Camera movement along a spline

Christopher Twigg

March 17, 2003

When we simulate our roller coaster, one of the key points to get right is camera position and orientation. Position is a solved problem, since we know how to evaluate points along a cubic curve. The remaining question is one of orientation; from a particular point along the curve, how do we decide what the Look and Up vectors are? (see figure 1)

We will make use of a construction from differential geometry called the "Frenet frame." Suppose we have a parameterized curve $\mathbf{f}(s)$ in $\mathbb{R}^3$. We start with the unit-length tangent vector to the surface, $\mathbf{T}(s) = \mathbf{f}'(s)/|\mathbf{f}'(s)|$. We can then calculate the *principle normal* vector $\mathbf{N}(s)$ and the binormal vector $\mathbf{B}(s)$ using the following equations,

$$\mathbf{T}(s) = \frac{\mathbf{f}'(s)}{|\mathbf{f}'(s)|}$$
$$\mathbf{B}(s) = \frac{\mathbf{f}'(s) \times \mathbf{f}''(s)}{|\mathbf{f}'(s) \times \mathbf{f}''(s)|}$$
$$\mathbf{N}(s) = \mathbf{B}(s) \times \mathbf{T}(s)$$

Note that when these three vectors exist, they form an orthonormal basis, with $\mathbf{T}(s)$ always pointing along the curve. Furthermore, if $\mathbf{f}''(s)$ doesn't vanish, these will be always continuous, which is nice because it ensures that our camera will never flip over or do anything similarly unpleasant. However, this is too restrictive a condition; $\mathbf{f}''(s)$ will be zero at any point where the curve changes direction or is perfectly straight.

Hence, we suggest as an alternative the following iterative method due to Ken Sloan [1] (figure 2). Suppose we are given some starting reference frame at the position $\mathbf{f}(0)$, $(\mathbf{T}_0, \mathbf{N}_0, \mathbf{B}_0)$. Now, given a reference frame

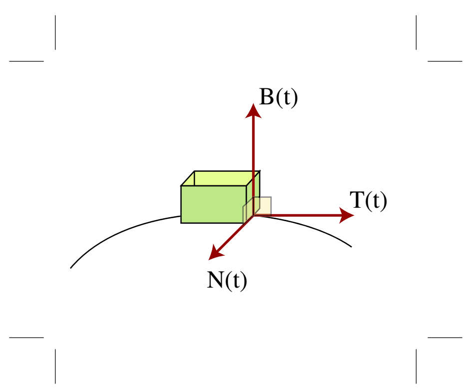

Figure 1: A camera reference frame

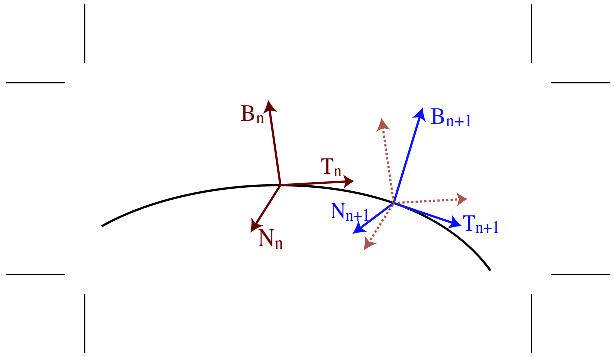

Figure 2: Generating a new reference frame at parameter value s_{n+1}

$(\mathbf{T}_n, \mathbf{N}_n, \mathbf{B}_n)$ at position $\mathbf{f}(s_n)$ we calculate the reference frame $(\mathbf{T}_{n+1}, \mathbf{N}_{n+1}, \mathbf{B}_{n+1})$ by

$$\begin{aligned}\mathbf{T}_{n+1} &= \mathbf{T}(s_{n+1})\\ \mathbf{N}_{n+1} &= \mathbf{B}_n \times \mathbf{T}_{n+1}\\ \mathbf{B}_{n+1} &= \mathbf{T}_{n+1} \times \mathbf{N}_{n+1}\end{aligned}$$

This has other uses as well; consider a cross-section of the track at point $\mathbf{f}(s_n)$. We can construct a transformation matrix using the quantities $\mathbf{f}(s_n)$, $\mathbf{T}_n$, $\mathbf{N}_n$, and $\mathbf{B}_n$ that will move our cross-section into place; consult the "Transformations and Viewing" lecture for suggestions (pay special attention to the way matrices are represented in OpenGL!)

References

[1] Bloomenthal, J. Calculation of reference frames along a space curve. In *Graphics Gems*. 1990, pp. 567–571.

[2] DoCarmo, M. P. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., 1976.