Paths for Volume Slices

by Demetrios Halazonetis

www.dhal.com

March 2007

The paths for the volume slices are cubic splines. They have control points for adjusting their shape and evenly spaced ‘vectors’ for adjusting the twist of the path.

The most difficult part of coding the slice paths was to figure out how the slice would be oriented. Specifying that the slice would be perpendicular to the path is not difficult, but what should the ‘up’ direction be? It was evident from the beginning that this direction could not be specified based on global reference axes. For example, if we want the path to follow along a cranial suture, then we would like the slice to always be oriented such that the inside of the cranium is in the ‘down’ direction and the outside in the ‘up’ direction, as shown below:

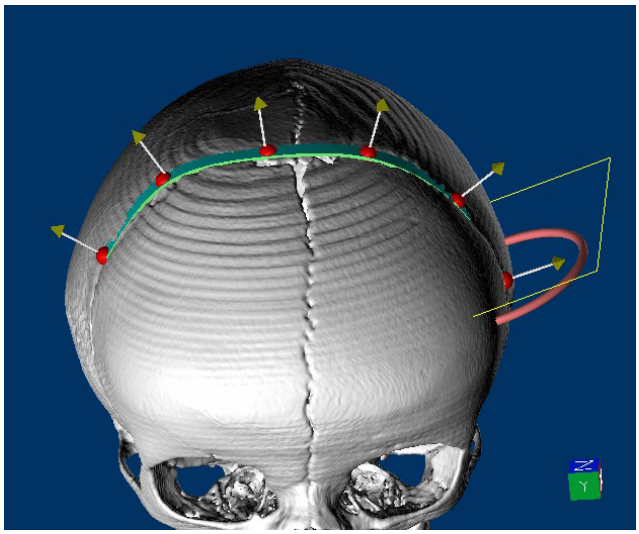

Fig. 1. A slice path with the control points of the spline drawn as red spheres and the ‘up’ vectors (which do not necessarily coincide with the control points). The outline of the slice is shown in yellow. The red torus is for adjusting the path’s overall twist.

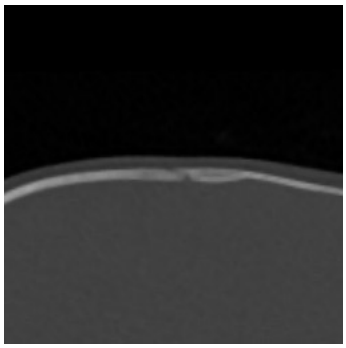

Fig. 2. The slice, showing the suture.

Here ‘up’ means relative to the slice image (Fig. 2) and not relative to the patient’s body axis or to the global 3D axes.

Since the path itself should be our reference system, a seemingly good idea is to use the path’s curvature to specify the reference frame. A very well-known reference frame for curves is the Frenet frame (see Wikipedia: http://en.wikipedia.org/wiki/Frenet_vector#Frenet_frame). This consists of three mutually perpendicular unit vectors; the tangent to the curve, the normal, which is perpendicular to the tangent and points towards the center of the osculating circle(see Wikipedia: http://en.wikipedia.org/wiki/Osculating_circle), and the binormal (the vector product of the tangent and normal). It is not difficult to calculate these vectors. The spline used to draw the path is already discretized for drawing in OpenGL, so it is easy to calculate the tangent as:

$\mathbf{T} = \mathbf{P}_i - \mathbf{P}_{i-1}$

where P is a vector of the coordinates of the ith point.

The normal is found by subtracting two consecutive tangents, and the binormal is calculated from the cross product of the tangent and normal (don’t forget to normalize all vectors to unit length).

However, the Frenet frame has two major shortcomings. It is not defined for areas of the curve that have zero curvature and it flips at inflection points (Fig. 3).

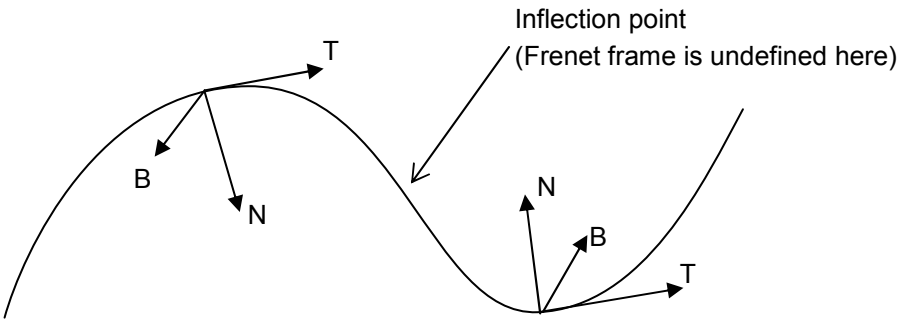

Fig. 3. Frenet frames along a curve. Note how the frame flips after passing through the inflection point.

The problem with the Frenet frame is the twisting of the frame around the tangent vector. The solution is to disengage the frame from the curvature of the curve and keep only the tangent as the vector that ties the frame to the curve. The other vectors will be initialized to an arbitrary orientation (of course always keeping them orthogonal to each other and the tangent) and subsequent positions along the curve will be calculated by trying to minimize changes in the orientation of the frames.

There are various ways to achieve such a ‘parallel’ transport of the frame along the curve. The paper “Calculation of Reference Frames along a Space Curve” (http://www.unchainedgeometry.com/jbloom/pdf/ref-frames.pdf) by Jules Bloomenthal (http://www.unchainedgeometry.com/jbloom/) is a very good start. You could also read the PhD thesis of Robert Scharein (http://www.knotplot.com/thesis/), which has nice figures showing the twisting caused by the Frenet frames (pages 66-

68). I have used Sloan's method (described in the previous papers), which is very simple and seems to work very well. Essentially, the frames are calculated consecutively, and the updated normal is computed as the cross product of the current tangent with the previous binormal:

$\mathbf{N}_i = \mathbf{T}_i \times \mathbf{B}_{i-1}$

$\mathbf{B}_i = \mathbf{N}_i \times \mathbf{T}_i$

Of course, 'normal' in the above equations is not the normal to the curve as defined in the Frenet frames. It is perpendicular to the tangent but does not necessarily lie on the osculating plane.

The twisting of the curve can be adjusted by dragging on 'up vectors' positioned at regular intervals along the curve. These 'vectors' add an extra twist to the frames; this is calculated by dividing the twist of the 'vector' by the number of segments (frames) between consecutive 'vectors' and distributing the result to these frames. There is also a global twist control (the red torus in Fig. 1).

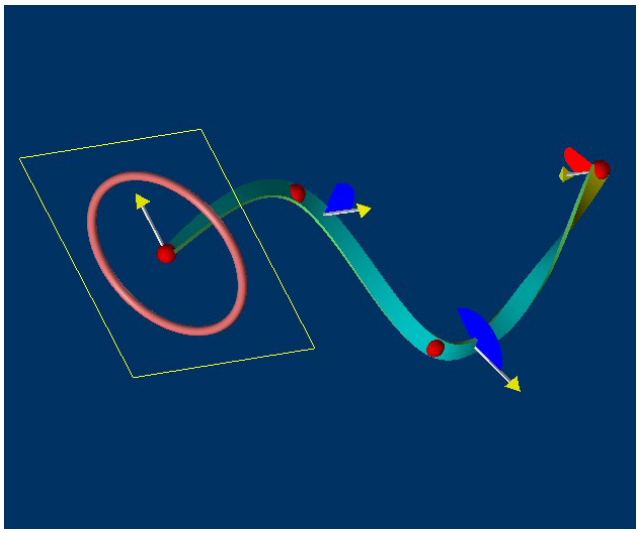

Fig. 4. Twist added at three positions along curve (denoted by blue and red partial disks). Note that twist positions are at equal distances along curve and do not coincide with spline control points.

The disadvantage of the parallel transport system is that reference frames need to be calculated iteratively, whereas the Frenet frame can be calculated at any position of the curve by using just the local properties of that position (the tangent and normal).