$

Now

$$P2_{k+1} - P2_k = \left\{ b^2 \left(x_{k+1} + \frac{1}{2}\right)^2 + a^2 (y_{k+1} - 1)^2 - a^2 b^2 \right\} - \left\{ b^2 \left(x_k + \frac{1}{2}\right)^2 + a^2 (y_{k-1})^2 - a^2 b^2 \right\}$$

Here only y_{k+1} is $y_k - 1$.

$$\Rightarrow \left\{ b^2 \left(x_{k+1} + \frac{1}{2}\right)^2 + a^2 (y_{k+1} - 1)^2 - a^2 b^2 - b^2 \left(x_k + \frac{1}{2}\right)^2 - a^2 (y_{k-1})^2 + a^2 b^2 \right\}$$

$$\Rightarrow \left\{ b^2 \left(x_{k+1} + \frac{1}{2}\right)^2 + a^2 \left(y_{k+1} - 1\right)^2 - b^2 \left(x_k + \frac{1}{2}\right)^2 - a^2 \left(y_{k-1}\right)^2 \right\}$$

$$\Rightarrow b^2 \left(x_{k+1} + \frac{1}{2}\right)^2 + a^2 \left\{ (y_k - 1) - 1 \right\}^2 - b^2 \left(x_k + \frac{1}{2}\right)^2 - a^2 (y_{k-1})^2$$

$$\Rightarrow b^2 x_{k+1}^2 + b^2 x_{k+1} + \frac{b^2}{4} + a^2 (y_k - 1)^2 - 2a^2 (y_k - 1) + a^2 - b^2 x_k^2 - b^2 x_k - \frac{b^2}{4} - a^2 (y_{k-1})^2$$

$$\Rightarrow \boxed{a^2 + b^2 (x_{k+1}^2 - x_k^2) - 2a^2 (y_k - 1) + b^2 (x_{k+1} - x_k)}$$

Case-I if $P2_k < 0$, then the predicted mid point is inside the boundary of ellipse and pixel x_k is closer to ellipse boundary

then,

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

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

and,

$P2_{k+1} = P2_k + b^2[(x_{k+1}^2 - x_k^2) + (x_{k+1} - x_k)] - 2a^2 (y_k - 1) + a^2$ {because $x_{k+1} = x_k + 1$}

on solving

$P2_{k+1} = P2_k + 2b^2 x_{k+1} - 2a^2 y_{k+1} - a^2$

Case-II: if $P2_k \geq 0$ then the predicted mid point is outside or on boundary of ellipse then we choose $x_k + 1$ as closer pixel.

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

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

and, $P2_{k+1} = P2_k + b^2[(x_{k+1}^2 - x_k^2) + (x_{k+1} - x_k)] - 2a^2(y_k - 1) + a^2$

$\Rightarrow P2_k - 2a^2(y_k - 1) + a^2$

$\boxed{P2_{k+1} \Rightarrow P2_k - 2a^2 y_{k+1} + a^2}$

To find Initial Decision parameter for R2.

When we enter in Region-II of ellipse, the initial position (x_0, y_0) is taken the last position selected in Region-1 and the initial decision parameter in R2 is:

$$P2_0 = f\left(x_0 + \frac{1}{2}, y_0 - 1\right)$$

put this point in the standard eqn of ellipse

$$\boxed{P2_0 = b^2\left(x_0 + \frac{1}{2}\right)^2 + a^2(y_0 - 1)^2 - a^2b^2}$$

Findings

for R1

$P1_0 = b^2 + 0.25a^2 - a^2b$

if $P1_k < 0$

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

$y_{k+1} = y_k$

$P1_{k+1} = P1_k + 2b^2(x_{k+1}) + b^2$

else,

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

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

$P1_{k+1} = P1_k + 2b^2x_{k+1} - 2a^2y_{k+1} + b^2$

for R2

$P2_0 = b^2(x0+0.5)^2 + a^2(y0-1)^2 - a^2b^2$

if $P2_k < 0$

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

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

$P2_{k+1} = P2_k + 2b^2x_{k+1} - 2a^2y_{k+1} + a^2$

else,

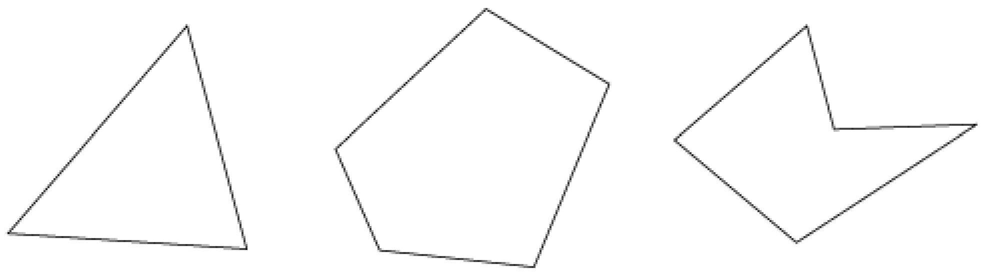
$x_{k+1} = x_k$

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

$P2_{k+1} = P2_k - 2a^2y_{k+1} + a^2$

Prepared By: Er. Mukunda Paudel
paudelmuku@gmail.com

Area Filling / Filled Area Primitives



- ❖ To provide more realistic view on the various objects filling of color is one of the best way.
- ❖ For filling a picture or object with color's, we have 2 different algorithmic approaches
  1. Area filling scan line algorithm
  2. Seed Fill Algorithm
     - ➢ Seed fill Algorithm is further categorized in 2 different algorithm

     2.1Boundary Fill Algorithm

     2.2Flood Fill Algorithm

Instead of Using filling algorithms such as Flood fill algorithm, Boundary fill algorithm and scanline polygon fill algorithm, Programmatically we can color the objects by Using inbuilt graphics functions such as **floodfill(), setfillstyle()** we can fill the object with color's directly without using any filling algorithm in C program.

Area filling scan line algorithm

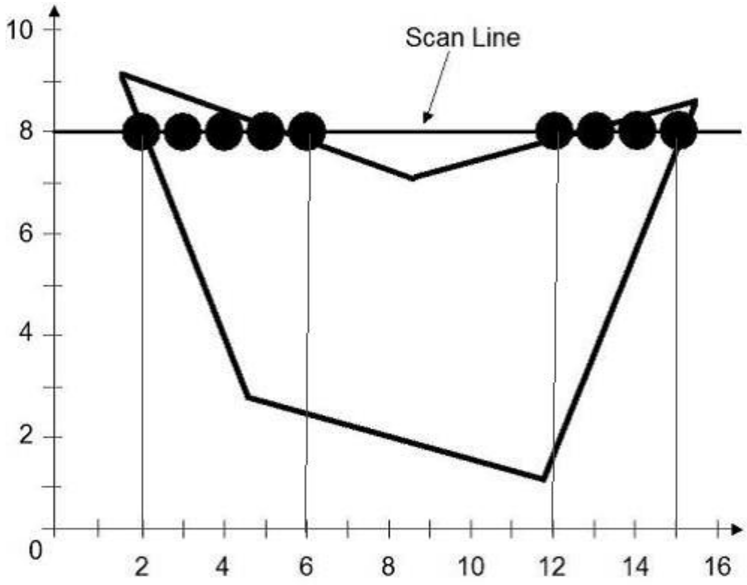
- ❖ Also called scan line polygon fill algorithm
- ❖ The basic scan-line algorithm is as follows:

1. find the intersections of the scan line with all edges of the polygon
2. Sort the intersections by increasing x coordinate (i.e. short in ascending order of X coordinate)
3. Make pair of two x coordinate
4. Fill in all pixels between pairs of intersections that lie interior to the polygon

- ❖ The scan-line polygon-filling algorithm involves
- ✓ the **horizontal scanning** of the polygon from its **lowermost** to its **topmost** vertex,

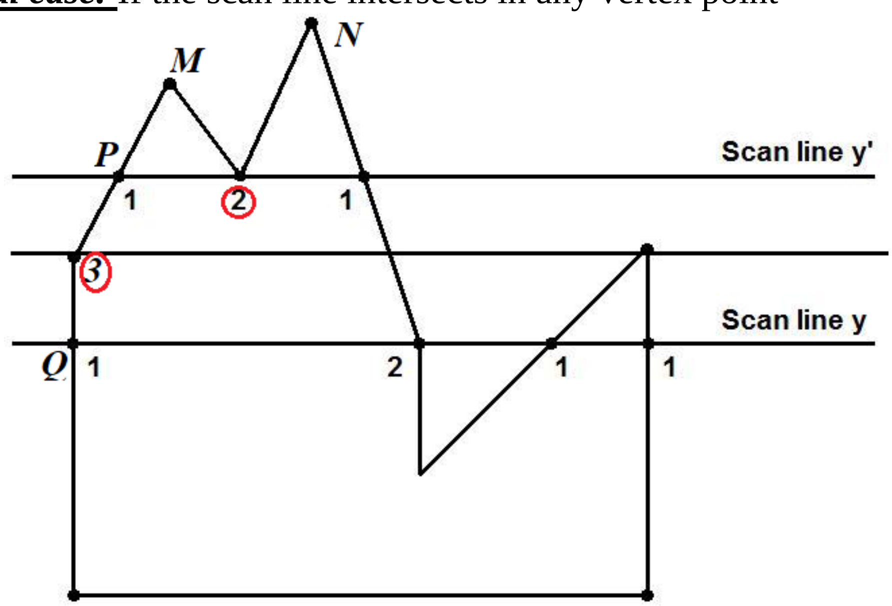
- ✓ identifying which edges intersect the scan-line, and
- ✓ Finally drawing the interior horizontal lines with the specified fill color.

For example



In this polygon ,
Scan line cuts the polygon edges in x= 2, x=6, x=12 and x=15.
Now sort the x cordinate in ascending order we get – [2,6,12,15]
Then make a pair of sorted pixel as (2,6) and (12,15)
Finally fill the two pair (2,6) and (12,15) with a color of your choice.

1. ***Special case:*** If the scan line intersects in any vertex point



As shown in above polygon, scan line Y’ intersects in the vertex 2. Similarly next scan line intersects in vertex 3.

In this case, if all the vertices that are connected in intersected vertex are in same direction of scan line then repeat the vertex value and pair with other intersecting point.

As→ M and N are two vertex that are connected with vertex 2. M and N both are in same direction of scan line y’ (i.e. both M, N are in up direction of scan line or on top). In this case repeat the value of vertex 2 during sorting x direction value in ascending order and make two pair as (1, 2) and (2, 1). Then fill color.

As→P and Q are two vertex that are connected with vertex 3. P and Q both are in opposite direction of second scan line (i.e. P in above the scan line and Q in below of scan line). In this case don’t repeat the value of vertex 3 during sorting x direction value in ascending order. Just take it once and make pair with other intersecting point of polygon boundary. Then fill color

Boundary Fill Algorithm

- Also called Edge Fill Algorithm
- An alternative approach for filling an area is to start at a point inside the area and “paint” the interior, point by point, out to the boundary
- This is a particularly useful technique for filling areas with irregular borders

The algorithm makes the following assumptions

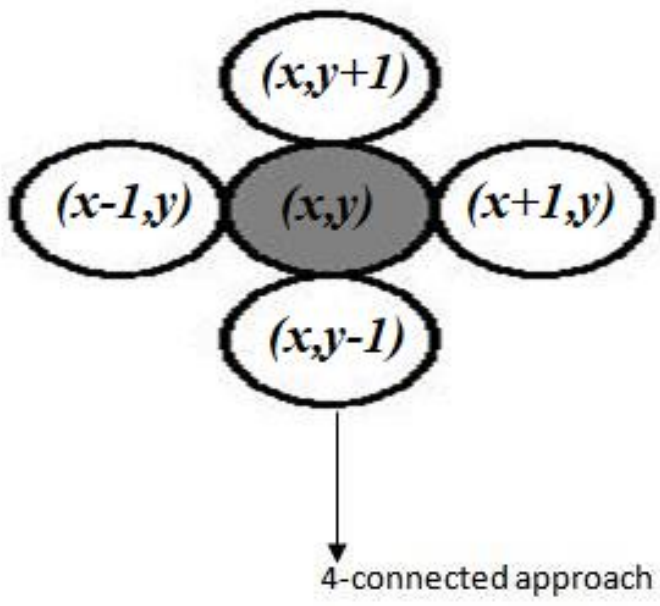
- one interior pixel is known, and
- Pixels in boundary are known.
- Boundary pixels are filled with a single color

- ✓ If the boundary of some region is specified in a single color, we can fill the interior of a region, pixel by pixel, until the boundary color is encountered.
- ✓ Basically, a boundary-fill algorithm starts from an interior point (x, y) and sets the neighboring points to the desired color
- ✓ This procedure continues until all pixels are processed up to the designated boundary for the area.
- ✓ Boundary fill algorithm is recursive in nature.

paudelmuku@gmail.com

There are two methods for processing neighboring pixels from a current point

1. **4-Connected**
   - ✓ Also called 4-neighbouring points method
   - ✓ These are the pixel positions that are right, left, above, and below the current pixel.



   - ✓ In 4 connected approach, we can fill an object in only 4 directions. We have 4 possibilities for proceeding to next pixel from current pixel.

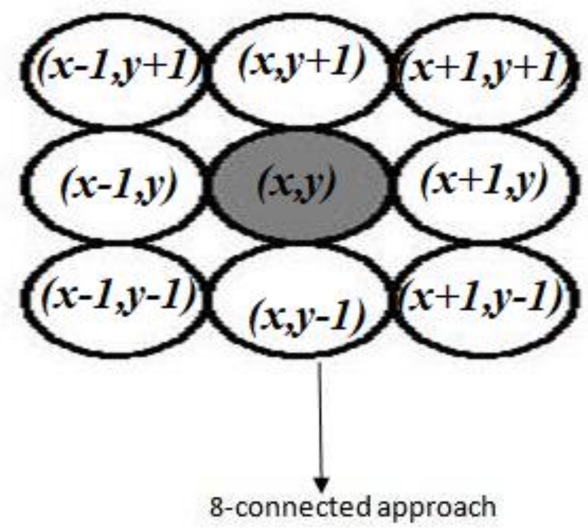
<u>Function for 4 connected approach:</u>

```
Void boundary_fill (int x, int y, int fill_color, int boundary_color)
{
        if ((getpixel(x, y) != boundary_color) && (getpixel(x, y) != fill_color))
                {
                        delay (10);
                        putpixel(x, y, fill_color);
                        boundary_fill(x + 1, y, fill_color, boundary_color);
                        boundary_fill(x - 1, y, fill_color, boundary_color);
                        boundary_fill(x, y + 1, fill_color, boundary_color);
                        boundary_fill(x, y - 1, fill_color, boundary_color);
                }
}
```

Prepared By: Er. Mukunda Paudel
paudelmuku@gmail.com

2. 8-Connected

- ✓ Also called 8-neighbouring points method
- ✓ This method is used to fill more complex figures.
- ✓ These are the pixel positions that are right, left, above, below and four diagonal points of the current pixel.



- ✓ In 8 connected approach, we can fill an object in 8 directions. We have 8 possibilities for proceeding to next pixel from current pixel

<u>Function for 8 connected approach:</u>

```
void boundary_fill(int x, int y, int fill_color, int boundary_color)
{
if ((getpixel(x, y) != boundary_color) && (getpixel(x, y) != fill_color))
        {
         delay(10);
                putpixel(x, y, fill_color);
                boundary_fill(x + 1, y, fill_color, boundary_color);
                boundary_fill(x , y+1, fill_color, boundary_color);
                boundary_fill(x+1, y + 1, fill_color, boundary_color);
                boundary_fill(x-1, y - 1, fill_color, boundary_color);
                boundary_fill(x-1, y, fill_color, boundary_color);
                boundary_fill(x , y-1, fill_color, boundary_color);
                boundary_fill(x-1, y + 1, fill_color, boundary_color);
                boundary_fill(x+1, y - 1, fill_color, boundary_color);
        }
}
```

paudelmuku@gmail.com

Algorithm

1. Start from an interior point.
2. If the current pixel is not already filled and if it is not an edge point, then set the pixel with the fill color, and store its neighboring pixels (**4 or 8-connected**). Store only neighboring pixel that is not already filled and is not an edge point.
3. Select the next pixel from the stack, and continue with step 2.

Flood Fill Algorithm

- Sometimes it is required to fill in an area that is not defined within a single color boundary.
- In such cases we can fill areas by replacing a specified interior color instead of searching for a boundary color.
- This approach is called a flood-fill algorithm. Like boundary fill algorithm, here we start with some seed and examine the neighbouring pixels.
- However, here pixels are checked for a specified interior color instead of boundary color and they are replaced by new color.
- ***Using either a 4-connected or 8-connected approach***, we can step through pixel positions until all interior point have been filled.
- The following procedure illustrates the recursive method for filling 4-connected region using flood-fill algorithm.

Procedure:

```
Void  flood-fill(x, y, new-color, old-color)
 {
 if(getpixel (x,y) = = old-color)
 {
delay(10);
putpixel (x, y, new-color)
flood-fill (x + 1, y, new-color, old -color);
flood-fill (x, y + 1, new -color, old -color);
flood-fill (x - 1, y, new -color, old -color);
```

```
flood-fill (x, y - l, new -color, old-color);
 }
}
```

In above program *'getpixel()'* function gives the color of .specified pixel and *'putpixel()'* function draws the pixel with specified color.

Algorithm insight

1. We start from a specified interior pixel (x, y) and reassign all pixel values that are currently set to a given interior color with the desired fill color.
2. If the area has more than one interior color, we can first reassign pixel values so that all interior pixels have the same color.
3. Using either 4-connected or 8-connected approach, we then step through pixel positions until all interior pixels have been repainted.

Difference between Boundary fill and Flood Fill

Boundary Fill Algorithm	**Flood Fill Algorithm**
1. Area filling is started inside a point with in a boundary region and fill the region with in the specified color until it reaches the the boundary.	Area filling is started from a point and it replaces the old color with the new color
2. It is used in interactive packages where we can specify the region boundary	It is used when we cannot specify the region boundary
3. It is less time consuming	It consumes more time
4. It searches for boundary.	It searches for old color
5. Write algorithmic program yourself from above..	Write algorithmic program yourself from above..

End of Unit 1