CS499/554 Geometric Modeling

Project I

Due April 25, 2025

The purpose of the this project is to help you

(1) become familiar with the .PLY format and typical data structures for mesh processing.
(2) review relevant mathematical concepts and their numerical computation algorithms.
(3) acquire experience and develop intuition with shapes.

Detailed tasks are listed below. Please submit everything necessary to compile and run your program, and a project report that contains results and discussions. **Note it is impossible to obtain a good grade without a well-written report. Figures are mostly welcome.** Feel free to explore and have fun!

Note: the program **learnply** is provided for your convenience. It contains three components: I/O to .ply files, visualization, and a user-interface. Only the first component is necessary for this project and future projects. Feel free to develop your own data structures for meshes. Also, you may use other tools to construct the user interface, such as glui, Microsoft API 32, or MFC. Be sure to provide a README file so that I know how to run your program.

1. **(Mesh coloring)** Add the following functionalities to **learnply**:

   a. Assign a unique color to each polygon based on its polygon ID. Use any simple scheme. (Polygon ID map).
   b. For each triangle, assign the first vertex with the red color (1, 0, 0), the second vertex with the green color (0, 1, 0), and the last vertex the blue color (0, 0, 1). (Barycentric coordinates map)
   c. Assign a color to each polygon based on its normal as follows, $R = |N_x|,\ G = |N_y|,\ B = |N_z|$. Compare this to the visualization that every vertex is colored based on its normal. (Normal Map).
   (Parts d-f are for CS554 students only)
   d. Assign a color to each vertex based on its 3D coordinates according to the following map $R = f(\lfloor V_x / L \rfloor)$, $G = f(\lfloor V_y / L \rfloor)$, $B = f(\lfloor V_z / L \rfloor)$ where L is a use-defined positive real number and $f(n) = \begin{cases} 0 & n \text{ is odd} \\ 1 & n \text{ is even} \end{cases}$. (3D checkerboard).
   e. Run your program using the schemes from (a-d) on the Bunny, the Happy Buddha, the Dragon, and the Feline. For the 3D checkerboard, try different L's. What problems do you encounter? Propose some solutions.

f. Compute the 3D checkerboard on the Icosahedron. Do you see any new problem(s) that weren't obvious with the other models? What do you think is the cause of the problem? Suggest some solution(s).

2. **(Corner)** Construct the corner list for a 3D model.

3. **(Topology and curvatures)** Use the corner data structure to construct a table of the following four quantities for every model provided with learnPly.

   a. The Euler characteristic V-E+F.
   b. The Gaussian curvature of the model defined as the total angle deficit. Note: the angle deficit of a vertex is 2π minus the total angle around the vertex from all incident triangles, and the total angle deficit of a model is the sum of angle deficit of all the vertices in the model.
   c. The discrete Gaussian curvature of the model defined as $\pi/3$ times the total valence deficit. Note, the valence deficit of a vertex is 6 minus the valence of the vertex, and the total valence deficit of a model is the sum of the valence deficit of all the vertices in the model.
   d. The number of handles in the model.

   For each model, circle its handle(s) and find the relationship between the Euler characteristic and the number of handles.

   Visualize the distribution of Gaussian curvature. Where do you likely see very positive Gaussian curvatures? What about very negative Gaussian curvatures? Can you justify your observations?

   Visualize the irregular vertices, whose valence deficits are not zero. Where do you see irregular vertices with positive valence deficits? How about irregular vertices with negative valence deficits? Do the locations of irregular vertices relate to the location of Gaussian curvatures? Should they be related? Why or why not? Can you see how to use the difference in the two curvatures as a way of measuring the quality of a mesh?