**Debre Berhan university**

**College of computing**

Department of computer science

Computer graphics group assignment

Course title: Computer graphics

Course code: cosc 3121

**Group members**

1. Senait Yohhanis…………………………………………………………………… DBUR/1303/10
2. Wendwesen Abebe……………………………………………………………. DBUR/1341/10
3. Woldeyes Afework……………………………………………………………… DBUR/1343/10
4. Natnael Antenyismu………………………………………………………….. DBUR/1284/10
5. Nebiyu Daniel……………………………………………………………………… DBUR/1287/10

Submitted to: Ir Kinde

Date:1/6/2021

**Acknowledgement**

First of all we would like to thank our God for bringing us together from nine month of home stay to school safe. We also want you to thank for your support and the sacrifice that you are facing for us to give what we should get in spite of the situation what we are facing so far so we again want to give our best thank to you.

Table of Contents

[**1. APPILICATION OF COMPUTER GRAPHICS** 4](#_Toc60842006)

[Advantages: 4](#_Toc60842007)

[Example of Computer Graphics Packages: 5](#_Toc60842008)

[**2. What Is an Aspect Ratio** 5](#_Toc60842009)

[Advantage 5](#_Toc60842010)

[**3.** **Random Scan and Raster Scan Display** 9](#_Toc60842011)

[**Random Scan Display** 9](#_Toc60842012)

[**Advantages:** 10](#_Toc60842013)

[**Disadvantages:** 10](#_Toc60842014)

[**Raster Scan Display** 10](#_Toc60842015)

[Advantages: 11](#_Toc60842016)

[Disadvantages: 11](#_Toc60842017)

[**Differentiations between Random and Raster Scan Display** 11](#_Toc60842018)

[4. **Cathode Ray Tube (CRT)** 12](#_Toc60842019)

[**Components of CRT:** 13](#_Toc60842020)

[**Working of CRT** 13](#_Toc60842021)

[5. frame buffer 14](#_Toc60842022)

[6. **Solution:** 14](#_Toc60842023)

[7. Basic 2D Geometric Transformation 15](#_Toc60842031)

[**Homogenous Coordinates** 15](#_Toc60842032)

[**Translation** 15](#_Toc60842033)

[**Rotation** 16](#_Toc60842034)

[**Scaling** 17](#_Toc60842035)

[**Reflection** 18](#_Toc60842036)

[**Shear** 18](#_Toc60842037)

[X-Shear 18](#_Toc60842038)

[Y-Shear 19](#_Toc60842039)

[**Composite Transformation** 19](#_Toc60842040)

[**8.** **Basic concept in line drawing** 20](#_Toc60842041)

[Properties of a Line Drawing Algorithm 20](#_Toc60842042)

[Equation of the straight line 21](#_Toc60842043)

[Algorithms of Line Drawing 21](#_Toc60842044)

[**9.** **Circle generation** 22](#_Toc60842045)

[**Defining a Circle:** 22](#_Toc60842046)

[**Bresenham’s circle drawing algorithm** 23](#_Toc60842047)

[10. **Animation** 27](#_Toc60842048)

[**Application Areas of Animation** 29](#_Toc60842049)

# **1. APPILICATION OF COMPUTER GRAPHICS**

**1.1 Education and Training:** Computer-generated model of the physical, financial and economic system is often used as educational aids. Model of physical systems, physiological system, population trends or equipment can help trainees to understand the operation of the system.

* For some training applications, particular systems are designed. For example Flight Simulator.

**Flight Simulator:** It helps in giving training to the pilots of airplanes. These pilots spend much of their training not in a real aircraft but on the ground at the controls of a Flight Simulator.

Advantages:

1. Fuel Saving
2. Safety
3. Ability to familiarize the training with a large number of the world's airports.

**1.2 Use in Biology:** Molecular biologist can display a picture of molecules and gain insight into their structure with the help of computer graphics.

**1.3 Computer-Generated Maps:** Town planners and transportation engineers can use computer-generated maps which display data useful to them in their planning work.

**1.4 Architect:** Architect can explore an alternative solution to design problems at an interactive graphics terminal. In this way, they can test many more solutions that would not be possible without the computer.

**1.5 Presentation Graphics:** Example of presentation Graphics are bar charts, line graphs, pie charts and other displays showing relationships between multiple parameters. Presentation Graphics is commonly used to summarize

* Financial Reports
* Statistical Reports
* Mathematical Reports
* Scientific Reports
* Economic Data for research reports
* Managerial Reports
* Consumer Information Bulletins
* And other types of reports

**1.6 Computer Art:** Computer Graphics are also used in the field of commercial arts. It is used to generate television and advertising commercial.

**1.7 Entertainment:** Computer Graphics are now commonly used in making motion pictures, music videos and television shows.

**1.8 Visualization:** It is used for visualization of scientists, engineers, medical personnel, business analysts for the study of a large amount of information.

**1.9 Educational Software:** Computer Graphics is used in the development of educational software for making computer-aided instruction.

**1.10 Printing Technology:** Computer Graphics is used for printing technology and textile design.

Example of Computer Graphics Packages:

1. LOGO
2. COREL DRAW
3. AUTO CAD
4. 3D STUDIO
5. CORE
6. GKS (Graphics Kernel System)
7. PHIGS
8. CAM (Computer Graphics Metafile)
9. CGI (Computer Graphics Interface)

# **2. What Is an Aspect Ratio**

The aspect ratio of an image is the proportional difference between width and height. The most popular aspect ratio is 3:2. It is commonly expressed as two numbers separated by a colon, as in *16:9*.For an *x*:*y* aspect ratio, the image is *x* units wide and *y* units high. Widely used aspect ratios include 1.85:1 and 2.39:1 in [film](https://en.wikipedia.org/wiki/Film) photography, 4:3 and [16:9](https://en.wikipedia.org/wiki/16:9) in [television](https://en.wikipedia.org/wiki/Television), and 3:2 in [still camera](https://en.wikipedia.org/wiki/Still_camera) photograph  
This is the aspect ratio of 35mm [film cameras](https://expertphotography.com/the-complete-guide-to-film-photography-94-tips/) and has been around for a long time. It became the standard for modern full frame cameras as well.

Different images on different screens have a different ratio.

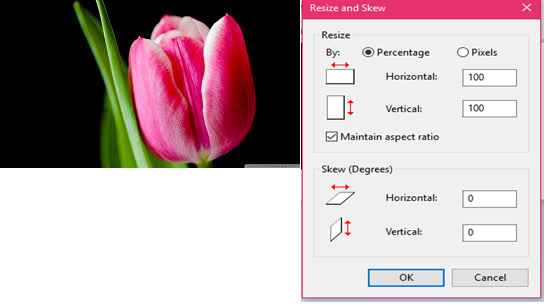
**Common aspect ratio is as follows:**

1.33:1, 1.37:1, 1.43:1, 1.50:1, 1.56:1, 1.66:1, 1.75:1, 1.78:1, 1.85:1, 2.00:1, e.t.c

### Advantage

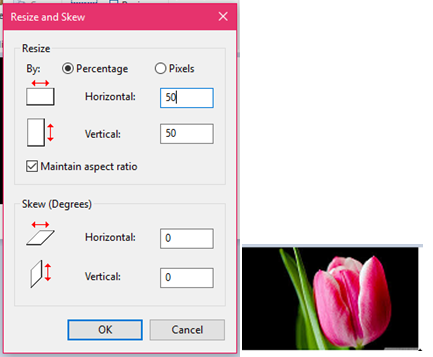
1. It maintains balance in the appearance of an image (ratio between horizontal and vertical pixel).
2. When the aspect ratio is increased, images do not distort.

**For Example:**



Above image has 100 rows and 100 columns. If we resize the image, the image quality will remain the same.

Now changing the value of row and column, we get the image in balanced resolution.



**For example:**

Here is a gray scale image given with aspect ratio 5:2, and the resolution of the image is 480000 pixels.

**Calculate the following things:**

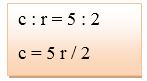
1. Dimensions of the image.
2. Size of the image.

**Given**

Aspect ratio: c:r = 5:2  
Pixel resolution: c \* r = 480000  
Bits per pixel: grayscale image = 8bpp

**Solution 1:**

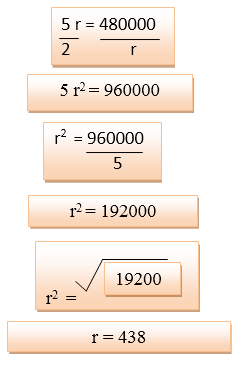
Equation 1 -----------------------------



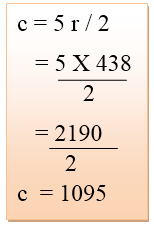
Equation 2 -----------------------------

Pixels, Dots, and Lines per Inchd

Comparing equation 1 and equation 2

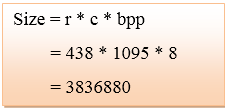


Putting value of r in equation 1

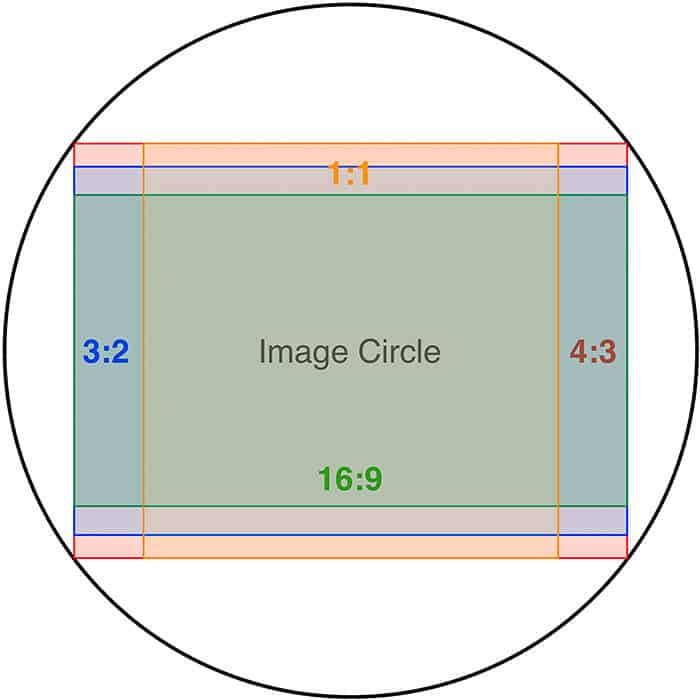


So, row = 438 and column = 1095

**Solution 2:**



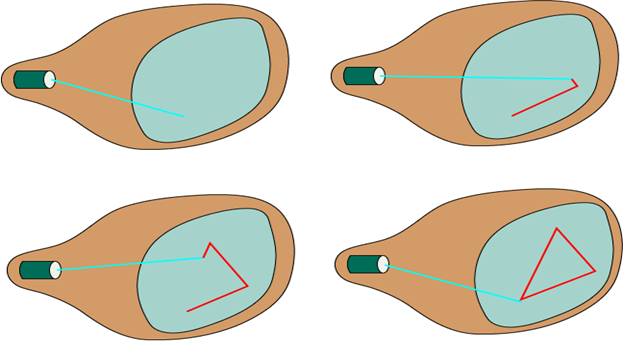
Besides 3:2 aspect ratio, there are other ratios that are gaining popularity like 4:3, 1:1 and 16:9. They all have their own uses. Some are more popular on [social media](https://expertphotography.com/best-instagram-camera/). Others fit better on modern day [monitors](https://expertphotography.com/calibrate-monitor-photography/) when viewed in full screen.  
Also, if you are shooting with the aspect ratio in mind, you can use your camera’s sensor to its maximum potential. You won’t lose details while producing big prints.The reason it's so essential **is** because it influences how **you** compose your photograph and how big it's going to be.



# **Random Scan and Raster Scan Display**

## **Random Scan Display**

Random Scan System uses an electron beam which operates like a pencil to create a line image on the CRT screen. The picture is constructed out of a sequence of straight-line segments. Each line segment is drawn on the screen by directing the beam to move from one point on the screen to the next, where its x & y coordinates define each point. After drawing the picture. The system cycles back to the first line and design all the lines of the image 30 to 60 time each second. The process is shown in fig:



Random-scan monitors are also known as vector displays or stroke-writing displays or calligraphic displays.

## **Advantages:**

1. A CRT has the electron beam directed only to the parts of the screen where an image is to be drawn.
2. Produce smooth line drawings.
3. High Resolution

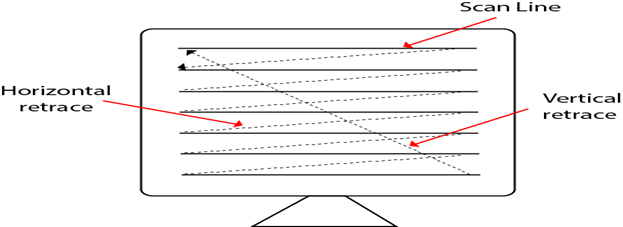
## **Disadvantages:**

1. Random-Scan monitors cannot display realistic shades scenes.

## **Raster Scan Display**

A Raster Scan Display is based on intensity control of pixels in the form of a rectangular box called Raster on the screen. Information of on and off pixels is stored in refresh buffer or Frame buffer. Televisions in our house are based on Raster Scan Method. The raster scan system can store information of each pixel position, so it is suitable for realistic display of objects. Raster Scan provides a refresh rate of 60 to 80 frames per second.

Frame Buffer is also known as Raster or bit map. In Frame Buffer the positions are called picture elements or pixels. Beam refreshing is of two types. First is horizontal retracing and second is vertical retracing. When the beam starts from the top left corner and reaches the bottom right scale, it will again return to the top left side called at vertical retrace. Then it will again more horizontally from top to bottom call as horizontal retracing shown in fig:



**Types of Scanning or travelling of beam in Raster Scan**

1. Interlaced Scanning
2. Non-Interlaced Scanning

In Interlaced scanning, each horizontal line of the screen is traced from top to bottom. Due to which fading of display of object may occur. This problem can be solved by Non-Interlaced scanning. In this first of all odd numbered lines are traced or visited by an electron beam, then in the next circle, even number of lines are located.

For non-interlaced display refresh rate of 30 frames per second used. But it gives flickers. For interlaced display refresh rate of 60 frames per second is used.

### Advantages:

1. Realistic image
2. Million Different colors to be generated
3. Shadow Scenes are possible.

### Disadvantages:

1. Low Resolution
2. Expensive

## **Differentiations between Random and Raster Scan Display**

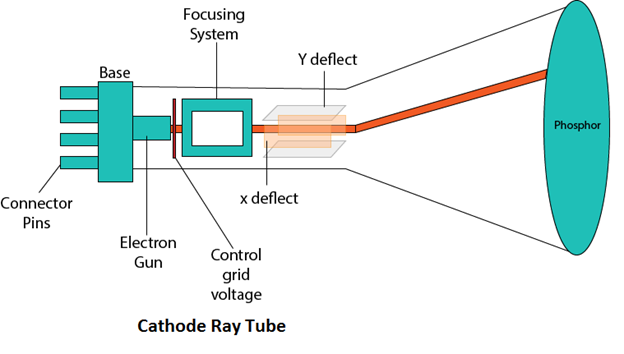
|  |  |
| --- | --- |
| **Random Scan** | **Raster Scan** |
| 1. It has high Resolution | 1. Its resolution is low. |
| 2. It is more expensive | 2. It is less expensive |
| 3. Any modification if needed is easy | 3.Modification is tough |
| 4. Solid pattern is tough to fill | 4.Solid pattern is easy to fill |
| 5. Refresh rate depends or resolution | 5. Refresh rate does not depend on the picture. |
| 6. Only screen with view on an area is displayed. | 6. Whole screen is scanned. |
| 7. Beam Penetration technology come under it. | 7. Shadow mark technology came under this. |
| 8. It does not use interlacing method. | 8. It uses interlacing |
| 9. It is restricted to line drawing applications | 9. It is suitable for realistic display. |

# **Cathode Ray Tube (CRT)**

CRT stands for Cathode Ray Tube. The CRT is a **display screen** which produces images in the form of the **video signal.** It is a type of vacuum tube which displays images when the electron beam through **electron guns** are **strik**es on the **phosphorescent surface**. In other Words, the CRT **generates** the beams, **accelerates** it at high velocity and **deflec**t it for creating the images on the **phosphorous screen** so that the beam becomes **visible.**

CRT is a technology used in traditional computer monitors and televisions. The image on CRT display is created by firing electrons from the back of the tube of phosphorus located towards the front of the screen.

Once the electron heats the phosphorus, they light up, and they are projected on a screen. The color you view on the screen is produced by a blend of red, blue and green light.



## **Components of CRT:**

Main Components of CRT are:

**1. Electron Gun:** Electron gun consisting of a series of elements, primarily a heating filament (heater) and a cathode. The electron gun creates a source of electrons which are focused into a narrow beam directed at the face of the CRT.

**2. Control Electrode:** It is used to turn the electron beam on and off.

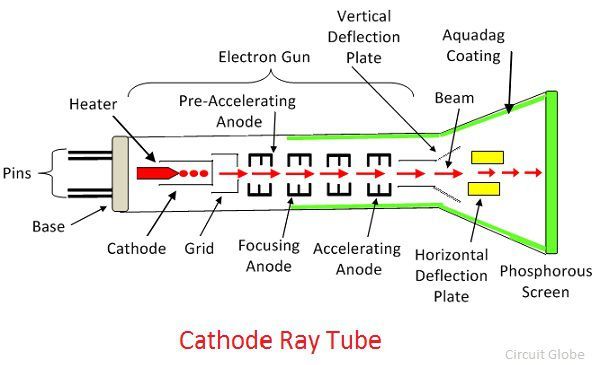
**3. Focusing system:** It is used to create a clear picture by focusing the electrons into a narrow beam.

**4. Deflection Yoke:** It is used to control the direction of the electron beam. It creates an electric or magnetic field which will bend the electron beam as it passes through the area. In a conventional CRT, the yoke is linked to a sweep or scan generator. The deflection yoke which is connected to the sweep generator creates a fluctuating electric or magnetic potential.

**5. Phosphorus-coated screen:** The inside front surface of every CRT is coated with phosphors. Phosphors glow when a high-energy electron beam hits them. Phosphorescence is the term used to characterize the light given off by a phosphor after it has been exposed to an electron beam.

## **Working of CRT**

The working of CRT depends on the movement of electrons beams. The electron guns generate sharply focused electrons which are accelerated at high voltage. This high-velocity electron beam when strikes on the fluorescent screen creates luminous spot

[](https://circuitglobe.com/wp-content/uploads/2017/06/cathode-ray-tube.jpg)

After exiting from the electron gun, the beam passes through the pairs of electrostatic deflection plate. These plates deflected the beams when the voltage applied across it. The one pair of plate moves the beam upward and the second pair of plate moves the beam from one side to another. The horizontal and vertical movement of the electron are independent of each other, and hence the electron beam positioned anywhere on the screen.

The working parts of a CRT are enclosed in a vacuum glass envelope so that the emitted electron can easily move freely from one end of the tube to the other.

1. **frame buffer**

A frame buffer is a large, contiguous piece of [computer](https://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) [memory](https://ecomputernotes.com/fundamental/input-output-and-memory/memory). At a minimum  there is one [memory](https://ecomputernotes.com/fundamental/input-output-and-memory/memory) bit for each pixel in the rater; this amount of memory is called a bit  plane. The picture is built up in the frame buffer one bit at a time.

A framebuffer (frame buffer, or sometimes framestore) is a portion of [random-access memory](https://en.wikipedia.org/wiki/Random-access_memory) (RAM) containing a [bitmap](https://en.wikipedia.org/wiki/Bitmap) that drives a video display. It is a [memory buffer](https://en.wikipedia.org/wiki/Data_buffer) containing data representing all the [pixels](https://en.wikipedia.org/wiki/Pixel) in a complete [video frame](https://en.wikipedia.org/wiki/Video_frame). Modern [video cards](https://en.wikipedia.org/wiki/Video_card) contain framebuffer circuitry in their cores. This circuitry converts an in-memory bitmap into a [video signal](https://en.wikipedia.org/wiki/Video_signal) that can be displayed on a computer monitor.

In [computing](https://en.wikipedia.org/wiki/Computing), a screen buffer is a part of [computer memory](https://en.wikipedia.org/wiki/Computer_memory) used by a computer application for the representation of the content to be shown on the [computer display](https://en.wikipedia.org/wiki/Computer_display). The screen buffer may also be called the video buffer, the regeneration buffer, or regen buffer for short. Screen buffers should be distinguished from [video memory](https://en.wikipedia.org/wiki/Video_memory). To this end, the term off-screen buffer is also used.

1. **Solution**

The size of screen =8 inch by 10 inch

Resolution of each direction =100 pixels

Number of pixel =8\* 100 \* 10 \*100

=800\*1000

=800,000

Memory (storage) = Number of pixel \*1 byte=800,000byte

The frame buffer address is at (0,0) in (x,y) coordinate

1. **Basic 2D Geometric Transformation**

Transformation means changing some graphics into something else by applying rules. We can have various types of transformations such as translation, scaling up or down, rotation, shearing, etc. When a transformation takes place on a 2D plane, it is called 2D transformation.

Transformations play an important role in computer graphics to reposition the graphics on the screen and change their size or orientation.

## **Homogenous Coordinates**

To perform a sequence of transformation such as translation followed by rotation and scaling, we need to follow a sequential process −

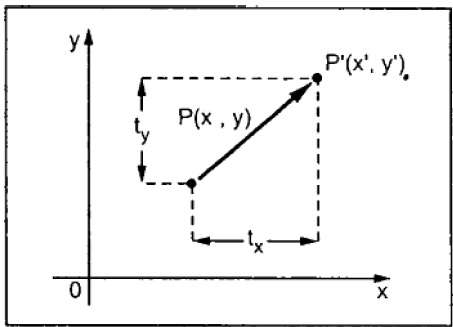
* Translate the coordinates,
* Rotate the translated coordinates, and then
* Scale the rotated coordinates to complete the composite transformation.

To shorten this process, we have to use 3×3 transformation matrix instead of 2×2 transformation matrix. To convert a 2×2 matrix to 3×3 matrix, we have to add an extra dummy coordinate W.

In this way, we can represent the point by 3 numbers instead of 2 numbers, which is called **Homogenous Coordinate** system. In this system, we can represent all the transformation equations in matrix multiplication. Any Cartesian point PX,YX,Y can be converted to homogenous coordinates by P’ (Xh, Yh, h).

## **Translation**

A translation moves an object to a different position on the screen. You can translate a point in 2D by adding translation coordinate (tx, ty) to the original coordinate X,YX,Y to get the new coordinate X′,Y′X′,Y′.



From the above figure, you can write that −

**X’ = X + tx**

**Y’ = Y + ty**

The pair (tx, ty) is called the translation vector or shift vector. The above equations can also be represented using the column vectors.

P=[X][Y]P=[X][Y] p' = [X′][Y′][X′][Y′]T = [tx][ty][tx][ty]

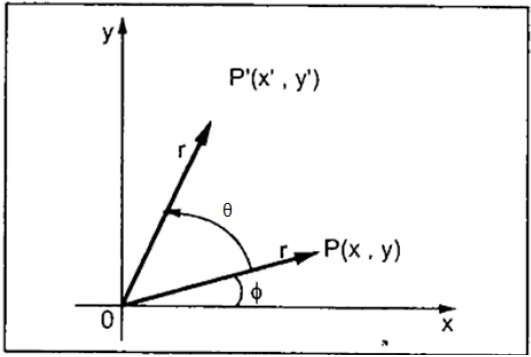
We can write it as −

**P’ = P + T**

## **Rotation**

In rotation, we rotate the object at particular angle θ thetatheta from its origin. From the following figure, we can see that the point PX,YX,Y is located at angle φ from the horizontal X coordinate with distance r from the origin.

Let us suppose you want to rotate it at the angle θ. After rotating it to a new location, you will get a new point P’ X′,Y′X′,Y′.



Using standard trigonometric the original coordinate of point PX,YX,Y can be represented as −

X=rcosϕ......(1)X=rcosϕ......(1)

Y=rsinϕ......(2)Y=rsinϕ......(2)

Same way we can represent the point P’ X′,Y′X′,Y′ as −

x′=rcos(ϕ+θ)=rcosϕcosθ−rsinϕsinθ.......(3)x′=rcos(ϕ+θ)=rcosϕcosθ−rsinϕsinθ.......(3)

y′=rsin(ϕ+θ)=rcosϕsinθ+rsinϕcosθ.......(4)y′=rsin(ϕ+θ)=rcosϕsinθ+rsinϕcosθ.......(4)

Substituting equation 11 & 22 in 33 & 44 respectively, we will get

x′=xcosθ−ysinθx′=xcosθ−ysinθ

y′=xsinθ+ycosθy′=xsinθ+ycosθ

Representing the above equation in matrix form,

[X′Y′]=[XY][cosθ−sinθsinθcosθ]OR[X′Y′]=[XY][cosθsinθ−sinθcosθ]OR

P’ = P . R

Where R is the rotation matrix

R=[cosθ−sinθsinθcosθ]R=[cosθsinθ−sinθcosθ]

The rotation angle can be positive and negative.

For positive rotation angle, we can use the above rotation matrix. However, for negative angle rotation, the matrix will change as shown below −

R=[cos(−θ)−sin(−θ)sin(−θ)cos(−θ)]R=[cos(−θ)sin(−θ)−sin(−θ)cos(−θ)]

=[cosθsinθ−sinθcosθ](∵cos(−θ)=cosθandsin(−θ)=−sinθ)=[cosθ−sinθsinθcosθ](∵cos(−θ)=cosθandsin(−θ)=−sinθ)

## **Scaling**

To change the size of an object, scaling transformation is used. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result.

Let us assume that the original coordinates are X,YX,Y, the scaling factors are (SX, SY), and the produced coordinates are X′,Y′X′,Y′. This can be mathematically represented as shown below −

**X' = X . SX and Y' = Y . SY**

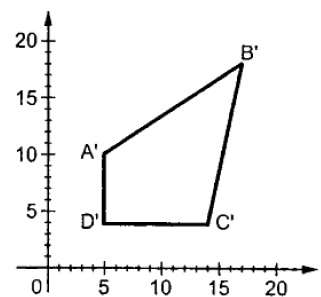
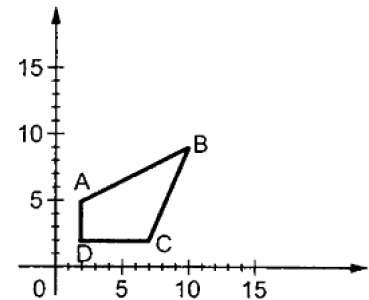
The scaling factor SX, SY scales the object in X and Y direction respectively. The above equations can also be represented in matrix form as below −

(X′Y′)=(XY)[Sx00Sy](X′Y′)=(XY)[Sx00Sy]

OR

**P’ = P . S**

Where S is the scaling matrix. The scaling process is shown in the following figure.

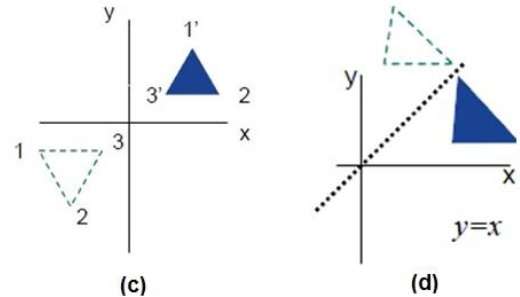
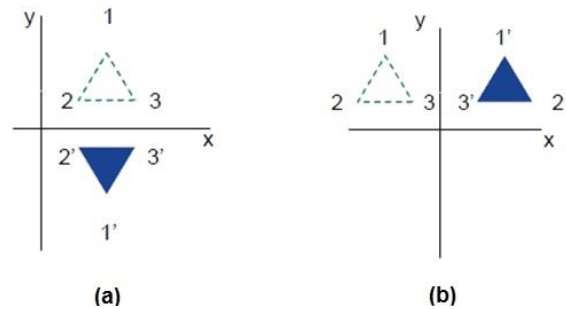


If we provide values less than 1 to the scaling factor S, then we can reduce the size of the object. If we provide values greater than 1, then we can increase the size of the object.

## **Reflection**

Reflection is the mirror image of original object. In other words, we can say that it is a rotation operation with 180°. In reflection transformation, the size of the object does not change.

The following figures show reflections with respect to X and Y axes, and about the origin respectively.

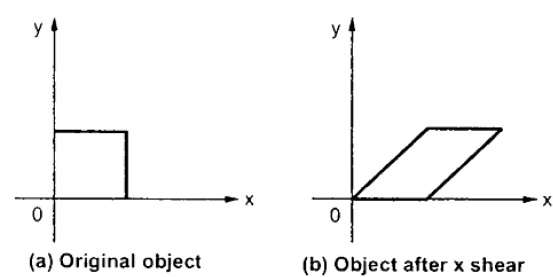


## **Shear**

A transformation that slants the shape of an object is called the shear transformation. There are two shear transformations **X-Shear** and **Y-Shear**. One shifts X coordinates values and other shifts Y coordinate values. However; in both the cases only one coordinate changes its coordinates and other preserves its values. Shearing is also termed as **Skewing**.

### X-Shear

The X-Shear preserves the Y coordinate and changes are made to X coordinates, which causes the vertical lines to tilt right or left as shown in below figure.



The transformation matrix for X-Shear can be represented as −

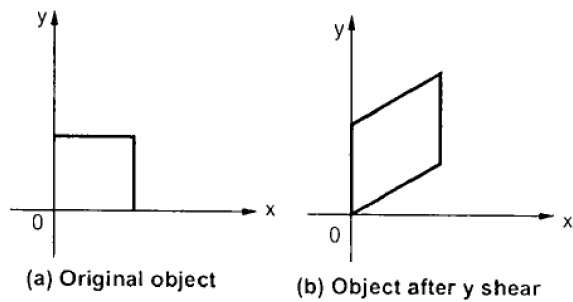
Xsh=⎡⎣⎢100shx10001⎤⎦⎥Xsh=[1shx0010001]

Y' = Y + Shy . X

X’ = X

### Y-Shear

The Y-Shear preserves the X coordinates and changes the Y coordinates which causes the horizontal lines to transform into lines which slopes up or down as shown in the following figure.



The Y-Shear can be represented in matrix from as −

Ysh⎡⎣⎢1shy0010001⎤⎦⎥Ysh[100shy10001]

X’ = X + Shx . Y

Y’ = Y

## **Composite Transformation**

If a transformation of the plane T1 is followed by a second plane transformation T2, then the result itself may be represented by a single transformation T which is the composition of T1 and T2 taken in that order. This is written as T = T1∙T2.

Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.

A combined matrix −

**[T][X] = [X] [T1] [T2] [T3] [T4] …. [Tn]**

Where [Ti] is any combination of

* Translation
* Scaling
* Shearing
* Rotation
* Reflection

The change in the order of transformation would lead to different results, as in general matrix multiplication is not cumulative, that is [A] . [B] ≠ [B] . [A] and the order of multiplication. The basic purpose of composing transformations is to gain efficiency by applying a single composed transformation to a point, rather than applying a series of transformation, one after another.

For example, to rotate an object about an arbitrary point (Xp, Yp), we have to carry out three steps −

* Translate point (Xp, Yp) to the origin.
* Rotate it about the origin.
* Finally, translate the center of rotation back where it belonged.

1. **Basic concept in line drawing**

**The Line drawing algorithm is a graphical algorithm which is used to represent the line segment on discrete graphical media, i.e., printer and pixel-based media.** A line contains two points. The point is an important element of a line.

### Properties of a Line Drawing Algorithm

There are the following properties of a good Line Drawing Algorithm.

* **An algorithm should be precise:**Each step of the algorithm must be adequately defined.
* **Finiteness:**An algorithm must contain finiteness. It means the algorithm stops after the execution of all steps.
* **Easy to understand:**An algorithm must help learners to understand the solution in a more natural way.
* **Correctness:**An algorithm must be in the correct manner.
* **Effectiveness:**The steps of an algorithm must be valid and efficient.
* **Uniqueness:**All steps of an algorithm should be clearly and uniquely defined, and the result should be based on the given input.
* **Input:** A good algorithm must accept at least one or more input.
* **Output:**An algorithm must generate at least one output.

In computer graphics, a **line drawing** algorithm is an algorithm for approximating a **line** segment on discrete graphical media, such as pixel-based displays and printers. On such media, **line drawing** requires an approximation (in nontrivial cases). **Basic** algorithms rasterize **lines** in one color.

### Equation of the straight line

We can define a straight line with the help of the following equation.

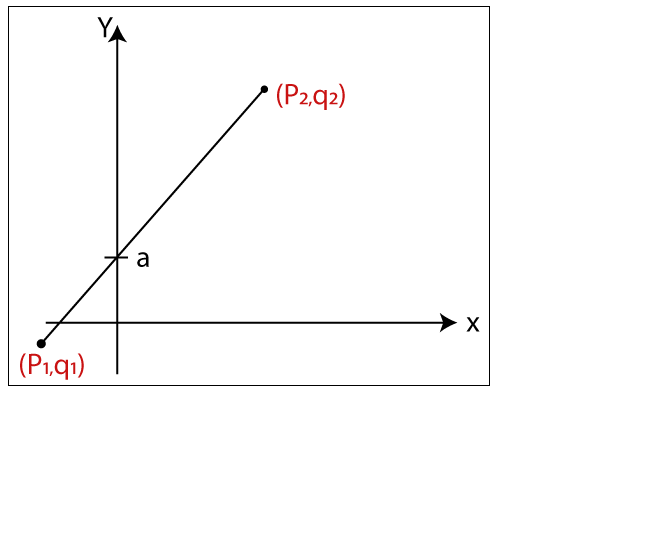
**y= mx + a**

Where,

**(x, y**) = axis of the line.

**m**= Slope of the line.

**a =**Interception point



Let us assume we have two points of the line (**p1, q1**) and (**p2, q2**).

Now, we will put values of the two points in straight line equation, and we get

**y = mx + a**

**q2= mp2                                                                                                       …(1)**

**q1= mp1 + a                                                                                                 …(2)**

We have from equation (1) and (2)

**q2– q1 = mp2­– mp1**

**q2– q1= m (p2­–p1)**

The value of **m = (q2– q1)/ (p2­–p1)**

**m = ▲q / ▲p**

### Algorithms of Line Drawing

 There are  following algorithms used for drawing a line:

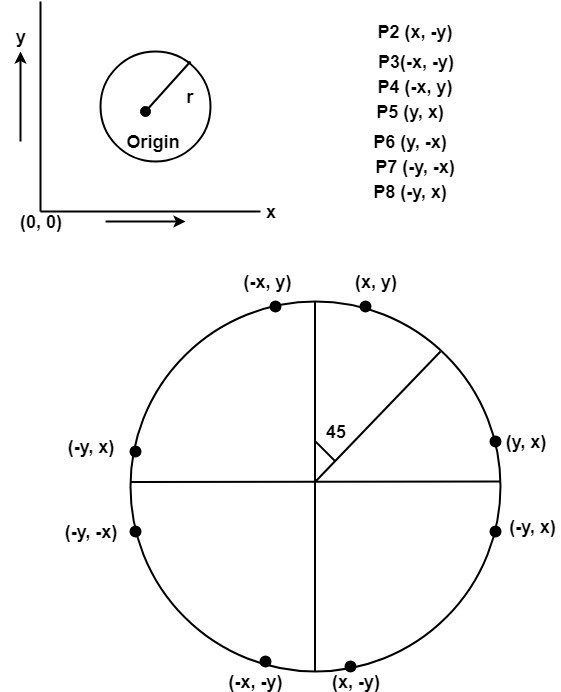
* **DDA (Digital Differential Analyzer) Line Drawing Algorithm**
* **Bresenham’s Line Drawing Algorithm**
* **Mid-Point Line Drawing Algorithm**

1. **Circle generation**

# **Defining a Circle:**

Circle is an eight-way symmetric figure. The shape of circle is the same in all quadrants. In each quadrant, there are two octants. If the calculation of the point of one octant is done, then the other seven points can be calculated easily by using the concept of eight-way symmetry.

For drawing, circle considers it at the origin. If a point is P1(x, y), then the other seven points will be



So we will calculate only 45°arc. From which the whole circle can be determined easily.

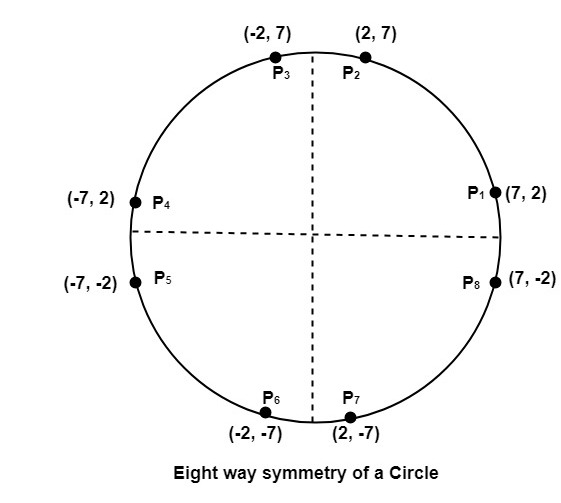
If we want to display circle on screen then the putpixel function is used for eight points as shown below:

          putpixel (x, y, color)  
          putpixel (x, -y, color)  
          putpixel (-x, y, color)  
          putpixel (-x, -y, color)  
          putpixel (y, x, color)  
          putpixel (y, -x, color)  
          putpixel (-y, x, color)  
          putpixel (-y, -x, color)

**Example:** Let we determine a point (2, 7) of the circle then other points will be (2, -7), (-2, -7), (-2, 7), (7, 2), (-7, 2), (-7, -2), (7, -2)

These seven points are calculated by using the property of reflection. The reflection is accomplished in the following way:

The reflection is accomplished by reversing x, y co-ordinates.



There are two standards methods of mathematically defining a circle centered at the origin.

1. Defining a circle using Polynomial Method
2. Defining a circle using Polar Co-ordinates

# **Bresenham’s circle drawing algorithm**

It is not easy to display a continuous smooth arc on the computer screen as our computer screen is made of pixels organized in matrix form. So, to draw a circle on a computer screen we should always choose the nearest pixels from a printed pixel so as they could form an arc. There are two algorithm to do this:

1. [Mid-Point circle drawing algorithm](https://www.geeksforgeeks.org/mid-point-circle-drawing-algorithm/)
2. Bresenham’s circle drawing algorithm  
   Both of these algorithms uses the key feature of circle that it is highly symmetric. So, for whole 360 degree of circle we will divide it in 8-parts each octant of 45 degree. In order to that we will use Bresenham’s Circle Algorithm for calculation of the locations of the pixels in the first octant of 45 degrees. It assumes that the circle is centered on the origin. So for every pixel (x, y) it calculates, we draw a pixel in each of the 8 octants of the circle as shown below :  
   circle 1

Now, we will see how to calculate the next pixel location from a previously known pixel location (x, y). In Bresenham’s algorithm at any point (x, y) we have two option either to choose the next pixel in the east i.e. (x+1, y) or in the south east i.e. (x+1, y-1).

circle 2

And this can be decided by using the decision parameter d as:

* If d > 0, then (x+1, y-1) is to be chosen as the next pixel as it will be closer to the arc.
* else (x+1, y) is to be chosen as next pixel.

Now to draw the circle for a given radius ‘r’ and centre (xc, yc) We will start from (0, r) and move in first quadrant till x=y (i.e. 45 degree). We should start from listed initial condition:

d = 3 - (2 \* r)

x = 0

y = r

Now for each pixel, we will do the following operations:

1. Set initial values of (xc, yc) and (x, y)
2. Set decision parameter d to d = 3 – (2 \* r).
3. call drawCircle(int xc, int yc, int x, int y) function.
4. Repeat steps 5 to 8 until x < = y
5. Increment value of x.
6. If d < 0, set d = d + (4\*x) + 6
7. Else, set d = d + 4 \* (x – y) + 10 and decrement y by 1.
8. call drawCircle(int xc, int yc, int x, int y) function

**drawCircle() function:**

filter\_none,edit,play\_arrow,brightness\_4

|  |
| --- |
| // function to draw all other 7 pixels  // present at symmetric position  drawCircle(int xc, int yc, int x, int y)  {      putpixel(xc+x, yc+y, RED);      putpixel(xc-x, yc+y, RED);      putpixel(xc+x, yc-y, RED);      putpixel(xc-x, yc-y, RED);      putpixel(xc+y, yc+x, RED);      putpixel(xc-y, yc+x, RED);      putpixel(xc+y, yc-x, RED);      putpixel(xc-y, yc-x, RED);  } |

**Below is C implementation of above approach.**

filter\_none,edit,play\_arrow,brightness\_4

|  |
| --- |
| // C-program for circle drawing  // using Bresenham’s Algorithm  // in computer-graphics  #include <stdio.h>  #include <dos.h>  #include <graphics.h>    // Function to put pixels  // at subsequence points  void drawCircle(int xc, int yc, int x, int y)  {      putpixel(xc+x, yc+y, RED);      putpixel(xc-x, yc+y, RED);      putpixel(xc+x, yc-y, RED);      putpixel(xc-x, yc-y, RED);      putpixel(xc+y, yc+x, RED);      putpixel(xc-y, yc+x, RED);      putpixel(xc+y, yc-x, RED);      putpixel(xc-y, yc-x, RED);  }    // Function for circle-generation  // using Bresenham's algorithm  void circleBres(int xc, int yc, int r)  {      int x = 0, y = r;      int d = 3 - 2 \* r;      drawCircle(xc, yc, x, y);      while (y >= x)      {          // for each pixel we will          // draw all eight pixels            x++;            // check for decision parameter          // and correspondingly          // update d, x, y          if (d > 0)          {              y--;              d = d + 4 \* (x - y) + 10;          }          else              d = d + 4 \* x + 6;          drawCircle(xc, yc, x, y);          delay(50);      }  }      // driver function  int main()  {      int xc = 50, yc = 50, r2 = 30;      int gd = DETECT, gm;      initgraph(&gd, &gm, "");  // initialize graph      circleBres(xc, yc, r);    // function call      return 0;  } |

Advantages

* It is simple agorithm.
* It can be implement easily
* It is totally based on the equation of circle

Disadvantage

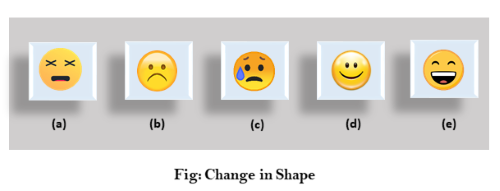
* There is the problem of accuracy while generating points.
* This algorithm is not suitable for the complex and high graphic images

# **Animation**

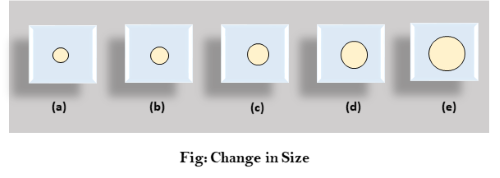
Animation refers to the movement on the screen of the display device created by displaying a sequence of still images. Animation is the technique of designing, drawing, making layouts and preparation of photographic series which are integrated into the multimedia and gaming products. Animation connects the exploitation and management of still images to generate the illusion of movement. A person who creates animations is called animator. He/she use various computer technologies to capture the pictures and then to animate these in the desired sequence.

Animation includes all the visual changes on the screen of display devices. These are:

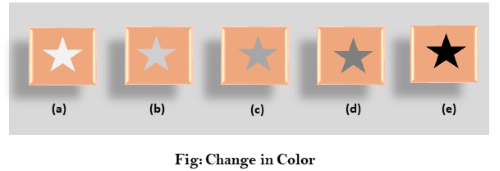
1. Change of shape as shown in fig:



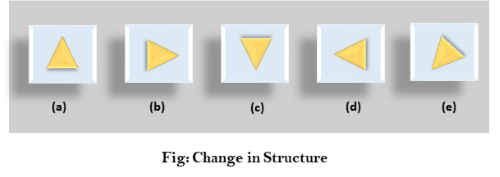
2. Change in size as shown in fig:



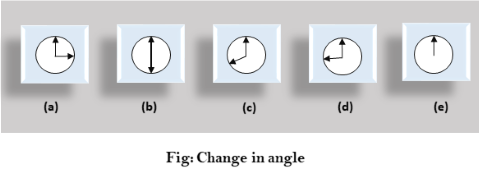
3. Change in color as shown in fig:



4. Change in structure as shown in fig:



5. Change in angle as shown in fig:



Animation is not the art of making drawings move, but the art of drawing movement. Based on the persistence of vision, animation, like all film, is an illusion of fluid movement, when in fact it is a series of static drawings moving so quickly, twenty-four frames a second, that they give the appearance of movement. While there are many forms of animation, we are going to focus on the three most prominent.

# **Application Areas of Animation**

**1. Education and Training:** Animation is used in school, colleges and training centers for education purpose. Flight simulators for aircraft are also animation based.

**2. Entertainment:** Animation methods are now commonly used in making motion pictures, music videos and television shows, etc.

**3. Computer Aided Design (CAD):** One of the best applications of computer animation is Computer Aided Design and is generally referred to as CAD. One of the earlier applications of CAD was automobile designing. But now almost all types of designing are done by using CAD application, and without animation, all these work can't be possible.

**4. Advertising:** This is one of the significant applications of computer animation. The most important advantage of an animated advertisement is that it takes very less space and capture people attention.

**5. Presentation:** Animated Presentation is the most effective way to represent an idea. It is used to describe financial, statistical, mathematical, scientific & economic data.