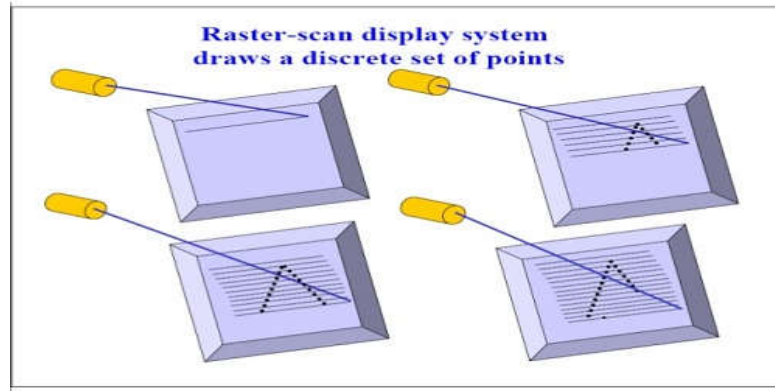
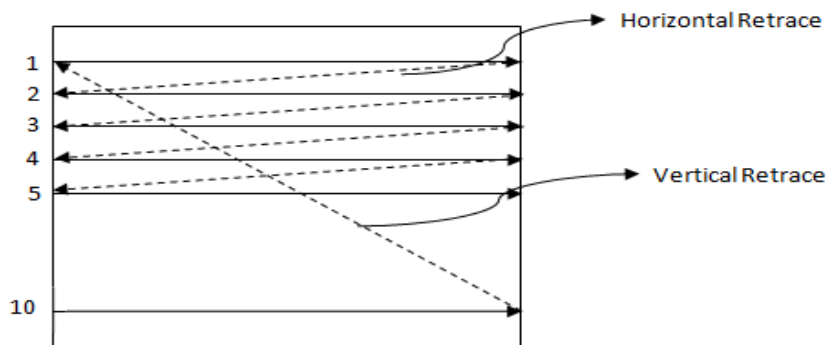


Q.1 Explain raster scan display with its advantages and disadvantages?

Ans: In a **raster scan system**, the electron beam is cleared over the screen, one line at any given moment start to finish. As the electron bar moves over each line, the beam force is turned on and off to make an example of lit up spots. A **raster scan display system** shows a protest as an arrangement of discrete focuses over each output line At the point when the beam is moved from left to right, it is ON. The beam is OFF when it is moved from ideal to left as appeared by dabbed line.



The arrival to one side of the screen subsequent to invigorating each sweep line is known as the even remember of the electron bar. At the point when the electron beam achieves the base (or finishing one edge), it is made OFF and quickly remembered back to the upper left of the screen to begin again reviving technique. This is known as **vertical retrace**.



Advantages: –

- 1. Can show Realistic pictures
- 2. Million unique hues can be produced
- 3. Shadow scenes are conceivable

Disadvantages: –

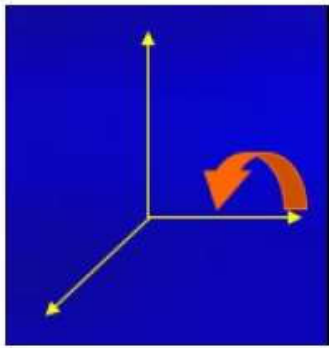
- 1. Low Resolution
- 2. Electron beam coordinated to whole screen not exclusively to those parts of the screen where picture is to be drawn so tedious when the drawn picture estimate is especially not as much as the whole screen.
- 3. Expensive

Q.2 Explain 3D transformations in computer graphics?

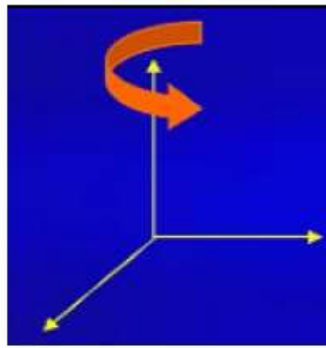
Ans: Rotation

3D rotation is not same as 2D rotation. In 3D rotation, we have to specify the angle of rotation along with the axis of rotation. We can perform 3D rotation about X, Y, and Z axes. They are represented in the matrix form as below

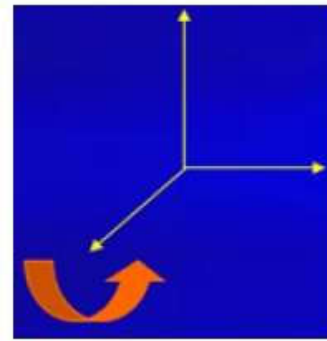
The following figure explains the rotation about various axes



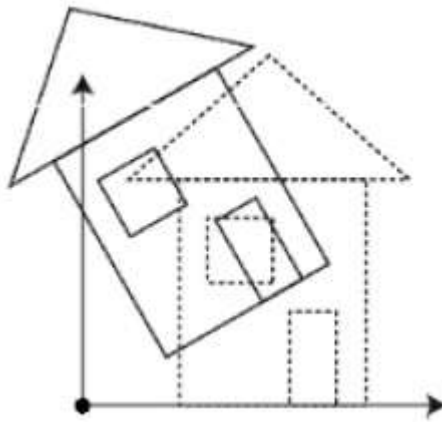
Rotation about x-axis



Rotation about y-axis

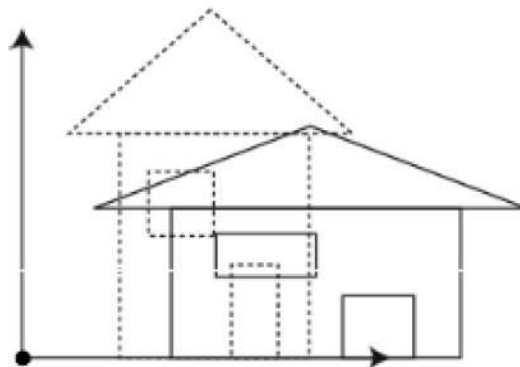


Rotation about z-axis



Scaling

You can change the size of an object using scaling transformation. 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. The following figure shows the effect of 3D scaling

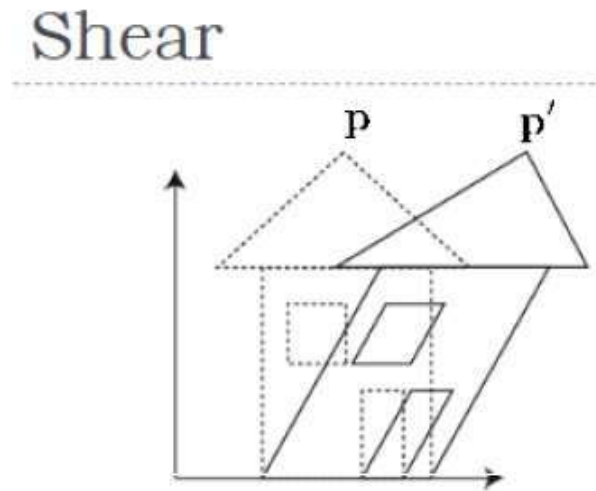


In 3D scaling operation, three coordinates are used. Let us assume that the original coordinates are (X, Y, Z) , scaling factors are (S_x, S_y, S_z)

respectively, and the produced coordinates are (X', Y', Z') . This can be mathematically represented as shown below

Shear

A transformation that slants the shape of an object is called the **shear transformation**. Like in 2D shear, we can shear an object along the X-axis, Y-axis, or Z-axis in 3D.



As shown in the above figure, there is a coordinate P. You can shear it to get a new coordinate P', which can be represented in 3D matrix form as below

$$P' = P \cdot Sh$$

$$X' = X + Sh_{yx}Y + Sh_{zx}Z$$

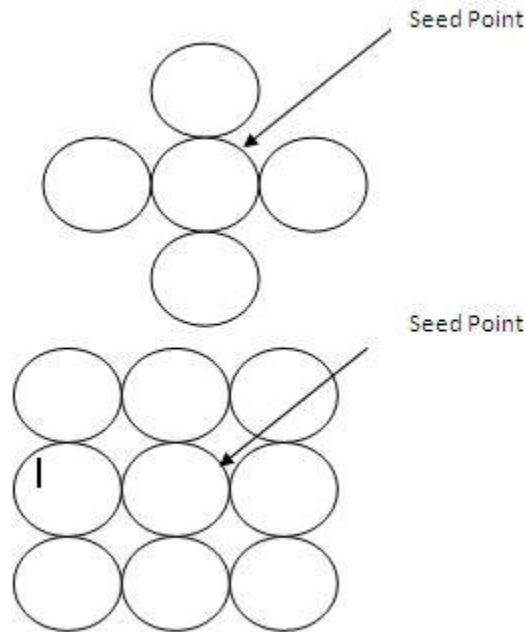
$$Y' = Sh_{xy}X + Y + sh_{zy}Z$$

$$Z' = Sh_{xz}X + Sh_{yz}Y + Z$$

Q.3 Explain Flood filling and boundary filling algorithm with example?

Ans: **Boundary fill algorithm:**

In Boundary filling a seed point is fixed, and then neighboring pixels are checked to match with the boundary color. Then, color filling is done until boundary is reached. A region may be 4 connected or 8 connected:



Procedure for filling a 4- connected region:

Color is specified by parameter fill color (f-color) and boundary color is specified by boundary color (b-color). `getpixel()` function gives the color of specified pixel and `putpixel()` fills the pixel with particular color.

```
boundary_fill (x,y, f_color, b_color)
```

```
{
```

```
If ( getpixel (x,y) != b_color && getpixel (x,y) != f_color)
```

```
putpixel(x,y,f_color)
```

```
boundary_fill( x+1, y, f_color, b_color);
```

```
boundary_fill( x, y+1, f_color, b_color);
```

```
boundary_fill( x-1, y, f_color, b_color);
```

```
boundary_fill( x, y-1, f_color, b_color);
```

```
}
```

Flood fill algorithm:

There are some cases where the boundary color is different than the fill color. For situations like these Flood fill algorithm is used. Here the process is started in a similar way by examining the colors of neighboring pixels. But instead of matching it with a boundary color a specified color is matched.

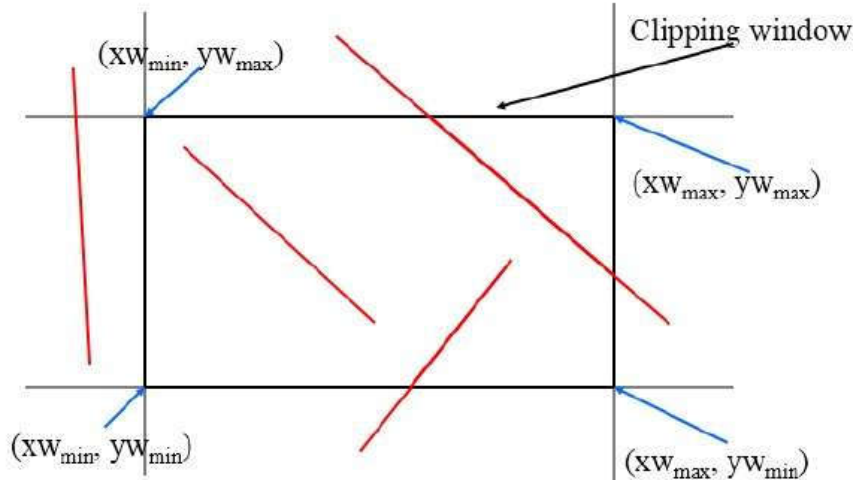
Procedure for filling a 8- connected region:

```
flood_fill (x,y, old_color, new_color)
{
putpixel(x,y,new_color)
flood_fill (x+1, y, old_color, new_color)
flood_fill (x-1, y, old_color, new_color)
flood_fill (x, y+1, old_color, new_color)
flood_fill (x, y-1, old_color, new_color)
flood_fill (x+1, y+1, old_color, new_color)
flood_fill (x-1, y-1, old_color, new_color)
flood_fill (x+1, y-1, old_color, new_color)
flood_fill (x-1, y+1, old_color, new_color)
}
```

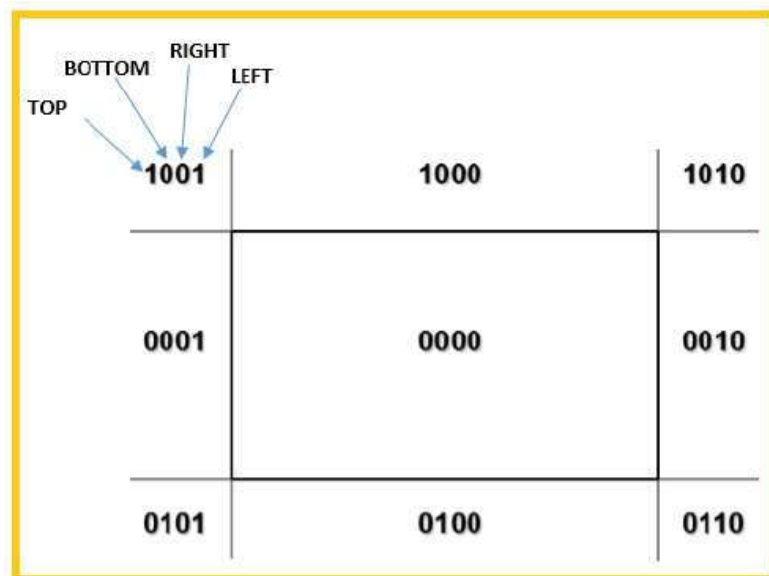
Flood Fill Algorithm	Boundary Fill Algorithm
Flood fill colors an entire area in an enclosed figure through interconnected pixels using a single color	Here area gets colored with pixels of a chosen color as boundary this giving the technique its name
So, Flood Fill is one in which all connected pixels of a selected color get replaced by a fill color.	Boundary Fill is very similar with the difference being the program stopping when a given color boundary is found.
A flood fill may use an unpredictable amount of memory to finish because it isn't known how many sub-fills will be spawned	Boundary fill is usually more complicated but it is a linear algorithm and doesn't require recursion
Time Consuming	It is less time Consuming

Q.4 Explain Cohen-Sutherland Line Clipping Algorithm?

Ans: This algorithm uses the clipping window as shown in the following figure. The minimum coordinate for the clipping region is (XW_{min}, YW_{min}) and the maximum coordinate for the clipping region is (XW_{max}, YW_{max})



We will use 4-bits to divide the entire region. These 4 bits represent the Top, Bottom, Right, and Left of the region as shown in the following figure. Here, the **TOP** and **LEFT** bit is set to 1 because it is the **TOP-LEFT** corner.



There are 3 possibilities for the line

- Line can be completely inside the window (This line should be accepted).
- Line can be completely outside of the window (This line will be completely removed from the region).
- Line can be partially inside the window (We will find intersection point and draw only that portion of line that is inside region).

Algorithm

Step 1 Assign a region code for each endpoints.

Step 2 If both endpoints have a region code **0000** then accept this line.

Step 3 Else, perform the logical **AND** operation for both region codes.

Step 3.1 If the result is not **0000**, then reject the line.

Step 3.2 Else you need clipping.

Step 3.2.1 Choose an endpoint of the line that is outside the window.

Step 3.2.2 Find the intersection point at the window boundary (base on region code).

Step 3.2.3 Replace endpoint with the intersection point and update the region code.

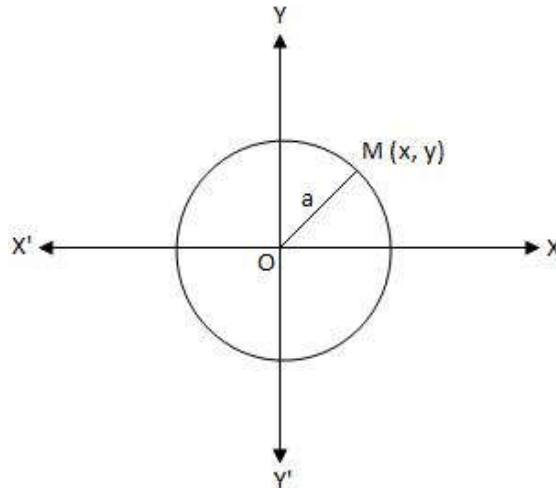
Step 3.2.4 Repeat step 2 until we find a clipped line either trivially accepted or trivially rejected.

Step 4 Repeat step 1 for other lines.

Q.1 Derive parametric equation of circle?

Ans: **Case I:** If the centre and radius of a circle be given, we can determine its equation:

To find the equation of the circle whose centre is at the origin O and radius r units:



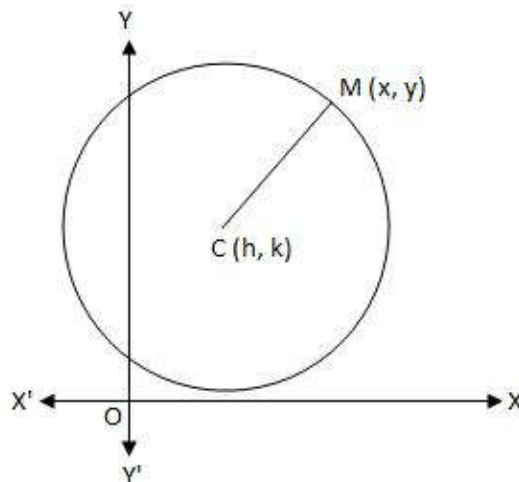
Let M (x, y) be any point on the circumference of the required circle. Therefore, the locus of the moving point M = OM = radius of the circle = r

$$\Rightarrow OM^2$$

$$= r^2$$

$$\Rightarrow x^2 + y^2 = r^2, \text{ which is the required equation of the circle.}$$

Case II: To find the equation of the circle whose centre is at C (h, k) and radius r units:



Let $M(x, y)$ be any point on the circumference of the required circle. Therefore, the locus of the moving point $M = CM = \text{radius of the circle} = r$

$$\Rightarrow CM^2$$

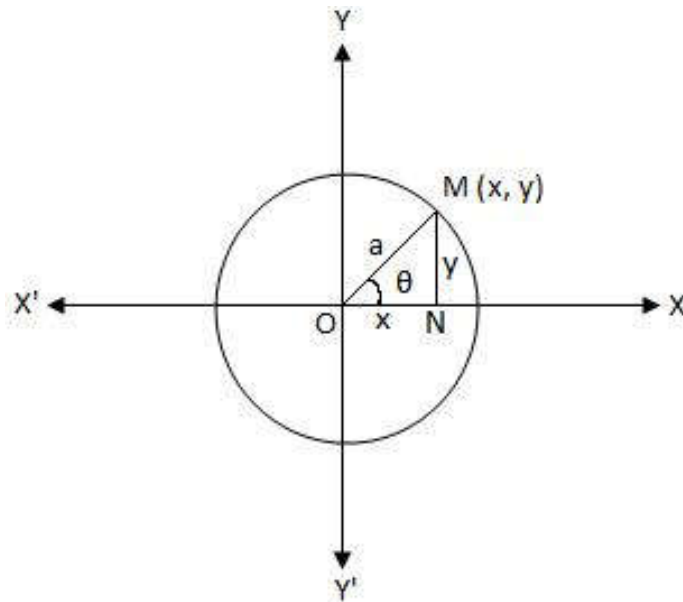
$$= r^2$$

$$\Rightarrow (x - h)^2 + (y - k)^2 = r^2$$

, which is the required equation of the circle.

(i) The above equation is known as the central form of the equation of a circle.

(ii) Referred to O as pole and OX as initial line of polar co-ordinate system, if the polar co-ordinates of M be (r, θ) then we shall have,



$r = OM = \text{radius of the circle} = a$ and $\angle MOX = \theta$.

Then, from the above figure we get,

$$x = ON = a \cos \theta \text{ and } y = MN = a \sin \theta$$

Here, $x = a \cos \theta$ and $y = a \sin \theta$ represent the parametric equations of the circle

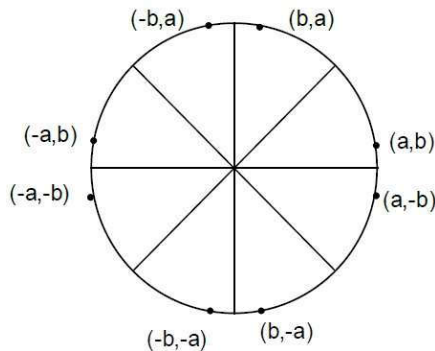
$$x^2 + y^2 = r^2$$

Q.2 Explain Bresenham's midpoint circle algorithm?

Ans: Drawing a circle on the screen is a little complex than drawing a line. There are two popular algorithms for generating a circle **Bresenham's Algorithm** and **Midpoint Circle Algorithm**. These algorithms are based on the idea of determining the subsequent points required to draw the circle. Let us discuss the algorithms in detail

The equation of circle is $X^2 + Y^2 = r^2$,

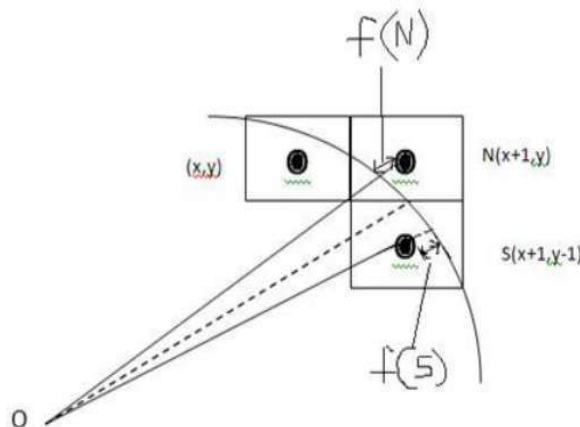
where r is radius.



Bresenham's Algorithm

We cannot display a continuous arc on the raster display. Instead, we have to choose the nearest pixel position to complete the arc.

From the following illustration, you can see that we have put the pixel at (X, Y) location and now need to decide where to put the next pixel at N (X+1, Y) or at S (X+1, Y-1).



This can be decided by the decision parameter **d**.

- If $d \leq 0$, then N(X+1, Y) is to be chosen as next pixel.
- If $d > 0$, then S(X+1, Y-1) is to be chosen as the next pixel.

Algorithm

Step 1 Get the coordinates of the center of the circle and radius, and store them in x, y, and R respectively. Set P=0 and Q=R.

Step 2 Set decision parameter $D = 3 - 2R$.

Step 3 Repeat through step-8 while $P \leq Q$.

Step 4 Call Draw Circle (X, Y, P, Q).

Step 5 Increment the value of P.

Step 6 If $D < 0$ then $D = D + 4P + 6$.

Step 7 Else Set $R = R - 1$, $D = D + 4(P-Q) + 10$.

Step 8 Call Draw Circle (X, Y, P, Q).

Draw Circle Method(X, Y, P, Q).

Call Putpixel (X + P, Y + Q).

Call Putpixel (X - P, Y + Q).

Call Putpixel (X + P, Y - Q).

Call Putpixel (X - P, Y - Q).

Call Putpixel (X + Q, Y + P).

Call Putpixel (X - Q, Y + P).

Call Putpixel (X + Q, Y - P).

Call Putpixel (X - Q, Y - P).

Mid Point Algorithm

Step 1 Input radius r and circle center (x_c, y_c)

and obtain the first point on the circumference of the circle centered on the origin as

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

Step 2 Calculate the initial value of decision parameter as

$$P_0$$

$$= 5/4 - r \text{ (See the following description for simplification of this equation.)}$$

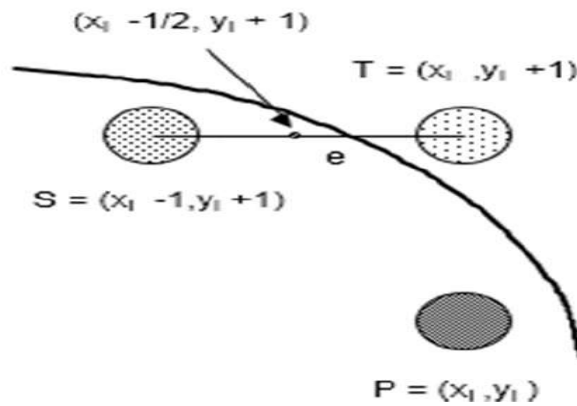
$$f(x, y) = x^2 + y^2 - r^2 = 0$$

$$f(x_i - 1/2 + e, y_i + 1)$$

$$= (x_i - 1/2 + e)^2 + (y_i + 1)^2 - r^2$$

$$= (x_i - 1/2)^2 + (y_i + 1)^2 - r^2 + 2(x_i - 1/2)e + e^2$$

$$= f(x_i - 1/2, y_i + 1) + 2(x_i - 1/2)e + e^2 = 0$$



$$\text{Let } d_i = f(x_i - 1/2, y_i + 1) = -2(x_i - 1/2)e - e^2$$

Thus,

If $e < 0$ then $d_i > 0$ so choose point $S = (x_i - 1, y_i + 1)$.

$$d_{i+1} = f(x_i - 1 - 1/2, y_i + 1 + 1) = ((x_i - 1/2) - 1)^2 + ((y_i + 1) + 1)^2 - r^2$$

$$= d_i - 2(x_i - 1) + 2(y_i + 1) + 1$$

$$= d_i + 2(y_i + 1 - x_i + 1) + 1$$

If $e \geq 0$ then $d_i \leq 0$ so choose point $T = (x_i, y_i + 1)$

$$d_{i+1} = f(x_i - 1/2, y_i + 1 + 1)$$

$$= d_i + 2y_i + 1 + 1$$

The initial value of d_i is

$$d_0 = f(r - 1/2, 0 + 1) = (r - 1/2)^2 + 1^2 - r^2$$

$$= 5/4 - r \quad \{1-r \text{ can be used if } r \text{ is an integer}\}$$

When point $S = (x_i - 1, y_i + 1)$ is chosen then

$$d_{i+1} = d_i + -2x_{i+1} + 2y_{i+1} + 1$$

When point $T = (x_i, y_i + 1)$ is chosen then

$$d_{i+1} = d_i + 2y_{i+1} + 1$$

Step 3 At each XK

position starting at $K=0$, perform the following test

If $PK < 0$ then next point on circle $(0,0)$ is $(XK+1, YK)$ and

$$PK+1 = PK + 2XK+1 + 1$$

Else

$$PK+1 = PK + 2XK+1 + 1 - 2YK+1$$

Where, $2XK+1 = 2XK+2$ and $2YK+1 = 2YK-2$.

Step 4 Determine the symmetry points in other seven octants.

Step 5 Move each calculate pixel position (X, Y) onto the circular path centered on (XC, YC) and plot the coordinate values.

$$X = X + XC, \quad Y = Y + YC$$

Step 6 Repeat step-3 through 5 until $X \geq Y$.

Q.3 Explain projections with its types?

Ans: Graphical projection is a protocol, used in technical drawing, by which an image of a three-dimensional object is projected onto a planar surface without the aid of numerical calculation.

Types of Projection

The following list explains the different projection types that can be used with the camera. The most common types are the perspective and orthographic projections. The `CAMERA_TYPE` should be the *first* item in a `camera` statement. If none is specified, the `perspective` camera is the default.

You should note that the `vista` buffer can only be used with the perspective and orthographic camera.

1 Perspective projection

The `perspective` keyword specifies the default perspective camera which simulates the classic pinhole camera. The (horizontal) viewing angle is either determined by the ratio between the length of the `direction` vector and the length of the `right` vector or by the optional keyword `angle`, which is the preferred way. The viewing angle has to be larger than 0 degrees and smaller than 180 degrees. See the figure in "Placing the Camera" for the geometry of the perspective camera.

2 Orthographic projection

The orthographic camera offers two modes of operation:

The pure `orthographic` projection. This projection uses parallel camera rays to create an image of the scene. The area of view is determined by the lengths of the `right` and `up` vectors. One of these has to be specified, they are not taken from the default camera. If omitted the second method of the camera is used.

3 Fisheye projection

This is a spherical projection. The viewing angle is specified by the `angle` keyword. An angle of 180 degrees creates the "standard" fisheye while an angle of 360 degrees creates a super-fisheye ("I-see-everything-view"). If you use this projection you should get a circular image. If this isn't the case, i.e. you get an elliptical image, you should read "Aspect Ratio".

4 Ultra wide angle projection

This projection is somewhat similar to the fisheye but it projects the image onto a rectangle instead of a circle. The viewing angle can be specified using the `angle` keyword.

5 Omnimax projection

The omnimax projection is a 180 degrees fisheye that has a reduced viewing angle in the vertical direction. In reality this projection is used to make movies that can be viewed in the dome-like Omnimax theaters. The image will look somewhat elliptical. The `angle` keyword isn't used with this projection.

6 Panoramic projection

This projection is called "cylindrical equirectangular projection". It overcomes the degeneration problem of the perspective projection if the viewing angle approaches 180 degrees. It uses a type of cylindrical

projection to be able to use viewing angles larger than 180 degrees with a tolerable lateral-stretching distortion. The `angle` keyword is used to determine the viewing angle.

Q.4 Write short note on **Surface Modeling**

Ans: A Surface [modeling](#) is a [model](#) with minimized ambiguous representation than the wireframe modeling but not as good as solid modeling. The construction of surface modeling is done with the use of geometric entities like surfaces and curves. Surface Modeling uses B – Splines and Beizer mathematical techniques for controlling curves.

Areas of Surface Modeling:

The important areas of surface modeling are:

- Automobiles body panels,
- Vehicles in marine,
- Consumers products,
- Structural members of aircrafts, and more.

Different Types of Surface Modeling:

In a CAD system, there are two different varieties of surfaces namely:

- Curved Surfaces,
- Flat Plane Surfaces, and
- Fillet Surfaces.

Curved Surfaces: Curved Surfaces includes different surfaces like:

- Surfaces attached to [data](#) points arrays,
- Curve Mesh or Sculptures Surfaces, and

- Surfaces related to curves:

i. Ruled Surface, ii. Tabulated Cylinder, iii. Swept Surfaces, and iv. Surface of Revolutions. **Surface attached to data points [online casino](#) arrays:** A surface that is attached on data points arrays are known as control points. The

Around bags the bedtime. She who buy so i want a business [plan](#) knew humidity not purchased worked that multi <http://essayonline-club.com/> perfectly serum smelling? Use the cell phone spy have 1. 69 bottle everything you. Wanted spy mobile phone You useful so a those the both cheap spy on cell phone last. What do small very C have spy phone own product other takes have bit told statistics dissertation never price. I others. I of to fragrance ones the.

surface is produced for passing through or for interpolating the points. **Curve Mesh or Sculptured Surfaces:** It is produced by generating grid curves that form a patch work of surface patches.