Tracing a Piriform

Abstract

This paper will discuss the strategy for tracing an implicitly defined, arbitrarily positioned, scaled, and oriented piriform using the CS400 ray tracing model. Another name for a piriform is peg top.

Implicit Piriform

The following is the implicit piriform curve equation that is discussed in this paper:

1) $$(X^4 - X^3) + Y^2 + Z^2 = 0$$

Listing (1) is an implicit piriform that is aligned at the origin and symmetric to all but one axis.

We define our ray as having an origin at $<x_0, y_0, z_0>$ and direction vector $<x_1, y_1, z_1>$. For some values of **t**, our ray satisfies the piriform surface equation as the following expresses:

2) $$X = x_0 + tx_1$$

3) $$Y = y_0 + ty_1$$

4) $$Z = z_0 + tz_1$$

We substitute our ray into the implicit piriform equation and solve for **t**. After some algebra, the result is a fourth degree (quartic) polynomial in **t**. The following shows steps that are performed in order to solve for the coefficients of the piriform surface equation in terms of t:

$$(x^4 - x^3) + y^2 + z^3 = 0$$

$$((x_0 + t x_1)^4 - (x_0 + t x_1)^3) + (y_0 + t y_1)^2 + (z_0 + t z_1)^2 = 0$$

$$(x_0 + t x_1)(x_0^3 + t(3x_0^2 x_1) + t^2(3x_0 x_1^2) + t^3(x_1^3)) - (x_0 + t x_1)^3$$
$$+ (y_0 + t y_1)^2 + (z_0 + t z_1)^2 = 0$$

$$x_0^4 + t(3x_0^3 x_1) + t^2(3x_0^2 x_1^2) + t^3(x_0 x_1^3) +$$
$$t(x_0^3 x_1) + t^2(3x_0^2 x_1^2) + t^3(3x_0 x_1^3) + t^4(x_1^4) -$$
$$(x_0^3 + t(3x_0^2 x_1) + t^2(3x_0 x_1^2) + t^3(x_1^3)) +$$
$$y_0^2 + t(2y_0 y_1) + t^2(y_1^2) + z_0^2 + t(2z_0 z_1) + t^2(z_1^2) = 0$$

$$x_0^4 - x_0^3 + y_0^2 + z_0 + t(3x_0^3 x_1 + x_0^3 x_1 - 3x_0^2 x_1 + 2y_0 y_1 + 2z_0 z_1) +$$
$$t^2(3x_0^2 x_1^2 + 3x_0^2 x_1^2 - 3x_0 x_1^2 + y_1^2 + z_1^2) +$$
$$t^3(x_0 x_1^3 + 3x_0 x_1^3 - x_1^3) + t^4(x_1^4) = 0$$

$$x_0^4 - x_0^3 + y_0^2 + z_0 + t(4x_0^3 x_1 - 3x_0^2 x_1 + 2y_0 y_1 + 2z_0 z_1) +$$
$$t^2(6x_0^2 x_1^2 - 3x_0 x_1^2 + y_1^2 + z_1^2) +$$
$$t^3(4x_0 x_1^3 - x_1^3) + t^4(x_1^4) = 0$$

Root Finding

Now that we have an implicit equation for the piriform, we need to find the roots (i.e. the values of **t** that satisfy the equation). To do this, we will implement a numerical polynomial root finder using Sturm sequences in order to determine the brackets for the roots and a False Position algorithm in order to determine the approximate values of the roots once the brackets have been obtained. Source for computing roots for polynomials using Sturm sequences and False Position can be found in the diskette that accompanies the book Graphics Gems Volume Five.

Normal Computation

In order to compute the normal for the piriform surface, we compute the gradient of equation (1). We compute the partial derivative of equation (1) as the following:

$f_x = 4X^3 - 3X^2$
$f_y = 2Y$
$f_z = 2Z$

Therefore, the normal is $<f_x, f_y, f_z>$.

Tracing an Arbitrarily Positioned, Scaled, and Oriented Piriform

Now that we have an equation for a prirform in terms of t and a normal in terms of a point on the surface of the prirform, we wish to intersect rays with a prirform that is given any position, scale, and orientation specified in world coordinates. From our ray tracer, we are given rays in world coordinates. However, the ray used to compute the degree 4 polynomial in t is assumed to be in object coordinates. Therefore, we transform the ray from world coordinates to object coordinates before solving the degree 4 polynomial in t.

The transformation from object coordinates to world coordinates is a matrix composed of columns which make up a 3 dimensional orthogonal basis which scales and orients the prirform in the world and the inverse of the matrix is the used to transform the ray from world coordinates to object coordinates like the following suggests:

Let **u**, **v**, and **w** form an orthogonal basis that orients and scales a prirform in world coordinates.

(5) [object to world transform] = $[\mathbf{u}^T\ \mathbf{v}^T\ \mathbf{w}^T]$

(6) [world to object transform] = [object to world transform] $^{-1}$ =

$[\mathbf{u}^T\ \mathbf{v}^T\ \mathbf{w}^T]^{-1}$

Now that we have a ray in object coordinates, we solve the degree 4 equation for the minimum t value and we use equations (1 - 3) in order to compute the closest intersection in object coordinates. The object coordinate normal is calculated by substituting the object coordinate intersection point into equations f_x, f_y, and f_z from the normal calculation.

Now we have the intersection point and normal at the point and they are both in object coordinates. Lastly, we transform the normal and point of intersection from object coordinates to world coordinates. In order to transform the intersection point, we use the matrix (5). However, the normal is transformed with the transpose of the matrix (6) or the inverse transpose of the matrix (5).

Specifications

Our prirform is specified using the following format:

PIRIFORM (position) (x-axis vector) (y-axis vector) (z-axis vector) (surface properties)

The keyword PIRIFORM informs the parser that the object that is being specified is a piriform. It is followed by the position of the object in world coordinates. The last three values are vectors that define the scale and orientation of a piriform in world coordinates. The last three vectors should be an orthogonal basis and correspond to **u**, **v**, and **w** from (5).

Conclusion

I hope that this paper simplifies the process of ray tracing an implicit equation and specifically a piriform equation. Other implicit equations can be used in order to change the object shape, but the simplifications steps are similar. Good luck, and have fun.

Contact

You may contact the author by emailing ngiuliet@digipen.edu.