

Higher Technological Institute
Computer Science Department



Computer Graphics

Dr Osama Farouk
Dr Ayman Soliman
Dr Adel Khaled

Lecture Four

Graphics Output Primitives

ELLIPSE-GENERATING ALGORITHMS

Properties of Ellipses

A precise definition of an ellipse can be given in terms of the distances from any point on the ellipse to two fixed positions, called the foci of the ellipse.

$$d_1 + d_2 = \text{constant} \quad (1)$$

Expressing distances d_1 and d_2

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

The general ellipse equation in the form

$$Ax^2 + By^2 + Cxy + Dx + Ey + F = 0$$

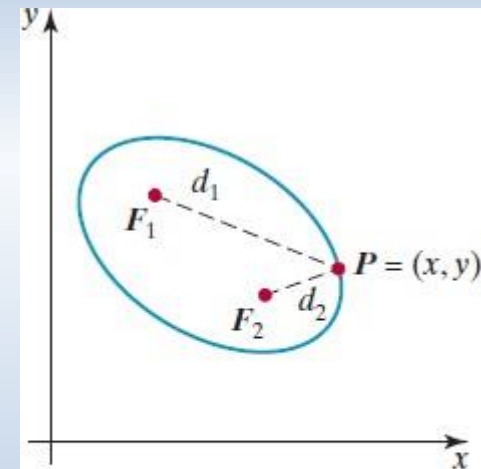


FIGURE Ellipse generated about foci F_1 and F_2 .

The major axis is the straight-line segment extending from one side of the ellipse to the other through the foci. The minor axis spans the shorter dimension of the ellipse.

The equation for the ellipse

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

Using polar coordinates r and θ

$$\begin{aligned} x &= x_c + r_x \cos \theta \\ y &= y_c + r_y \sin \theta \end{aligned} \quad (2)$$

If $r_x > r_y$, the radius of the bounding circle is $r = r_x$. Otherwise, the bounding circle has radius $r = r_y$.

As with the circle algorithm, symmetry considerations can be used to reduce computations. An ellipse in standard position is symmetric between quadrants, but, unlike a circle, it is not symmetric between the two octants of a quadrant. Thus, we must calculate pixel positions along the elliptical arc throughout one quadrant, then use symmetry to obtain curve positions in the remaining three quadrants

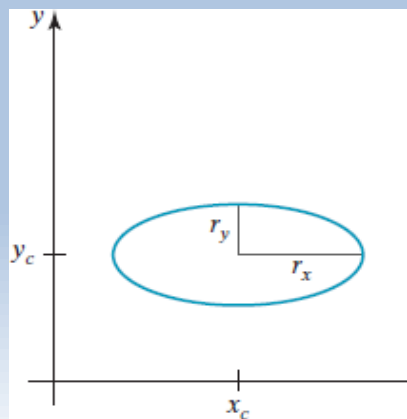


FIGURE Ellipse centered at (x_c, y_c) with semimajor axis r_x and semiminor axis r_y .

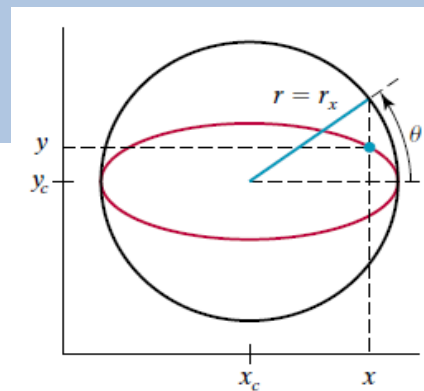
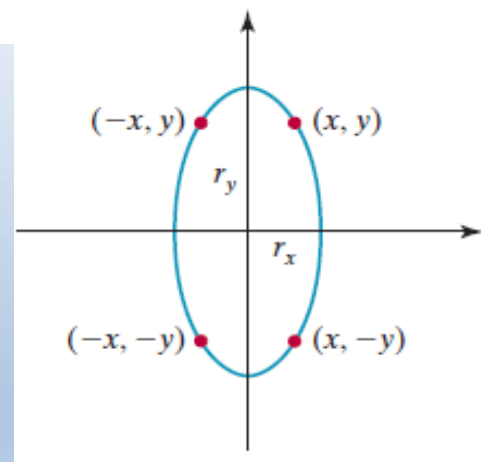


FIGURE The bounding circle and eccentric angle θ for an ellipse with $r_x > r_y$.



Midpoint Ellipse Algorithm

Given parameters r_x , r_y , and (x_c, y_c) , we determine curve positions (x, y) for an ellipse in standard position centered on the origin, then we shift all the points using a fixed offset so that the ellipse is centered at (x_c, y_c) . If we wish also to display the ellipse in nonstandard position, we could rotate the ellipse about its center coordinates to reorient the major and minor axes in the desired directions.

The midpoint ellipse method is applied throughout the first quadrant in two parts.

Regions 1 and 2 (Figure) can be processed in various ways. We can start at position $(0, r_y)$ and step clockwise along the elliptical path in the first quadrant, shifting from unit steps in x to unit steps in y when the slope becomes less than -1.0 . Alternatively, we could start at $(r_x, 0)$ and select points in a counterclockwise order, shifting from unit steps in y to unit steps in x when the slope becomes greater than -1.0 . With parallel processors, we could calculate pixel positions in the two regions simultaneously. As an example of a sequential implementation of the midpoint algorithm, we take the start position at $(0, r_y)$ and step along the ellipse path in clockwise order throughout the first quadrant.

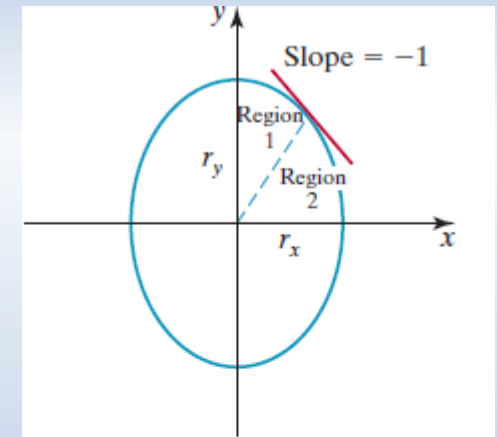


FIGURE Ellipse processing regions. Over region 1, the magnitude of the ellipse slope is less than 1.0; over region 2, the magnitude of the slope is greater than 1.0.

We define an **ellipse function** from Equation (2) with $(x_c , y_c) = (0, 0)$ as

$$f_{\text{ellipse}}(x, y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2 \quad (3)$$

which has the following properties:

$$f_{\text{ellipse}}(x, y) \begin{cases} < 0, & \text{if } (x, y) \text{ is inside the ellipse boundary} \\ = 0, & \text{if } (x, y) \text{ is on the ellipse boundary} \\ > 0, & \text{if } (x, y) \text{ is outside the ellipse boundary} \end{cases} \quad (4)$$

Starting at $(0, r_y)$, we take unit steps in the x direction until we reach the boundary between region 1 and region 2. Then we switch to unit steps in the y direction over the remainder of the curve in the first quadrant. At each step we need to test the value of the slope of the curve. The ellipse slope is calculated from Equation (3) as

$$\frac{dy}{dx} = -\frac{2r_y^2 x}{2r_x^2 y} \quad (5)$$

At the boundary between region -1 and region 2 $dy/dx = -1.0$ and

$$2r_y^2 x = 2r_x^2 y \quad (6)$$

Therefore, we move out of region 1 whenever

$$2r_y^2 x \geq 2r_x^2 y$$

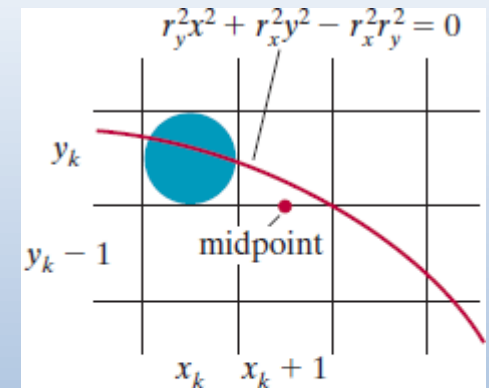


FIGURE Midpoint between candidate pixels at sampling position $x_k + 1$ along an elliptical path.

The decision parameter (that is, the ellipse function 3) at this midpoint:

$$\begin{aligned}
 p1_k &= f_{\text{ellipse}}\left(x_k + 1, y_k - \frac{1}{2}\right) \\
 &= r_y^2(x_k + 1)^2 + r_x^2\left(y_k - \frac{1}{2}\right)^2 - r_x^2 r_y^2
 \end{aligned}
 \tag{7}$$

If $p1_k < 0$, the midpoint is inside the ellipse and the pixel on scan line y_k is closer to the ellipse boundary. Otherwise, the midposition is outside or on the ellipse boundary, and we select the pixel on scan line $y_k - 1$.

At the next sampling position ($x_{k+1} + 1 = x_k + 2$), the decision parameter for region 1 is evaluated as

$$\begin{aligned}
 p1_{k+1} &= f_{\text{ellipse}}\left(x_{k+1} + 1, y_{k+1} - \frac{1}{2}\right) \\
 &= r_y^2[(x_k + 1) + 1]^2 + r_x^2\left(y_{k+1} - \frac{1}{2}\right)^2 - r_x^2 r_y^2
 \end{aligned}
 \tag{8}$$

or

$$p1_{k+1} = p1_k + 2r_y^2(x_k + 1) + r_y^2 + r_x^2 \left[\left(y_{k+1} - \frac{1}{2}\right)^2 - \left(y_k - \frac{1}{2}\right)^2 \right]$$

where y_{k+1} is either y_k or $y_k - 1$, depending on the sign of $p1_k$.

Decision parameters are incremented by the following amounts:

$$\text{increment} = \begin{cases} 2r_y^2 x_{k+1} + r_y^2, & \text{if } p1_k < 0 \\ 2r_y^2 x_{k+1} + r_y^2 - 2r_x^2 y_{k+1}, & \text{if } p1_k \geq 0 \end{cases} \quad (9)$$

At the initial position $(0, r_y)$, these two terms evaluate to

$$\begin{aligned} 2r_y^2 x &= 0 \\ 2r_x^2 y &= 2r_x^2 r_y \end{aligned} \quad (10)$$

In region 1, the initial value of the decision parameter is obtained by evaluating the ellipse function at the start position $(x_0, y_0) = (0, r_y)$:

$$\begin{aligned} p1_0 &= f_{\text{ellipse}}\left(1, r_y - \frac{1}{2}\right) \\ &= r_y^2 + r_x^2 \left(r_y - \frac{1}{2}\right)^2 - r_x^2 r_y^2 \end{aligned} \quad (11)$$

or

$$p1_0 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2$$

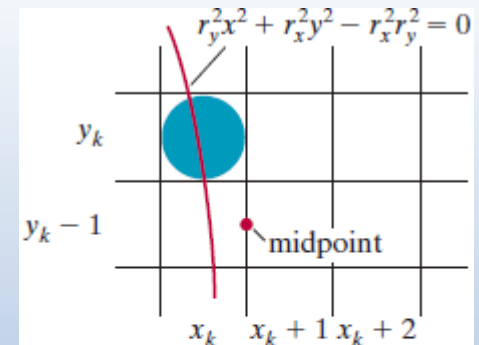


FIGURE Midpoint between candidate pixels at sampling position $y_k - 1$ along an elliptical path.

Over region 2, we sample at unit intervals in the negative y direction, and the midpoint is now taken between horizontal pixels at each step (Figure). For this region, the decision parameter is evaluated as

$$p2_k = f_{\text{ellipse}} \left(x_k + \frac{1}{2}, y_k - 1 \right)$$

$$= r_y^2 \left(x_k + \frac{1}{2} \right)^2 + r_x^2 (y_k - 1)^2 - r_x^2 r_y^2$$

$$p2_{k+1} = f_{\text{ellipse}} \left(x_{k+1} + \frac{1}{2}, y_{k+1} - 1 \right)$$

$$= r_y^2 \left(x_{k+1} + \frac{1}{2} \right)^2 + r_x^2 [(y_k - 1) - 1]^2 - r_x^2 r_y^2$$

or

$$p2_{k+1} = p2_k - 2r_x^2 (y_k - 1) + r_x^2 + r_y^2 \left[\left(x_{k+1} + \frac{1}{2} \right)^2 - \left(x_k + \frac{1}{2} \right)^2 \right]$$

$$p2_0 = f_{\text{ellipse}} \left(x_0 + \frac{1}{2}, y_0 - 1 \right)$$

$$= r_y^2 \left(x_0 + \frac{1}{2} \right)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$$

Midpoint Ellipse Algorithm

1. Input r_x, r_y , and ellipse center (x_c, y_c) , and obtain the first point on an ellipse centered on the origin as

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

2. Calculate the initial value of the decision parameter in region 1 as

$$p1_0 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2$$

3. At each x_k position in region 1, starting at $k = 0$, perform the following test: If $p1_k < 0$, the next point along the ellipse centered on $(0, 0)$ is (x_{k+1}, y_k) and

$$p1_{k+1} = p1_k + 2r_y^2 x_{k+1} + r_y^2$$

Otherwise, the next point along the ellipse is $(x_k + 1, y_k - 1)$ and

$$p1_{k+1} = p1_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_y^2$$

with

$$2r_y^2 x_{k+1} = 2r_y^2 x_k + 2r_y^2, \quad 2r_x^2 y_{k+1} = 2r_x^2 y_k - 2r_x^2$$

and continue until $2r_y^2 x \geq 2r_x^2 y$.

4. Calculate the initial value of the decision parameter in region 2 as

$$p2_0 = r_y^2 \left(x_0 + \frac{1}{2} \right)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$$

where (x_0, y_0) is the last position calculated in region 1.

5. At each y_k position in region 2, starting at $k = 0$, perform the following test: If $p2_k > 0$, the next point along the ellipse centered on $(0, 0)$ is $(x_k, y_k - 1)$ and

$$p2_{k+1} = p2_k - 2r_x^2 y_{k+1} + r_x^2$$

Otherwise, the next point along the ellipse is $(x_k + 1, y_k - 1)$ and

$$p2_{k+1} = p2_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_x^2$$

using the same incremental calculations for x and y as in region 1. Continue until $y = 0$.

6. For both regions, determine symmetry points in the other three quadrants.
7. Move each calculated pixel position (x, y) onto the elliptical path centered on (x_c, y_c) and plot these coordinate values:

$$x = x + x_c, \quad y = y + y_c$$

Example: Midpoint Ellipse Drawing (Textbook P114-115)

Given input ellipse parameters $r_x = 8$ and $r_y = 6$, we illustrate the steps in the midpoint ellipse algorithm by determining raster positions along the ellipse path in the first quadrant. Initial values and increments for the decision parameter calculations are

$$2r_y^2x = 0 \quad (\text{with increment } 2r_y^2 = 72)$$

$$2r_x^2y = 2r_x^2r_y \quad (\text{with increment } -2r_x^2 = -128)$$

For region 1, the initial point for the ellipse centered on the origin is $(x_0, y_0) = (0, 6)$, and the initial decision parameter value is

$$p1_0 = r_y^2 - r_x^2r_y + \frac{1}{4}r_x^2 = -332$$

Successive midpoint decision-parameter values and the pixel positions along the ellipse are listed in the following table:

k	$p1_k$	(x_{k+1}, y_{k+1})	$2r_y^2x_{k+1}$	$2r_x^2y_{k+1}$
0	-332	(1, 6)	72	768
1	-224	(2, 6)	144	768
2	-44	(3, 6)	216	768
3	208	(4, 5)	288	640
4	-108	(5, 5)	360	640
5	288	(6, 4)	432	512
6	244	(7, 3)	504	384

We now move out of region 1 because $2r_y^2x > 2r_x^2y$.

For region 2, the initial point is $(x_0, y_0) = (7, 3)$ and the initial decision parameter is

$$p_{2_0} = f_{\text{ellipse}}\left(7 + \frac{1}{2}, 2\right) = -151$$

The remaining positions along the ellipse path in the first quadrant are then calculated as

k	P_{2_k}	(x_{k+1}, y_{k+1})	$2r_y^2x_{k+1}$	$2r_x^2y_{k+1}$
0	-151	(8, 2)	576	256
1	233	(8, 1)	576	128
2	745	(8, 0)	—	—

A plot of the calculated positions for the ellipse within the first quadrant is shown in Figure 23.

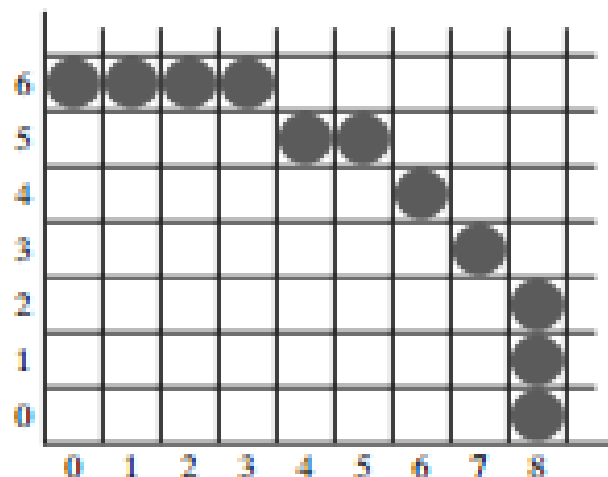


FIGURE 23

Pixel positions along an elliptical path centered on the origin with $r_x = 8$ and $r_y = 6$, using the midpoint algorithm to calculate locations within the first quadrant.

```

inline int round (const float a) { return int (a + 0.5); }

/* The following procedure accepts values for an ellipse
 * center position and its semimajor and semiminor axes, then
 * calculates ellipse positions using the midpoint algorithm.
 */
void ellipseMidpoint (int xCenter, int yCenter, int Rx, int Ry)
{
    int Rx2 = Rx * Rx;
    int Ry2 = Ry * Ry;
    int twoRx2 = 2 * Rx2;
    int twoRy2 = 2 * Ry2;
    int p;
    int x = 0;
    int y = Ry;
    int px = 0;
    int py = twoRx2 * y;
    void ellipsePlotPoints (int, int, int, int);

    /* Plot the initial point in each quadrant. */
    ellipsePlotPoints (xCenter, yCenter, x, y);

    /* Region 1 */
    p = round (Ry2 - (Rx2 * Ry) + (0.25 * Rx2));
    while (px < py) {
        x++;
        px += twoRy2;
        if (p < 0)
            p += Ry2 + px;
        else {
            y--;
            py -= twoRx2;
            p += Ry2 + px - py;
        }
        ellipsePlotPoints (xCenter, yCenter, x, y);
    }

    /* Region 2 */
    p = round (Ry2 * (x+0.5) * (x+0.5) + Rx2 * (y-1) * (y-1) - Rx2 * Ry2);
    while (y > 0) {
        y--;
        py -= twoRx2;
        if (p > 0)
            p += Rx2 - py;
        else {
            x++;
            px += twoRy2;
            p += Rx2 - py + px;
        }
        ellipsePlotPoints (xCenter, yCenter, x, y);
    }
}

void ellipsePlotPoints (int xCenter, int yCenter, int x, int y):
{
    setPixel (xCenter + x, yCenter + y);
    setPixel (xCenter - x, yCenter + y);
    setPixel (xCenter + x, yCenter - y);
    setPixel (xCenter - x, yCenter - y);
}

```



QUIZ

Given input ellipse parameters $r_x = 12$ and $r_y = 9$, we illustrate the steps in the midpoint ellipse algorithm by determining raster positions along the ellipse path in the first quadrant. Initial values and increments for the decision parameter calculations are

End of Lecture Good Luck!

See you
in next lecture...

