Date: 4-9-98
Due Date: 4-16-98

CS 523: Scientific Visualization

Assignment 1

The purpose of this problem set is to familiarize you with issues involved in data conversion and interpolation and give you a little experience with **vtk**.

We have provided a 2D scalar data set in files `{.pts,.data}` in the Data Directory of the CS 523 web page. The `.pts` file gives the X Y Z coordinates at a set of 618 sample points. The Z coordinate is always zero and can be ignored. The `.data` file gives the electric field potential at each of these data points. The data is a simulation result of the electric field produced by the heart in a slice through a human thorax. The lines in these two files are in a one-to-one correspondence. For example, the 100 line in the `.pts` file gives the coordinates of the 100th data site and the 100'th line in the `.data` file gives the corresponding scalar voltage at that site.

This is a scattered data set: no connectivity information is provided. Your first step is to produce a triangulation between the data sites. We have provided a program `sweep2` which can compute the Delaunay triangulation of a 2D points set. The `sweep2` program is in Code Directory on the web page with an executable compiled for the SGI Indigos, source code, and a manual page.

The next step is to visualize the triangulated data set with **vtk** in several ways: as a height field (map the voltage to a height displacement), as a false colored plane (map the electric field potential to color), and using both cues together. You will have to convert the triangulated data set into the appropriate **vtk** format.

Finally, you will interpolate the data set onto a 2D Cartesian grid and convert to a **vtk** structured/regular file format for import into **vtk**. Repeat the visualizations done above using the regular grid data. The easiest way to interpolate the triangulated data is by scan conversion into a 2D array: Loop through all the triangles, for each triangle, scan conversion can be accomplished as follows:

1. Map the triangle vertices into the integer Cartesian grid coordinates.

2. Find the triangle vertex with the smallest y value (call this ymin).

3. Step through the y values from ymin up to the vertex with the next smallest y value. For each y, compute xleft and xright, the x coordinates along the two triangle edges and Dleft and Dright, the interpolated data values along each of these edges. Use an inner loop to interpolate these data values in the X direction across the scanline and fill in the grid array as the values are produced.

4. When the vertex with the second smallest vertex is reached, one of the triangle edges becomes inactive and the third triangle edge becomes active. Repeat step 2 using the 2 remaining edges.

You may have to experiment with different grid resolutions to find a result which achieves acceptable quality with a reasonable rendering time. 128x128 is a reasonable starting point.

What to turn in

Please turn in the source code for all programs that you write. Peter will send you email about how to turn in your images.

Please also include a short writeup describing what you did in the assignment and comparing the unstructured grid visualizations to the structured grid visualizations, in terms of quality and speed.