Ray Tracing Generalized Cylinders

Pramook Khungurn

December 11, 2019

This document is written as a companion in reading the paper "Visible Light and X-Ray Ray Tracing of Generalized Cylinders" by Hsu and Chelberg [3].

1 Generalized Cylinders

- A *generalized cylinder* is a volumetric solid generated by sweeping an arbitrarily shaped closed 2D curve along an arbitrary three-dimensional space curve. The space curve is called the *axis*.
- We denote the axis with the function $\boldsymbol{a}(u) = (a_x(u), a_y(u), a_z(u))$ where $u_i \leq u \leq u_f$.
- We assume that $\boldsymbol{a}'(u) \neq 0$ for all $u \in [u_i, u_f]$.
- The *Frenet frame* gives an orientation of the axis curve at each point along it. The frame depends only on the local shape of the curve, not on the parameterization and the coordinate system in which the curve is defined. The frame consists of three orthogonal unit vector:

$$\begin{aligned} \hat{\boldsymbol{t}}(u) &= \frac{\boldsymbol{a}'(u)}{\|\boldsymbol{a}'(u)\|} && \text{(the tangent)} \\ \hat{\boldsymbol{b}}(u) &= \frac{\hat{\boldsymbol{t}}(u) \times \boldsymbol{a}''(u)}{\|\hat{\boldsymbol{t}}(u) \times \boldsymbol{a}''(u)\|} && \text{(the binormal vector)} \\ \hat{\boldsymbol{n}}(u) &= \hat{\boldsymbol{b}}(u) \times \hat{\boldsymbol{t}}(u) && \text{(the normal vector)} \end{aligned}$$

- The plane spanned by the normal vector and the binormal vector is called the *normal plane.* It is in this normal plane that the cross section is defined.

  We will use $\hat{\boldsymbol{n}}(u)$ and $\hat{\boldsymbol{b}}(u)$ as the first and the second coordinate axis of the normal plane.
- The cross section is defined as a closed curve $\boldsymbol{cs}(v) = (cs_n(v), cs_b(v))$ where $v_i \leq v \leq v_f$.
- For each u, the *contour* of the generalized cylinder is given by the curve

$$\boldsymbol{c}(u,v) = (c_n(u,v), c_b(u,v)) = \boldsymbol{s}(\boldsymbol{cs}(v), u)$$

  where $\boldsymbol{s}$ is the sweeping rule that specifies how the cross section changes along the axis of the genalized cylinder.
- The generalized cylinder is given by:

$$\boldsymbol{\Gamma}(u,v) = \boldsymbol{a}(u) + c_n(u,v)\hat{\boldsymbol{n}}(u) + c_b(u,v)\hat{\boldsymbol{b}}(u).$$

- One problem that might come up is that the Frenet frame is not defined when $\hat{\boldsymbol{t}}(u) \times \boldsymbol{a}''(u) = \mathbf{0}$. This happens only when $\boldsymbol{a}(u)$ is locally a straight line at u. That is, either $\boldsymbol{a}(u)$ is a straight line or u is an inflection point of the axis curve. The paper describes an implementation issue arising from this, and we shall cover it.

2 Ray Tracing Generalized Cylinders

- The first paper that deals with ray tracing of generalized cylinders is by Bronsvoort and Klok [1]. This work is later extended to allowing the cross section to be scaled along two orthogonal directions [2].
- Hsu and Chelberg [3] describes a ray tracing algorithm for generalized cylinders whose sweeping rule is invertible. This means that it can handle scaling and rotation, which is something that we want to use.
- The ray is given by a parameter curve $\boldsymbol{r}(w) = \boldsymbol{o} + w\boldsymbol{d}$.
- The ray tracing problem is find (u, v, w) where $\boldsymbol{\Gamma}(u, v) = \boldsymbol{r}(w)$.
- The approach is to first consider the intersection of the ray with each normal plane. These intersection points give rise to a 2D curve $\boldsymbol{\rho}(u)$.

  Now, if the cross section is constant, we can find the intersection point by finding the intersetion between $\boldsymbol{\rho}(u)$ and $\boldsymbol{cs}(v)$. This is exactly what is described in [1].

  However, in our case, the cross section varies along with u, so we need a more sophisticated technique.
- Hsu and Chelberg proposes applying the inverse of the sweeping rule to $\boldsymbol{\rho}(u)$ to get another curve $\boldsymbol{\phi}(u)$. Then, we can intersect $\boldsymbol{\phi}(u)$ with $\boldsymbol{cs}(v)$.

2.1 Computing $\boldsymbol{\rho}(u)$

- For each u, we project $\boldsymbol{v} = \boldsymbol{o} + w\boldsymbol{d} - \boldsymbol{a}(u)$ to the coordinate system defined by the Frenet frame at u. This is done by multiplying $\boldsymbol{v}(w, u)$ with the matrix $\boldsymbol{M}$ whose rows are the vectors $\hat{\boldsymbol{t}}(u)$, $\hat{\boldsymbol{n}}(u)$, and $\hat{\boldsymbol{b}}(u)$:

  $$\boldsymbol{r}_l(w, u) = \boldsymbol{M}\boldsymbol{v}$$

  where

  $$\boldsymbol{M} = \begin{bmatrix} \hat{\boldsymbol{t}}(u)^T \\ \hat{\boldsymbol{n}}(u)^T \\ \hat{\boldsymbol{b}}(u)^T \end{bmatrix}.$$

- For each u, the intersection of the ray $\boldsymbol{r}(w)$ and the normal plane can be found using the equation:

  $$\hat{\boldsymbol{t}}(u) \cdot (\boldsymbol{o} + w\boldsymbol{d} - \boldsymbol{a}(u)) = 0.$$

  Substituting $\hat{\boldsymbol{t}}(u) = \boldsymbol{a}'(u)/\|\boldsymbol{a}'(u)\|$, we have

  $$\boldsymbol{a}'(u) \cdot (\boldsymbol{o} + w\boldsymbol{d} - \boldsymbol{a}(u)) = 0.$$

  So,

  $$w = -\frac{\boldsymbol{a}'(u) \cdot (\boldsymbol{o} - \boldsymbol{a}(u))}{\boldsymbol{a}'(u) \cdot \boldsymbol{d}}$$

  provided that $\boldsymbol{a}'(u) \cdot \boldsymbol{d} \neq 0$. If this is not true, there are two cases:

  - If $\boldsymbol{a}'(u) \cdot (\boldsymbol{o} - \boldsymbol{a}(u)) = 0$ as well, then the ray lies in the normal plane. The problem then becomes finding the intersection of the ray with the contour directly.
  - If $\boldsymbol{a}'(u) \cdot (\boldsymbol{o} - \boldsymbol{a}(u)) \neq 0$, then there can be no intersection. We thus exclude this u from the description of $\boldsymbol{\rho}(u)$.

- We can find the coordinate of the intersection point in the normal plane by substituting the above w and u to $\boldsymbol{r}_l(w,u)$. This gives:

$$\boldsymbol{\rho}(u) = \begin{bmatrix} \hat{\boldsymbol{n}}(u) \cdot \big(\boldsymbol{o} - \frac{\boldsymbol{a}'(u)\cdot(\boldsymbol{o}-\boldsymbol{a}(u))}{\boldsymbol{a}'\cdot\boldsymbol{d}}\boldsymbol{d} - \boldsymbol{a}(u)\big) \\ \hat{\boldsymbol{t}}(u) \cdot \big(\boldsymbol{o} - \frac{\boldsymbol{a}'(u)\cdot(\boldsymbol{o}-\boldsymbol{a}(u))}{\boldsymbol{a}'\cdot\boldsymbol{d}}\boldsymbol{d} - \boldsymbol{a}(u)\big). \end{bmatrix}$$

2.2 Computing $\boldsymbol{\phi}(u)$

- Because we want to find (u,v) such that $\boldsymbol{\rho}(u) = \boldsymbol{c}(u,v) = \boldsymbol{s}(\boldsymbol{cs}(v),u)$. Given that $\boldsymbol{s}(\cdot,u)$ is invertible for all u, the equation is equivalent to

$$\boldsymbol{s}^{-1}(\boldsymbol{\rho}(u),u) = \boldsymbol{cs}(v).$$

The curve $\boldsymbol{\phi}(u)$ is thus defined as $\boldsymbol{s}^{-1}(\boldsymbol{\rho}(u),u)$.

2.3 Finding Intersection Points

- The curves $\boldsymbol{\phi}(u)$ and $\boldsymbol{cs}(v)$ are split at inflection points and points with horizontal or vertical tangents. As a result, the segments obtained are bounded within the rectangle defined by their end points.
- If each of the segments are subdivided, each subdivided segment is also bounded by the box defined by their end points.
- For each pair of $\boldsymbol{\phi}(u)$ and $\boldsymbol{cs}(v)$ segments, the following intersection routine is called:

```
procedure INTERSECT($curve_1, curve_2$)
  if not OVERLAP($curve_1, curve_2$)
    return
  if IS-LINEAR($curve_1$) and IS-LINEAR($curve_2$)
    LINE-INTERSECT($curve_1, curve_2$)
  else if IS-LINEAR($curve_1$) and not IS-LINEAR($curve_2$)
    $curve_{21}, curve_{22}$ = DIVIDE-CURVE($curve_2$)
    INTERSECT($curve_1, curve_{21}$)
    INTERSECT($curve_1, curve_{22}$)
  else if IS-LINEAR($curve_2$) and not IS-LINEAR($curve_1$)
    $curve_{11}, curve_{12}$ = DIVIDE-CURVE($curve_2$)
    INTERSECT($curve_{11}, curve_2$)
    INTERSECT($curve_{12}, curve_2$)
  else
    $curve_{11}, curve_{12}$ = DIVIDE-CURVE($curve_2$)
    $curve_{21}, curve_{22}$ = DIVIDE-CURVE($curve_2$)
    INTERSECT($curve_{11}, curve_{21}$)
    INTERSECT($curve_{12}, curve_{21}$)
    INTERSECT($curve_{11}, curve_{22}$))
    INTERSECT($curve_{12}, curve_{22}$)
```

Here,

- OVERLAP($curve_1, curve_2$) checks whether the bounding boxes of $curve_1$ and $curve_2$ overlaps.
- IS-LINEAR($curve$) tests whether the curve segment is linear enough. We will cover how to do this later.
- DIVIDE-CURVE($curve, curve_1, curve_2$) subdivides the curve at the midpoint of the range of parameter values.

- The aforementioned linearity test calculate the distance between:
  - the intersection between the tangent lines at the end-points of the segment, and
  - the line joining the end points.

  If the distance is less than ε, the segment is considered linear.

- To compute the distance, we may consider the x and the y axes separately. Let the two end-points be denoted by A and B. Let the u parameters at A and B be u_0 and u_1, respectively. Also, let us refer to the x-coordinates of the two points as x_0 and x_1, and let the derivative of x with respective to u at A and B be x'_0 and x'_1, respectively.

  The x-coordinate of the tangent line at A is given by:
  $$x = x_0 + (u - u_0)x'_0.$$
  Similarly, the x-coordinate of the tangent line at B is given by:
  $$x = x_1 + (u - u_1)x'_1.$$
  Solving, we get
  $$\begin{aligned} u^* &= \frac{x_0 - x_1 + u_1 x'_1 - u_0 x'_0}{x'_1 - x'_0}, \text{ and} \\ x^* &= x_0 + x'_0(u^* - u_0). \end{aligned}$$
  The equation of the chord is:
  $$x(u_1 - u_0) - (x_1 - x_0)u + [(x_1 - x_0)u_0 - x_0(u_1 - u_0)] = 0.$$
  The perpendicular distance of (u^*, x^*) from the chord is given by:
  $$d_x = \left| \frac{(u_1 - u_0)x^* + (x_0 - x_1)u^* + [(x_1 - x_0)u_0 - x_0(u_1 - u_0)]}{\sqrt{(u_1 - u_0)^2 - (x_0 - x_1)^2}} \right|.$$
  We can compute d_y in a similar manner, and the total distance is given by $d = \sqrt{d_x^2 + d_y^2}$.

2.4 Normal Calculation

- The surface normal at (u, v) is given by
  $$\hat{\boldsymbol{N}}(u, v) = \frac{\partial \boldsymbol{\Gamma}(u, v)}{\partial v} \times \frac{\partial \boldsymbol{\Gamma}(u, v)}{\partial u}$$
  where
  $$\begin{aligned} \frac{\partial \boldsymbol{\Gamma}(u, v)}{\partial u} &= \boldsymbol{a}'(u) + \hat{\boldsymbol{n}}(u)\frac{\partial c_n(u, v)}{\partial u} + c_n(u, v)\hat{\boldsymbol{n}}'(u) + \hat{\boldsymbol{b}}(u)\frac{\partial c_b(u, v)}{\partial u} + c_b(u, v)\hat{\boldsymbol{b}}'(u) \\ \frac{\partial \boldsymbol{\Gamma}(u, v)}{\partial v} &= \frac{\partial c_n(u, v)}{\partial v}\hat{\boldsymbol{n}}(u) + \frac{\partial c_b(u, v)}{\partial v}\hat{\boldsymbol{b}}(u). \end{aligned}$$

- The derivatives $\hat{\boldsymbol{n}}'(u)$ and $\hat{\boldsymbol{b}}'(u)$ can be computed by the Frenet-Serret formualas:
  $$\begin{aligned} \hat{\boldsymbol{n}}'(u) &= -\kappa(u)\hat{\boldsymbol{t}}(u) + \tau(u)\hat{\boldsymbol{b}}(u) \\ \hat{\boldsymbol{b}}'(u) &= -\tau(u)\hat{\boldsymbol{n}}(u) \end{aligned}$$
  where $\kappa(u)$ is the *curvature* and $\tau(u)$ is the *torsion*. They are defined as:
  $$\begin{aligned} \kappa(u) &= \frac{\|\boldsymbol{a}'(u) \times \boldsymbol{a}''(u)\|}{\|\boldsymbol{a}'(u)\|^3}, \text{ and} \\ \tau(u) &= \frac{(\boldsymbol{a}' \times \boldsymbol{a}''(u)) \cdot \boldsymbol{a}'''(u)}{\|\boldsymbol{a}'(u) \times \boldsymbol{a}''(u)\|^2}. \end{aligned}$$

3 Hair Geometry

- The cross section is a circle:

$$\boldsymbol{cs}(v) = \begin{bmatrix} \cos v \\ \sin v \end{bmatrix}$$

  where $v \in [0, 2\pi]$.

- We shall represent the axis as a Bezier curve $\boldsymbol{a}(u) = \boldsymbol{a}^{(0)} + \boldsymbol{a}^{(1)}u + \boldsymbol{a}^{(2)}u^2 + \boldsymbol{a}^{(3)}u^3$ where $u \in [0, 1]$.

- We would like to allow the user to specify the following parameters:

  - The major radius $a(u) = a_0 + (a_1 - a_0)u$.
  - The minor radius $b(u) = b_0 + (b_1 - b_0)u$.
  - The in-plane rotation angle $\theta(u) = \theta_0 + (\theta_1 - \theta_0)u$.

  The sweeping rule is given by:

$$\boldsymbol{s}(\boldsymbol{cs}(v), u) = S(u)\boldsymbol{cs}(v) = \begin{bmatrix} \cos(\theta(u)) & -\sin(\theta(u)) \\ \sin(\theta(u)) & \cos(\theta(u)) \end{bmatrix} \begin{bmatrix} a(u) & 0 \\ 0 & b(u) \end{bmatrix} \boldsymbol{cs}(v).$$

  As such, the inverse of the sweeping rule is given by:

$$S^{-1}(u) = \begin{bmatrix} 1/a(u) & 0 \\ 0 & 1/b(u) \end{bmatrix} \begin{bmatrix} \cos(\theta(u)) & \sin(\theta(u)) \\ -\sin(\theta(u)) & \cos(\theta(u)) \end{bmatrix}.$$

- Now, if we want to intersect the ray $\boldsymbol{r}(w) = \boldsymbol{o} + w\boldsymbol{d}$, we first have to compute the points where $\boldsymbol{a}'(u) \cdot \boldsymbol{d} = 0$. We have that

$$\boldsymbol{a}'(u) = \boldsymbol{a}^{(1)} + 2\boldsymbol{a}^{(2)}u + 3\boldsymbol{a}^{(3)}u^2$$

  So,

$$\boldsymbol{a}'(u) \cdot \boldsymbol{d} = (\boldsymbol{a}^{(1)} \cdot \boldsymbol{d}) + 2(\boldsymbol{a}^{(2)} \cdot \boldsymbol{d})u + 3(\boldsymbol{a}^{(3)} \cdot \boldsymbol{d})u^2.$$

  Therefore, $\boldsymbol{a}'(u) \cdot \boldsymbol{d} = 0$ is a quadratic equation.

References

[1] Willem F. Bronsvoort and Fopke Klok. Ray tracing generalized cylinders. *ACM Trans. Graph.*, 4(4):291–303, October 1985.

[2] Willem F. Bronsvoort, Peter R. van Nieuwenhuizen, and Frits H. Post. Display of profiled sweep objects. *The Visual Computer*, 5:147–157, 1989. 10.1007/BF01901389.

[3] Jean Hsu and David M. Chelberg. Visible light and x-ray ray tracing of generalized cylinders. In *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, pages 392–401, 1994.