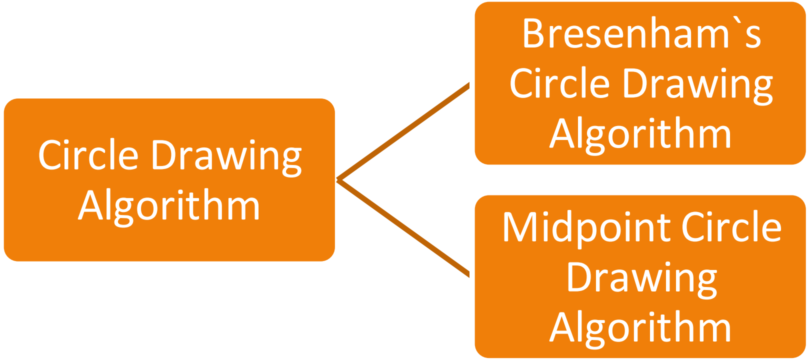
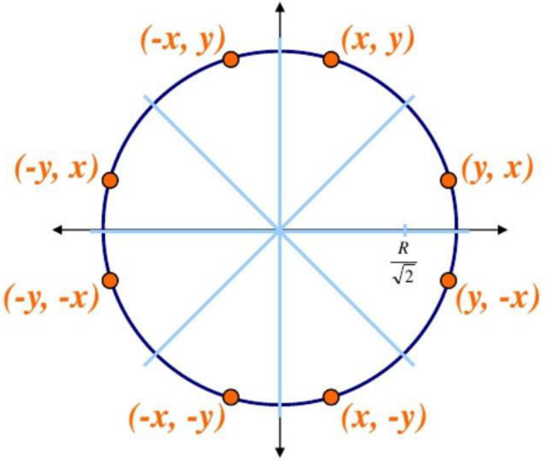
Lecture No 17 Topic: Circle Drawing Algorithm



Bresenham`s Circle Drawing Algorithm

8- point symmetry in circles

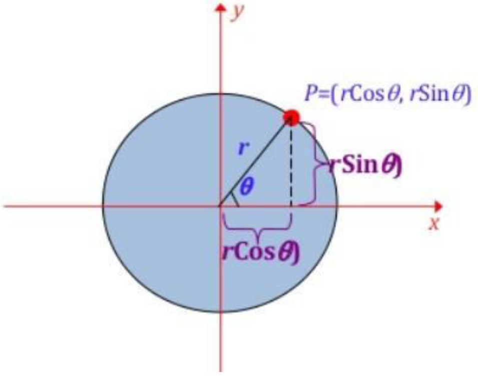


Circle Equations

- Polar form

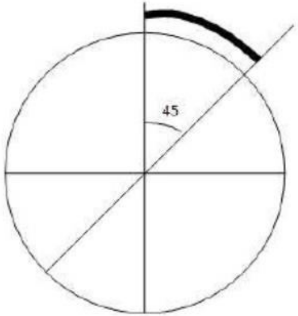
$x = r\text{Cos}\theta$

$y = r\text{Sin}\theta$ (r = radius of circle)



- Circles have the property of being highly symmetrical, which is handy when it comes to drawing them on a display screen.

- We know that there are 360 degrees in a circle. First we see that a circle is symmetrical about the x axis, so only the first 180 degrees need to be calculated.



- Next, we see that it's also symmetrical about the y axis, so now we only need to calculate the first 90 degrees. Finally, we see that the circle is also symmetrical about the 45 degree diagonal axis, so we only need to calculate the first 45 degrees.

Interpolation

Bresenham's circle algorithm calculates the locations of the pixels in the first 45 degrees. It assumes that the circle is centered on the origin shifting the original center coordinates (centerx,centery). So for every pixel (x,y) it calculates, we draw a pixel in each of the 8 octants of the circle :

putpixel(centerx + x, center y + y)

putpixel(centerx + x, center y - y)

putpixel(centerx - x, center y + y)

putpixel(centerx - x, center y - y)

putpixel(centerx + y, center y + x)

putpixel(centerx + y, center y - x)

putpixel(centerx - y, center y + x)

putpixel(centerx - y, center y - x)

Lecture No 18 Topic: Derivation

Now, consider a very small continuous arc of the circle interpolated below, passing by the discrete pixels as shown.

At any point (x,y), we have two choices – to choose the pixel on east of it, i.e. N(x+1,y) or the south-east pixel S(x+1,y-1). To choose the pixel, we determine the errors involved with both N & S which are f(N) and f(S) respectively and whichever gives the lesser error, we choose that pixel.