Computer Graphics

Scan-Conversion: Ellipse

Md. Biplob Hosen

Assistant Professor, IIT-JU

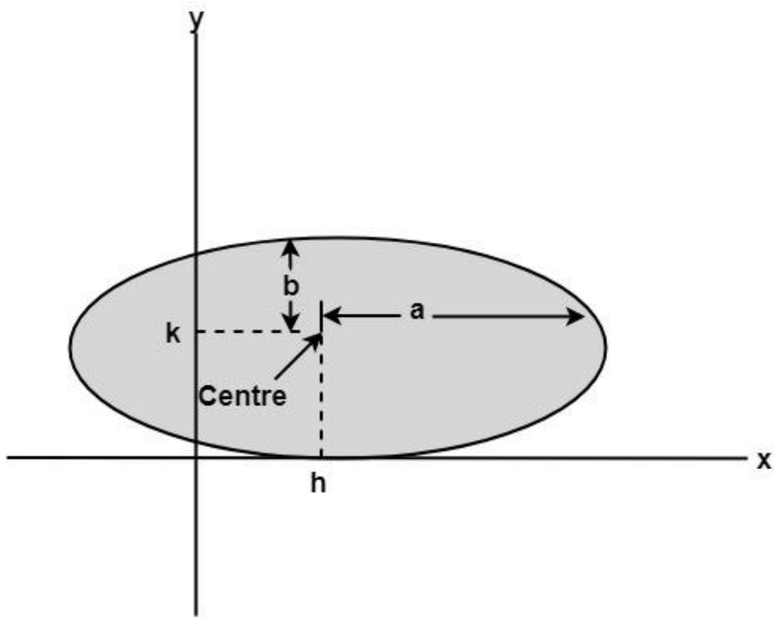
Email: biplob.hosen@juniv.edu

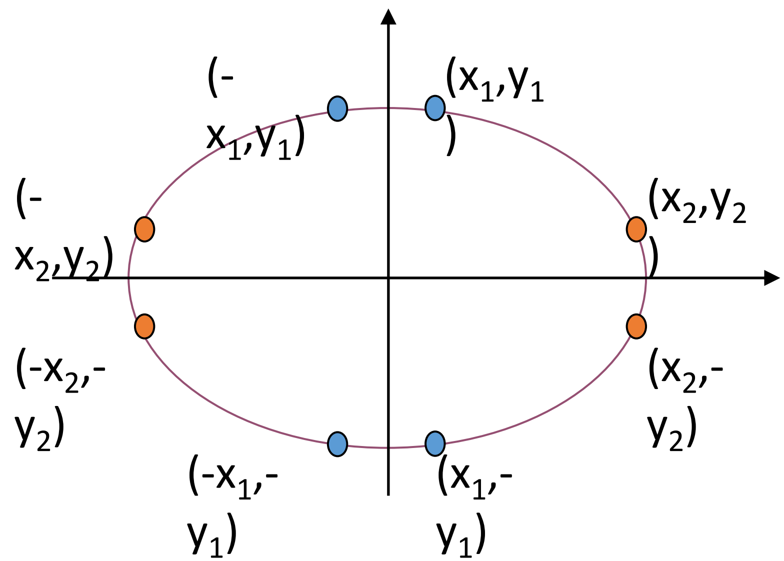
Contents

- Scan-Converting an Ellipse
- Ellipse Drawing Algorithm
  - ✓ Midpoint Algorithm
- Scan-Converting:
  - ✓ Arcs
  - ✓ Sectors
  - ✓ Rectangle

Scan Converting a Ellipse

- The ellipse is also a symmetric figure like a circle but is four-way symmetry rather than eight-way.





- There are two methods of defining an Ellipse:
  - Polynomial Method
  - Trigonometric Method

Polynomial Method

- The ellipse has a major and minor axis. If a and b are major and minor axis respectively. The centre of ellipse is (h, k). The value of x will be incremented from h to a and value of y will be calculated using the following formula:

$$y = b\sqrt{1 - \frac{(x-h)^2}{a^2}} + k$$

Drawback of Polynomial Method:

- It requires squaring of values. So floating point calculation is required.
- Routines developed for such calculations are very complex and slow.

Trignometric Method

- The following equation defines an ellipse trigonometrically:
  - $x = a \cos\theta + h$ and $y = b \sin\theta + k$.

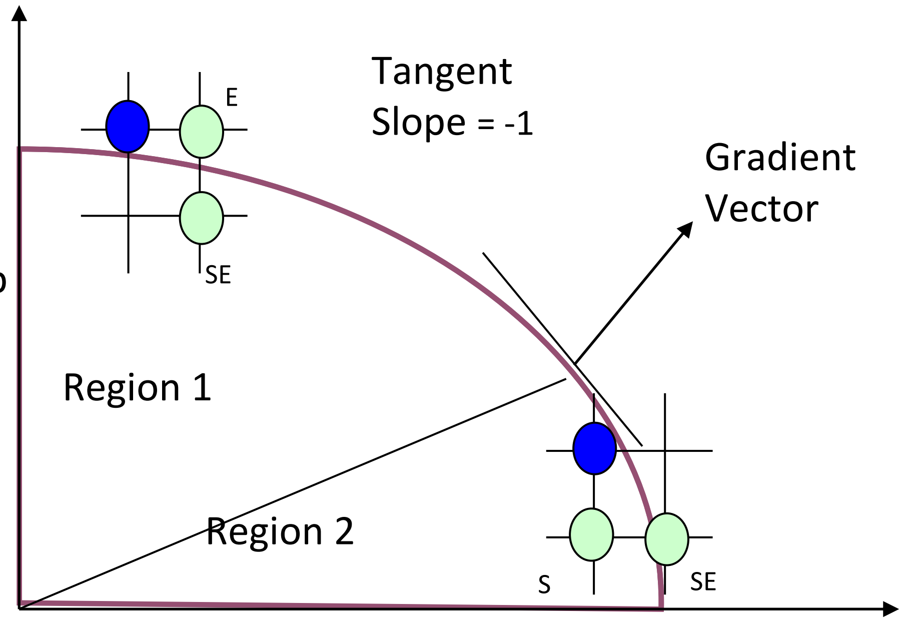
    where (x, y) = the current coordinates.
  - a = length of major axis.
  - b = length of minor axis.
  - θ= current angle.
  - (h, k) = ellipse center.
- In this method, the value of θ is varied from 0 to $\pi/2$ radians. The remaining points are found by symmetry.

Drawback:

- This is an inefficient method.
- It is not an interactive method for generating ellipse.
- The table is required to see the trigonometric value.
- Memory is required to store the value of θ.

Midpoint Algorithm

- Implicit equation is: $F(x,y) = b^2x^2 + a^2y^2 - a^2b^2 = 0$
- We have only 4-way symmetry
- There exists two regions:
  - In **Region 1** dx > dy
    - Increase x at each step
    - y may decrease
  - In **Region 2** dx < dy
    - Decrease y at each step
    - x may increase



Pseudo Code

```
x=0, y=b;
fx=0, fy=a²b
p=b²-a²b+1/4(a²)
while (fx<fy) {
  setPixel (x, y);
  x++;
  fx=fx+b²;
  if(p<0)
        p=p+fx+b²;
  else{
        y--;
        fy=fy-a²;
        p=p+fx-fy+b²;
  }
}
setPixel(x,y);
```

```
p=b²(x+0.5)²+a²(y-1)²-a²b²
while (y>0) {
  y--;
  fy=fy-a²;
  if(p>=0)
        p=p-fy+a²;
  else{
        x++;
        fx=fx+b²;
        p=p+fx-fy+a²;
  }
setPixel(x,y);
}
```

Example

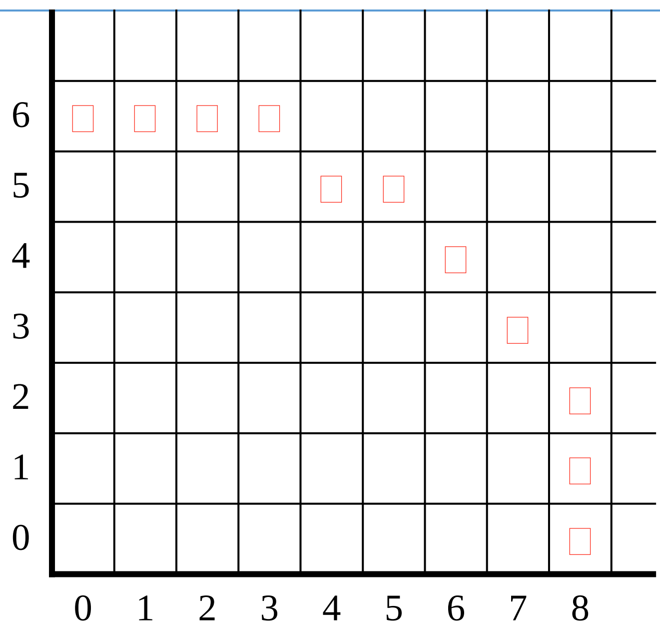
- Draw an ellipse with *a*= 8, *b* = 6 using Midpoint Ellipse Algorithm.
- **Solution:**
- **Region 1: initial (0,6)**

i	X_i	y_i	p_i	p_{i+1}	x_{i+1}	y_{i+1}	F_{x+1}	F_{y+1}
0	0	6	-332	-260	1	6	36	768
1	1	6	-260	-152	2	6	72	384
2	2	6	-152	-8	3	6	108	384
3	3	6	-8	172	4	6	144	384
4	4	6	172	68	5	5	180	320
5	5	5	68	64	6	4	216	256
6	6	4	64	160	7	3	252	192
7	7	3						

Continue…

- **Region 2: initial (7,3)**

i	x_i	y_i	p_i	p_{i+1}	x_{i+1}	y_{i+1}	f_x	f_y
8	7	3	--23	201	8	2	288	128
9	8	2	201	201	8	1	288	64
10	8	1	201	265	8	0	288	0
11	8	0						

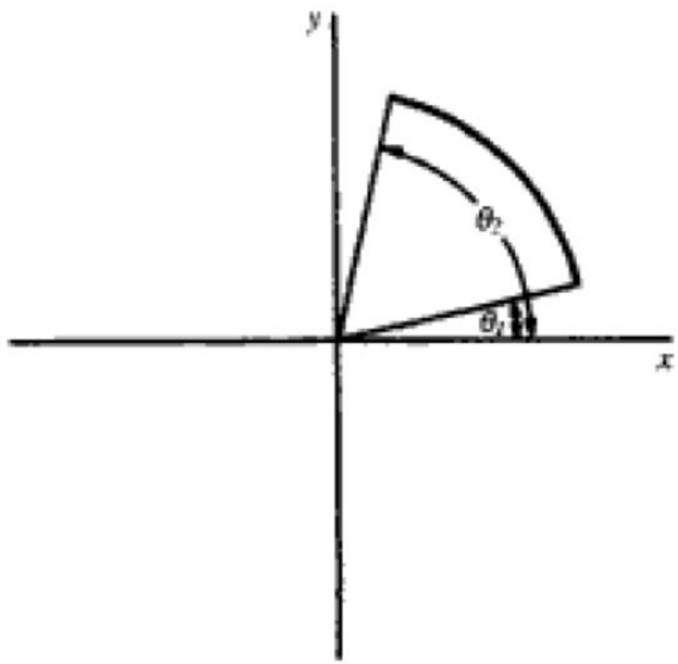


Exercise

- Generate all points of an ellipse with $a = 6$, $b = 8$ using Midpoint Ellipse Algorithm.
- Generate all points of an ellipse with $a = 14$, $b = 10$ using Midpoint Ellipse Algorithm center at (15, 10).

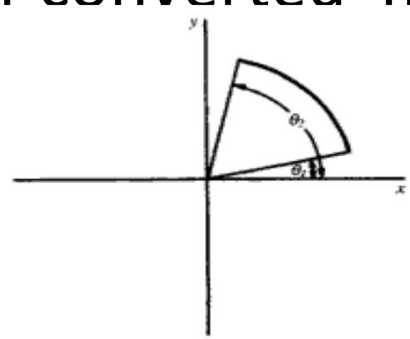
Scan-Converting ARCS

- An Arc can be generated using either the polynomial or the trigonometric method.
- When the trigonometric method is used, the starting value is set to θ_1 and the ending value is set to θ_2.
- The rest of the steps are similar to those used when scan-converting a circle, except that symmetry is not used.



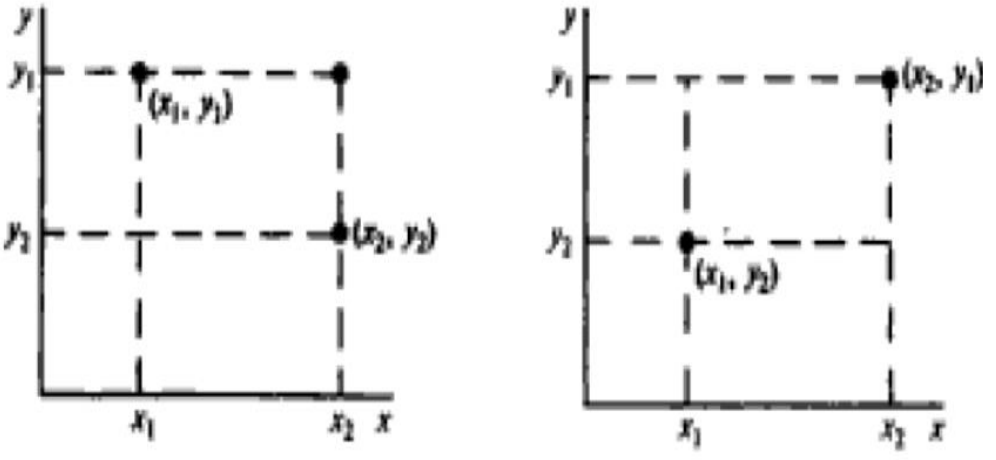
Scan-Converting Sectors

- A sector is scan-converted by using any of the methods of scan-converting an arc, and then scan-converting two lines from the center of the arc to the endpoints of the arc.
- For example, assume that a sector whose center is at point (h,k) is to be scan-converted.
- First scan-convert an arc from θ_1 to θ_2.
- Next, a line to be scan-converted from (h, k) to ($r\cos\theta_1+h$, $r\sin\theta_1+k$).
- A second line to be scan-converted from (h, k) to ($r\cos\theta_2+h$, $r\sin\theta_2+k$).



Scan-Converting a Rectangle

- A rectangle whose sides are parallel to the co-ordinates axes may be constructed if the locations of two vertices are known. The remaining corner points are then derived.
- Once the vertices are known, the four sets of co-ordinates are sent to line routine and the rectangle is scan-converted.
- In the case of the rectangle shown, lines would be drawn as follows:
  - Line: $(x_{1,} y_1)$ to $(x_{1,} y_2)$
  - Line: $(x_{1,} y_2)$ to $(x_{2,} y_2)$
  - Line: $(x_{2,} y_2)$ to $(x_{2,} y_1)$
  - Line: $(x_{2,} y_1)$ to $(x_{1,} y_1)$



Thank You!