Math 351 Project - Summer 2019

Part I

Model Views Phase:

Use a 3D package such as DirectX, OpenGL, or others, to render a model in 3D from a fixed
view point. The model should be complex enough to exhibit well any slight changes in orientation
through changes in lighting. (A cube is too simple.) The program should then accept a list of
orientations (given as axis-angle pairs) as a text file, and provide an interface to step through these
orientations as applied to the model, keeping the view point fixed. Axis-Angle pairs should be in
the following form: (four floats)

(2, 1.5, 3), 45

Note: the axis does not need to be normalized, for ease of entry. You will normalize it in your code.
The angle is entered in degrees, again for ease of entry, but you will want to convert to radians in
your code.

Part II

Data Interpolation Phase:

In this phase the first part is extended by filling in the orientations with intermediate orientations
and printing out to text. This will be done first with Piecewise Slerp, a continuous but not smooth
interpolation, then with the first smooth interpolation called Shoemake Bezier. In addition to
printing out the intermediate quaternions, we will do a comparison of slopes. The slopes will be
calculated based on successive triples of quaternions, which are 4-dimensional points. If the three
points are $P_0$, $P_1$, and $P_2$, then we can compute the deviation of $P_2$ away from the line through $P_0$
and $P_1$ by using vectors: First let $v_1$ be the vector from $P_0$ to $P_1$ and let $v_2$ be the vector from $P_0$
to $P_2$. Then let $u$ be the projection of $v_2$ onto $v_1$, and let $v$ be $v_2 - u$. Then the slope $m = |v|/|u|$ is
the measure of deviation. If the three points are collinear in $\mathbb{R}^4$ then the slope $m$ is zero. The
slopes should be printed to a text file so that we can compare these two interpolation techniques.
Small values of $m$ are consistent with smooth curves. A big jump in $m$ indicates a corner.

Part III

Graphical Interpolation Phase:

In this phase the first two parts are extended by displaying the intermediate orientations applied to
the object as a continuous animation. In addition to the previous two interpolation techniques we
will also do Circular Blending, Spherical Quadrangle, and others. The program should implement at
least one of these additional techniques as well as the previous two (Piecewise Slerp and Shoemake
Bezier) and have an interface that allows the user to select the technique. The program should loop
through the orientations continuously.
alternative Part III

Extended Interpolation Phase:

In this alternative to Part III, students can do the numerical work in phase II, but extended to include one more interpolation technique. The output is required to compute 100 intermediate quaternions between each pair of key quaternions. The output data should include also the original axis-angle pairs, which make up the first column. The second column is the continuous list of quaternions, including both the key quaternions and the intermediate quaternions. The third column should be the slope computed based on the current quaternion and one before and one after. The slope can be computed for ever line of data except the first and last one.