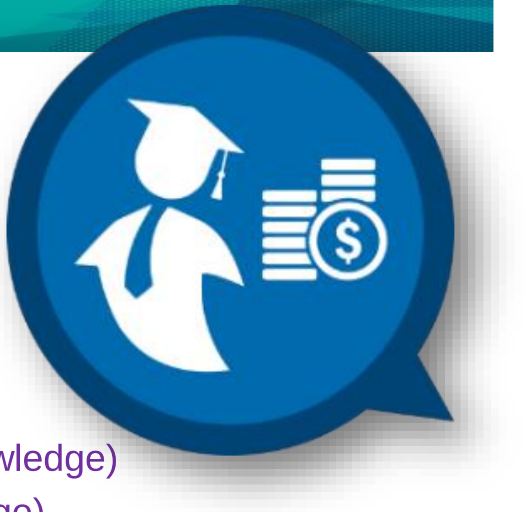
Computer Graphics

Introduction to Computer Graphics

Edited by
Dr. Md. Manjur Ahmed
Dr. Suryanti Awang
Faculty of Computer Systems and Software Engineering
manjur@ump.edu.my

Chapter Description



- Aims
  - Basic of Computer Graphics.

- Expected Outcomes
  - Understand the basic concept of computer graphics. (CO1: Knowledge)
  - Ability to use the computer graphics technology. (CO1: Knowledge)

- References

  - Computer Graphics by Zhigang Xiang, Schaum’s Outlines.
  - Donald Hearn & M. Pauline Baker, Computer Graphics with OpenGL, 4th Edition, Boston : Addison Wesley, 2011.

Circle Drawing

Simple way to start

Equation of a circle:

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

If we solve for *y*:

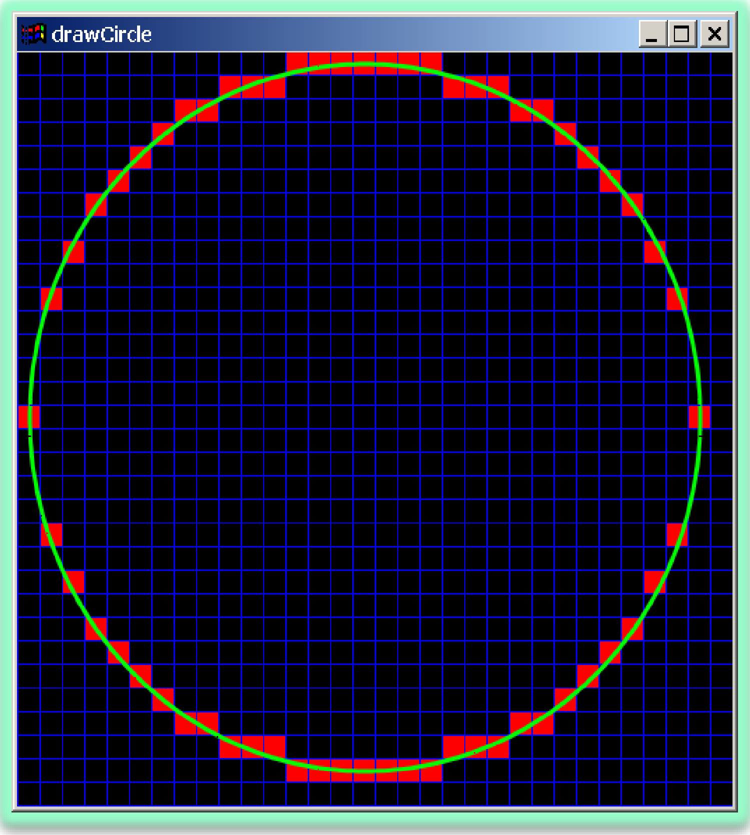
$$y = y_0 \pm \sqrt{r^2 - (x - x_0)^2}$$

Sample C codes:

```
void SIMPLE_CIRCLE(int X_center, int Y_center, int radius_R, Color c) {
  int x, y, r2;
  r2 = radius_R * radius_R;
  for (x = - radius_R; x <= radius_R; x++) {
    y = (int)(sqrt(r2 - x*x) + 0.5);
    setPixel((X_center + x), (Y_center + y, c));
    setPixel((X_center + x), (Y_center - y, c));
  }
}
```

Simple way to start "uncertainty"

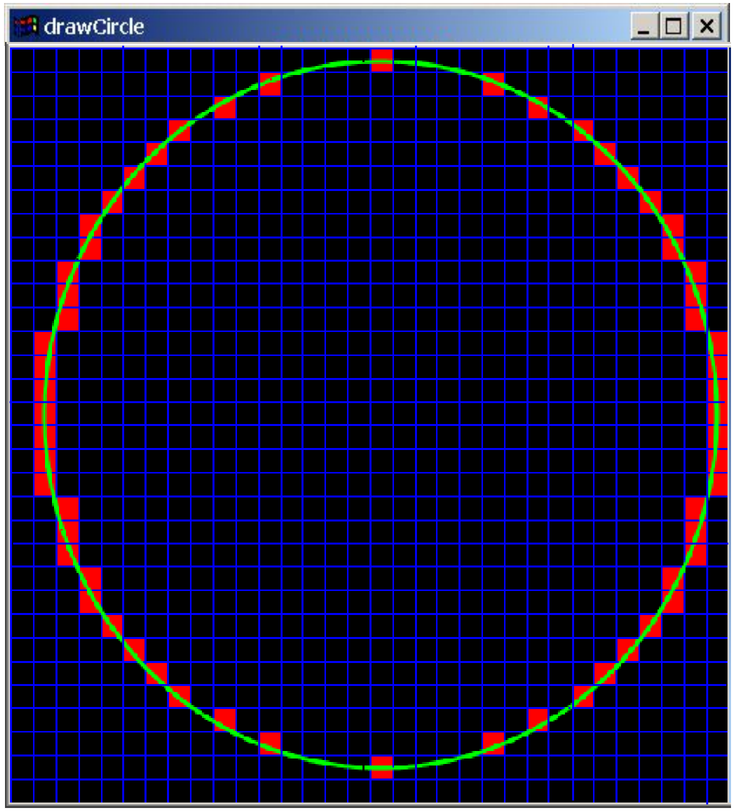
- In certain cases, the slope of the line tangent is greater than 1. Therefore, tangent > 1 shows the uncertainty of this algorithm.
- Looping (or stepping) using *x* won't work here.



Reference: Computer Graphics: Principles and Practice by James D. Foley and et.al.

Circles symmetrical nature

- To solve the issue "slope of the tangent"
- Let's take the advantage of the circle's symmetric nature.
- Consider both positive and negative values for *y* (or x).

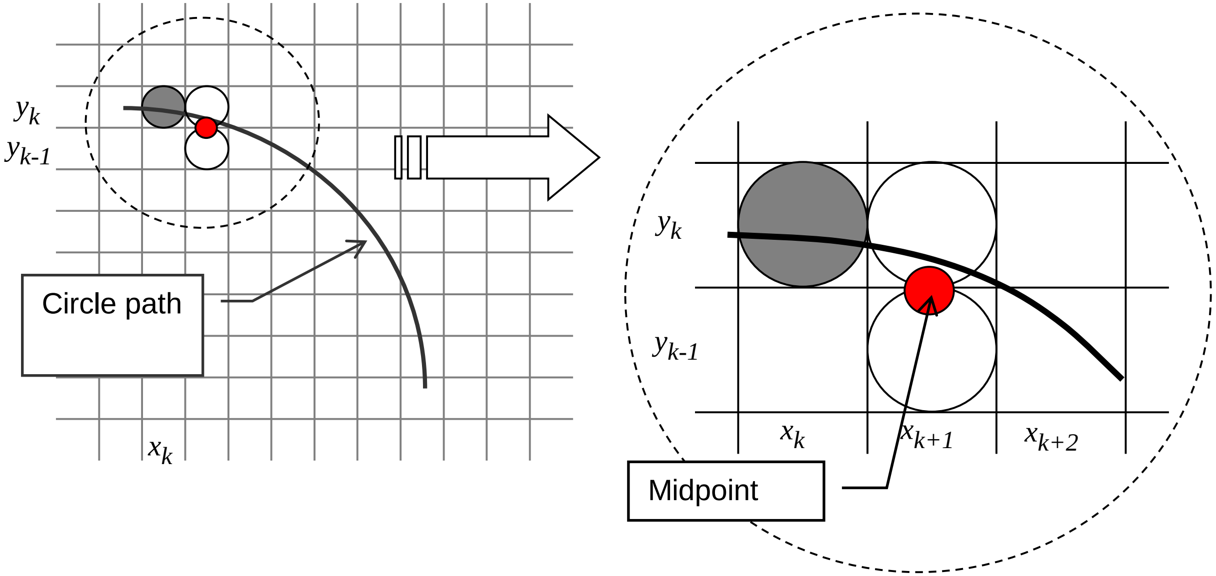


Midpoint Circle Algorithm

- Consider current point is at (x_k, y_k)
- Next point: (x_k+1, y_k), or (x_k+1, y_k-1)?
- Take the midpoint: (x_k+1, y_k-0.5)

•Use the discriminator function to decide:

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



Using the circle discriminator

Based on the value return:

$f(x, y)$
- < 0 inside the circle
- $= 0$; on the circle path
- >0; outside the circle

By using the midpoint between 2 pixel candidates, we can introduce a decision parameter, p_k , to decide which to plot next:

1)

$$p_k = f(x_k + 1, y_k - 0.5)$$
$$= (x_k + 1)^2 + (y_k - 0.5)^2 - r^2$$

2 and 3)

p_k
- -ve: midpoint is inside the circle; plot (x_k+1, y_k)
- +ve: midpoint is outside the circle; plot (x_k+1, y_k-1)

If the current point is inside the circle ...

If $p_k < 0$

We want to know f(x+1, y) so we can update p:

$$f(x+1, y) = (x + 1)^2 + y^2 - r^2$$

$$f(x+1, y) = (x^2 + 2x + 1) + y^2 - r^2$$

$$f(x+1, y) = f(x, y) + 2x + 1$$

P_{k+1} P_k

So we increment:

$$p \mathrel{+}= 2x + 1$$

If the current point is outside the circle …

If $p_k > 0$

Let's drop the subscript for a while…

We want to know f(x+1, y-1) so we can update p:

$$f(x+1, y-1) = (x + 1)^2 + (y - 1)^2 - r^2$$

$$f(x+1, y-1) = (x^2 + 2x + 1) + (y^2 - 2y + 2) - r^2$$

$$f(x+1, y-1) = f(x, y) + 2x - 2y + 2$$

P_{k+1} P_k

And we increment:

$$p \mathrel{+}= 2x - 2y + 2$$

Where to begin?

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

We can determine where to go next, how do we start?

We have a variety of choices for our first point on the circle, but we've designed the increments to work from (0, r).

Calculate initial value of p_0 by evaluating:

$$p_0 = f(1, r–0.5) = 1^2 + (r – 0.5)^2 – r^2$$

$$p_0 = f(1, r–0.5) = 1 + (r^2 – r + 0.25) – r^2$$

$$p_0 = 1.25 – r$$

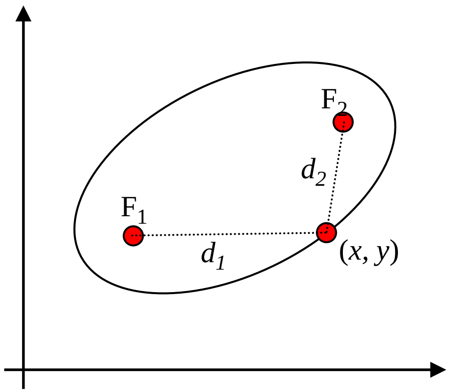
****We want to use integer calculation; you can round p_0**

Midpoint circle algorithm

Homework

Ellipse Drawing

Equation

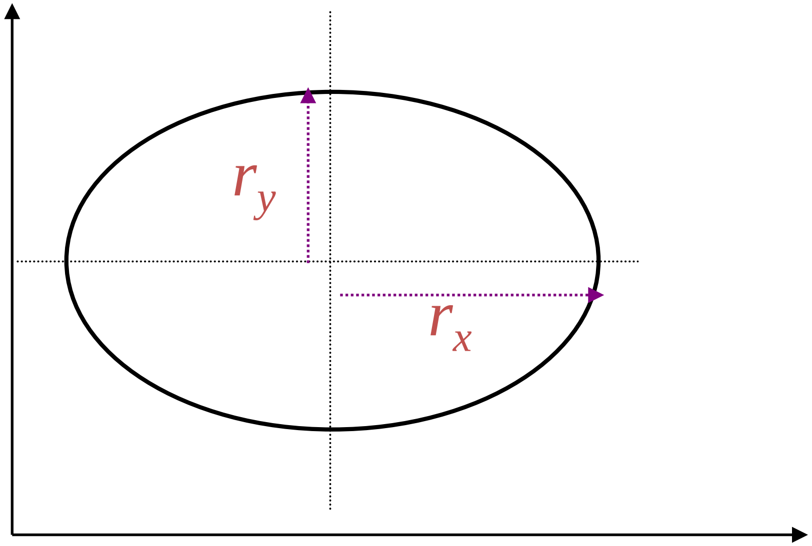


General equation of an ellipse:

$$\sqrt{(x-x_1)^2+(y-y_1)^2}+\sqrt{(x-x_2)^2+(y-y_2)^2}=\text{constant}$$

$$d_1 + d_2 = constant$$

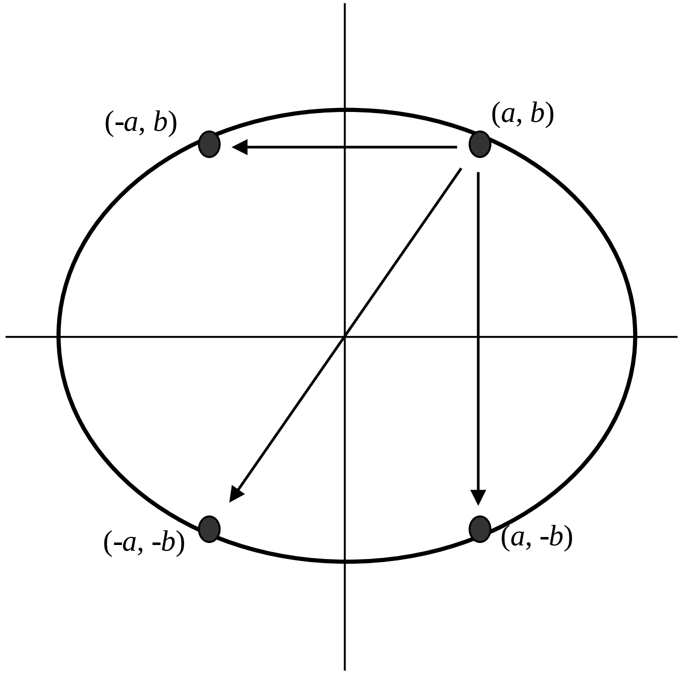
Or,



$$\left(\frac{x-x_c}{r_x}\right)^2+\left(\frac{y-y_c}{r_y}\right)^2=1$$

Symmetry

An ellipse only has a 2-way symmetry.



Equation of an ellipse revisited

Consider an ellipse centered at the origin:

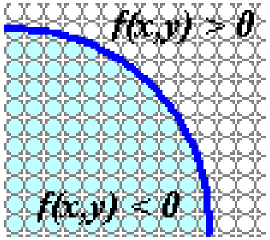
$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 = 1$$

What is the discriminator function?

$$f_e(x, y) = r_y{}^2 x^2 + r_x{}^2 y^2 - r_x{}^2 r_y{}^2$$

…and its properties:

$f_e(x,y) < 0$ for a point inside the ellipse
$f_e(x,y) > 0$ for a point outside the ellipse
$f_e(x,y) = 0$ for a point on the ellipse



Midpoint Ellipse Algorithm

- Ellipse is different from circle.
- Similar approach with circle, different is sampling direction.
- Region 1:
  - Sampling is at *x* direction
  - Choose between (x_k+1, y_k), or (x_k+1, y_k-1)
  - Midpoint: $(x_k+1, y_k-0.5)$
- Region 2:
  - Sampling is at *y* direction
  - Choose between (x_k, y_k-1), or (x_k+1, y_k-1)
  - Midpoint: $(x_k+0.5, y_k-1)$