From Ellipses to Fillings

CS116A
Chris Pollett
Sep8, 2004.

Introduction

- Ellipse Generating Algorithms
- Conics and Splines
- Parallel Curve Algorithms
- Pixel Addressing
- Fill-Area Primitives

Ellipse Generating Algorithms

What is an ellipse? Given two foci: $F_1=(x_1,y_1)$ and $F_2=(x_2,y_2)$ in the plane and a poin,t (x,y), let d_i be the distance from F_i to (x,y) for $i=1,2$. The ellipse based on these foci is the set of points such that:

$$d_1+d_2 = \text{constant}$$

Using equations for distance this gives:

$$[(x-x_1)^2+(y-y_1)^2]^{1/2}+[[(x-x_2)^2+(y-y_2)^2]^{1/2} = \text{constant}$$

More ellipses

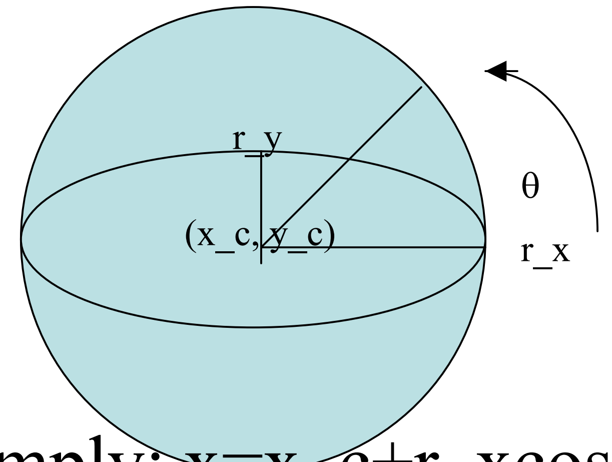
- So can specify an ellipse by giving F_1,F_2 and one point on the boundary. With these we could determine what the constant was and then start drawing the ellipse by varying the x value and seeing what y evaluates to.
- This would be slow.
- If the ellipse is oriented on the x-y axes can do midpoint-like algorithm.

Still more ellipse

When oriented on the x-y axis the equation of an ellipse can be written as:

$[(x-x_c)/r_x]^2 + [(y-y_c)/r_y]^2 = 1$

Where:



Or more simply: $x = x_c + r_x \cos\theta$, $y = y_c + r_y \sin\theta$

Midpoint Ellipse Algorithm

Suppose $(x_c, y_c) = (0, 0)$. Start at $(0, r_y)$, move clockwise. Let $f_{ellipse} = r_y^2 x^2 + r_x y^2 - r_x^2 r_y^2$

Then $f_{ellipse} > 0$ if (x,y) is outside the ellipse, <0 if inside and $=0$ if on the ellipse.

So can use this as our decision function.

More midpoint ellipse algorithm

Have to be careful about when the slope has magnitude less than/greater than 1. Break the first quadrant into regions. One where the magnitude of slope less than 1, in which case step x by 1; the other region where the magnitude is >1 in which case step y.

Using derivatives can determine boundary between two regions given by when:

$2r\_y^2x = 2r\_x^2y$.

Still more midpoint algorithm

In region 1:

$p1_k = f_{ellipse}(x_k+1, y_k-1/2)$.

In region 2:

$p2_k = f_{ellipse}(x_k+1/2, y_k-1)$.

Again, these can be incrementally computed in terms of previous k values starting at 0.

Conics and Splines

Note midpoint idea can be extended to general curves of the form $f(x,y)=0$. Such gives are said to be implicitly given. This is as opposed to curves with an explicit representation: $y=f(x)$. Note: if have latter can always convert to former.

As an example consider a conic section given by:

$Ax^2+by^2+Cxy+Dx+Ey+F=0$.

More Conic Sections and Splines

Discriminant determines type of section:

$B^2-4AC < 0 \Rightarrow$ ellipse, $>0 \Rightarrow$ hyperbola, $=0 \Rightarrow$ parabola.

These curves useful when doing graphics for physical simulations: planetary motion, objects falling, charged particle systems.

Polynomials and Splines

Polynomials are given by equations of the form:

$y=a_0+a_1x+ \ldots a_nx^n$

(Above curve said to be degree n)

Splines are given by equations like:

$x=a_0+a_1u+ \ldots a_nu^n$

$y=b_0+b_1u+ \ldots b_nu^n$

At this point I discussed LaGrange Interpolation on the board

Parallel Curve Algorithms

As with Bresenham can modify the circle and ellipse midpoint algorithms to work in parallel. Need to calculate the starting (x_k, y_k) and p_k values for each processor. Can show this can be done with minimal overhead.

Pixel Addressing

How does a point (x,y) correspond to a pixel on the screen.

In particular, each pixel has some width and height.

If use grid-coordinates then the point (x,y) corresponds to the screen rectangle given between (x,y) and (x+1, y+1).

If want to maintain geometric magnitudes should address draw only pixels interior to the object.

Fill-Area Primitives

- How do we draw interiors of object? These are called fill areas. Will discuss methods on Monday.