CS 461 - Computer Graphics

Drawing basic primitives - II

Amal Dev Parakkat

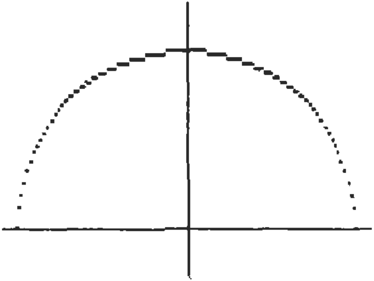
Drawing circles - Idea 1

- Center: (x_c, y_c), Radius: r
- Eq. of circle:

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \tag{1}$$

- Simple solution:

$$y = y_c \pm \sqrt{r^2 - (x - x_c)^2} \tag{2}$$

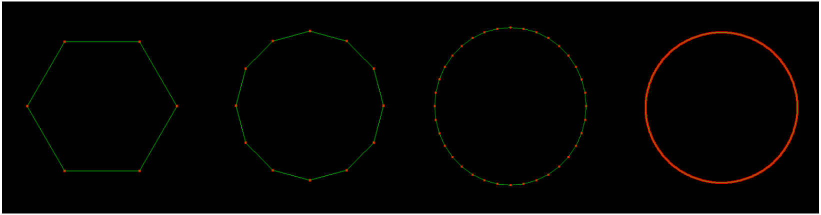


Drawing circles - Idea 2

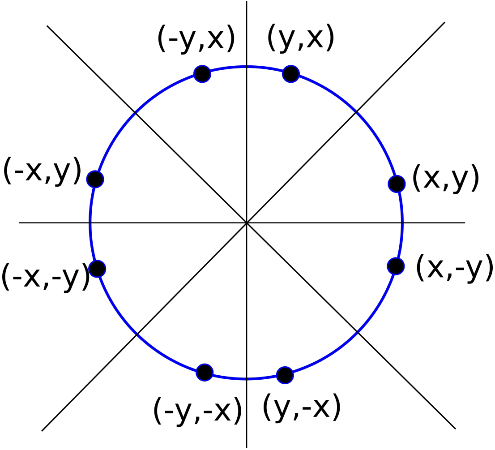
- Use polar coordinates r and θ
- Eq. of circle in parametric polar form:

$$x = x_c + rcos\theta \tag{3}$$

$$y = y_c + rsin\theta \tag{4}$$

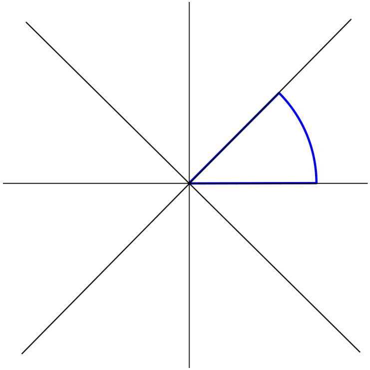


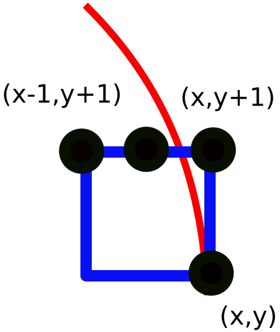
Symmetry of a circle



Midpoint circle drawing - idea

- $F(x,y) = x^2 + y^2 - r^2$
- $F(x,y) = \begin{cases} < 0, \text{ if } (x,y) \text{ is inside the circle} \\ = 0, \text{ if } (x,y) \text{ is on the circle} \\ > 0, \text{ if } (x,y) \text{ is outside the circle} \end{cases}$





Midpoint circle drawing - idea

- $P_k = (x_k - 0.5)^2 + (y_k + 1)^2 - r^2$
- Now P_{k+1} can be $(x_k, y_k + 1)$ or $(x_k - 1, y_k + 1)$
- $P_{k+1} = (x_{k+1} - 0.5)^2 + (y_{k+1} + 1)^2 - r^2$
  $= (x_{k+1} - 0.5)^2 + ((y_k + 1) + 1)^2 - r^2$
  $= (x_{k+1} - 0.5)^2 + (y_k + 1)^2 + 2(y_k + 1) + 1 - r^2$
  $= (x_{k+1} - 0.5)^2 + [-(x_k - 0.5)^2 + (x_k - 0.5)^2] + (y_k + 1)^2 + 2(y_k + 1) + 1 - r^2$
  $= P_k + (x_{k+1} - 0.5)^2 - (x_k - 0.5)^2 + 2(y_k + 1) + 1$
  $=> P_k + 2(y_k + 1) + 1$ (when $x_{k+1} = x_k$)
  $=> P_k + 2(y_k + 1) - 2(x_k - 1) + 1$ (when $x_{k+1} = x_k - 1$)
- Initial point is plotted on (r,0) and P_0 is calculated as:
  $P_0 = (r - 0.5)^2 + (0 + 1)^2 - r^2$
  $= 1.25 - r$

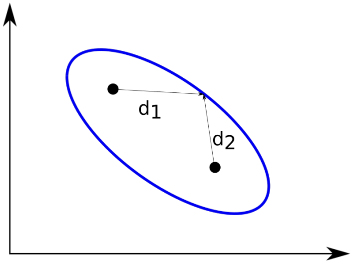
Midpoint circle drawing - algorithm

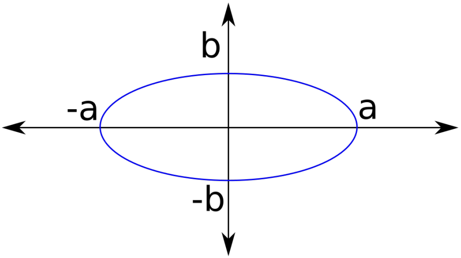
Algorithm 1: Midpoint circle drawing Algorithm

Assume the circle is centered at (0,0) and radius = ***r***
Assign x=r, y=0
Calculate the initial decision parameter P_0 = 1.25 -r
Initialize k to 0
while $x \geq y$ **do**
    SetPixel(x,y)
    y=y+1
    if $P_k < 0$ **then**
        $P_{k+1} = P_k + 2(y_k + 1) + 1$
    else
        x=x-1
        $P_{k+1} = P_k + 2(y_k + 1) - 2(x_k - 1) + 1$
    end
end
Determine symmetry points in the other seven octants

Drawing ellipses

- $d_1 + d_2$ = constant
- $(\frac{x - x_c}{a})^2 + (\frac{y - y_c}{b})^2 = 1$

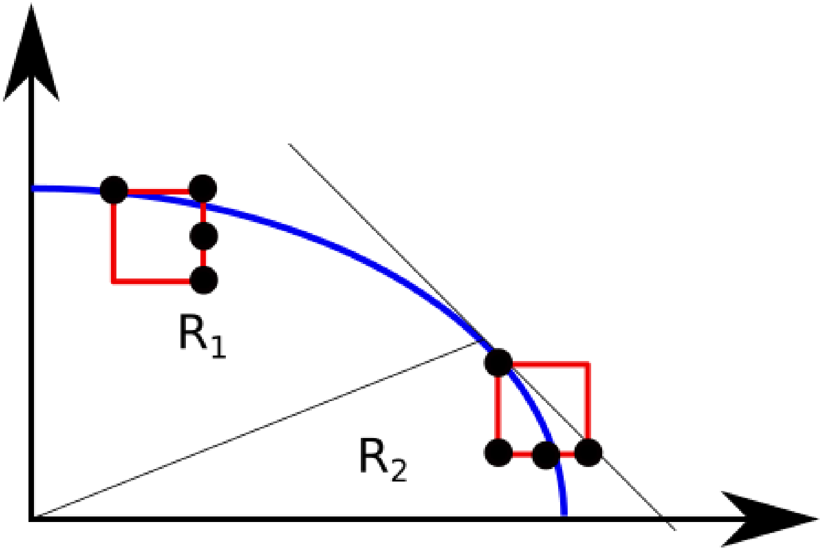




Midpoint ellipse drawing algorithm - Idea

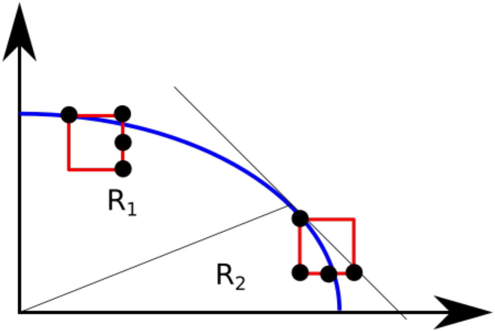
- Assuming (x_c, y_c)=(0,0)
- Equation of ellipse F(x,y)=$b^2x^2 + a^2y^2 - a^2b^2$
- F(x,y)= $\begin{cases} < 0, \text{if (x,y) is inside the ellipse boundary} \\ = 0, \text{if (x,y) is on the ellipse boundary} \\ > 0, \text{if (x,y) is outside the ellipse boundary} \end{cases}$

Midpoint ellipse drawing algorithm - Idea



Midpoint ellipse drawing algorithm - Idea

- We know F(x,y)=$b^2x^2 + a^2y^2 - a^2b^2$
- The switching between regions happens at: $\mid \frac{\delta F}{\delta x} \mid = \mid \frac{\delta F}{\delta y} \mid$
- At region 1, $\mid \frac{\delta F}{\delta x} \mid > \mid \frac{\delta F}{\delta y} \mid$
- At region 2, $\mid \frac{\delta F}{\delta x} \mid < \mid \frac{\delta F}{\delta y} \mid$
- We move out of region 1, if $2b^2x \geq 2a^2y$ gets violated



Midpoint ellipse algorithm - Decision parameter - 1

- In region 1:

$$P1_k = F(x_k + 1, y_k - \frac{1}{2})$$

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

$$P1_{k+1} = F(x_{k+1} + 1, y_{k+1} - \frac{1}{2}) \text{ where, } x_{k+1} = x_k + 1$$

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

$$= P1_k + 2b^2(x_k + 1) + b^2 + a^2[(y_{k+1} - \frac{1}{2})^2 - (y_k - \frac{1}{2})^2]$$

Again, y_{k+1} can be: y_k or y_k-1, *Gives*

$$P1_{k+1} = p1_k + 2b^2x_{k+1} + b^2; \text{ if } p1_k < 0$$

and

$$P1_{k+1} = p1_k + 2b^2x_{k+1} + b^2 - 2a^2y_{k+1}; \text{ if } p1_k > 0$$

Midpoint ellipse algorithm - Decision parameter - 2

- In region 2:

$$P2_k = F(x_k + \frac{1}{2}, y_k - 1)$$
$$= b^2(x_k + \frac{1}{2})^2 + a^2(y_k - 1)^2 - a^2b^2$$
$$P2_{k+1} = F(x_{k+1} + \frac{1}{2}, y_{k+1} - 1) \text{ where, } y_{k+1} = y_k - 1$$
$$= b^2(x_{k+1} + \frac{1}{2})^2 + a^2[(y_k - 1) - 1)]^2 - a^2b^2$$
$$= P2_k - 2a^2(y_k - 1) + a^2 + b^2[(x_{k+1} + \frac{1}{2})^2 - (x_k + \frac{1}{2})^2]$$

Again, x_{k+1} can be: x_k or x_k+1, *Gives*

$$P2_{k+1} = p2_k - 2a^2y_{k+1} + a^2; \text{ if } p2_k > 0$$

and

$$P2_{k+1} = p2_k + 2b^2x_{k+1} + a^2 - 2a^2y_{k+1}; \text{ if } p2_k < 0$$

Midpoint ellipse algorithm - Initial dp

- Region 1 starts at (0,b):

$$
\begin{aligned}
P1_0 &= F(1, b - \frac{1}{2}) \\
&= b^2 + a^2(b - \frac{1}{2})^2 - a^2b^2 \\
&= b^2 - a^2b + \frac{1}{4}a^2
\end{aligned}
$$

- Region 2 starts at (x_0, y_0)

$$
\begin{aligned}
P2_0 &= F(x_0 + \frac{1}{2}, y_0 - 1) \\
&= b^2(x_0 + \frac{1}{2})^2 + a^2(y_0 - 1)^2 - a^2b^2
\end{aligned}
$$

- Putting it all together - Overall algorithm

Next class

- Time: 10th Sep 2020, Thursday : 9 to 10
- Topic: 2D geometry with a focus on polygons
- First set of seminars: Thursday 11 to 12
- Seminar topics:
  - 39 - Modeling plant life in computer graphics
  - 26 - Will be updated soon
  - 46 - High Resolution neural Face Swapping For Visual Effects
  - 80 - Automatically modeling piecewise planar furniture shapes from unorganized point cloud
- Make sure to submit three titles (which you feel comfortable) - one week before the seminar date