Swept Surfaces

Swept surfaces are generated by moving a section curve along a trajectory curve.

Denoting the trajectory curve as ***T***(*v*) and the section curve as ***C***(*u*), a general swept surface is given by:

$$\boldsymbol{S}(u, v) = \boldsymbol{T}(v) + \boldsymbol{M}(v)\boldsymbol{C}(u)$$

where $M(v)$ is a 3 x 3 matrix incorporating rotation, and nonuniform scaling as a function of v.

This general formulation, however, can lead to unwanted behavior, e.g., self-intersection, degeneracies, and discontinuities

In practice, most swept surfaces are restricted to one of two types:

1) M(v) is the identity matrix, i.e., C(u) is simply translated by T(v).

2) M(v) is not the identity matrix.

Case 1 is referred to as a general translational sweep:

$$\boldsymbol{S}(u, v) = \boldsymbol{T}(v) + \boldsymbol{C}(u)$$

This swept surface is defined as follows, Let:

$$\boldsymbol{T}(v) = \frac{\sum_{j=0}^{m} N_{j,q}(v)\, w_j^T \boldsymbol{T}_j}{\sum_{j=0}^{m} N_{j,q}(v)\, w_j^T}$$

$$V = \{v_0, \ldots, v_s\}$$

and,

$$\boldsymbol{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\, w_i^{C} \boldsymbol{Q}_i}{\sum_{i=0}^{n} N_{i,p}(u)\, w_i^{C}}$$

$$U = \{u_0, \ldots, u_r\}$$

Then the swept surface is defined by,

$$\boldsymbol{S}(u, v) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{ij} \boldsymbol{P}_{ij}}{\sum_{i=0}^{m} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{ij}}$$

on knot vectors *U* and *V*.

With control points:

$$P_{ij} = T_j + Q_i$$

$$i = 0, \ldots, n \qquad j = 0, \ldots, m$$

and weights:

$$w_{i,j} = w_i^C w_j^T$$

$$i = 0, \ldots, n \qquad j = 0, \ldots, m$$

See Figure 10.11

For case 2 things get more interesting...

Let {**O**, **X**, **Y**, **Z**} denote the global coordinate system.

Then introduce a local orthonormal coordinate system:

$$\{\boldsymbol{o}(v), \boldsymbol{x}(v), \boldsymbol{y}(v), \boldsymbol{z}(v)\}$$

which moves along ***T***(*v*).

Define,

$$\boldsymbol{o}(v) = \boldsymbol{T}(v) \qquad \boldsymbol{x}(v) = \frac{\boldsymbol{T}'(v)}{|\boldsymbol{T}'(v)|}$$

We need one more independent vector to define the local moving coordinate system.

Suppose ***B***(*v*) is defined so that it satisfies

$$\boldsymbol{B}(v) \cdot \boldsymbol{x}(v) = 0$$

Now, set

$$\boldsymbol{z}(v) = \frac{\boldsymbol{B}(v)}{\left|\boldsymbol{B}(v)\right|}$$

and,

$$\boldsymbol{y}(v) = \boldsymbol{z}(v) \times \boldsymbol{x}(v)$$

We will return to the question of choosing ***B***(*v*) later.

Finally, to incorporate differential scaling, introduce a three-dimensional vector function,

$$\boldsymbol{s}(v) = (s_x(v), s_y(v), s_z(v))$$

Now, the more general form of the swept surface can be written as,

$$\boldsymbol{S}(u, v) = \boldsymbol{T}(v) + \boldsymbol{A}(v)\, S(v)\, \boldsymbol{C}(u)$$

where,

- S(v) is a 3 x 3 diagonal matrix of elements from s(v)
- A(v) is the general transformation matrix between global, {**O**, **X**, **Y**, **Z**}, and local, {**o(v), x(v), y(v), z(v)**} coordinate systems. (i.e., the rows of A(v) are **x(v), y(v), z(v)).**

If $A(v)$ and $S(v)$ can be precisely represented as rational B-spline functions, then so can $\boldsymbol{S}(u,v)$.

However, $A(v)$ is generally not representable as a NURBS curve. So we have to approximate. There are several approaches...

here are two:

1) Use

$$\boldsymbol{S}(u, v) = \boldsymbol{T}(v) + A(v) S(v) \boldsymbol{C}(u)$$

to evaluate an *n* x *m* grid of points lying on ***S***(*u*,*v*), and then interpolate, or approximate

2) Transform and place ***C***(*u*) at K + 1 instances, and use skinning to get ***S***(*u*,*v*). Increasing K increases the accuracy. (See algorithms A10.1 and A10.2)

What about $\boldsymbol{B}(v)$?

If the trajectory curve $\boldsymbol{T}(v)$ is twice differentiable, we can use the *Frenet* frame, i.e., define,

$$\boldsymbol{B}(v) = \frac{\boldsymbol{T}'(v) \times \boldsymbol{T}''(v)}{\left|\boldsymbol{T}'(v) \times \boldsymbol{T}''(v)\right|}$$

and,

$$\boldsymbol{y}(v) = \boldsymbol{N}(v) = \boldsymbol{B}(v) \times \boldsymbol{T}'(v)$$

T’(*v*), ***N***(*v*), and ***B***(*v*) are mutually orthogonal for all *v*, and when normalized form a moving local coordinate system on ***T***(*v*) called the Frenet frame.

B(*v*), so defined, can be used in algorithms A10.1 and A10.2. However, several problems can arise.

1) B(v) is not defined for linear segments or at inflection points, i.e., where

$$\boldsymbol{T}'(v) \times \boldsymbol{T}''(v) = 0$$

2) B(v) flips abruptly to the opposite direction at an inflection point.

3) For 3D trajectories, B(v) can rotate excessively around T(v) causing unwanted twisting of the swept surface.

Your text presents a method to control this twisting behavior while taking advantage of the Frenet frame.