

CMSC 435/634: Introduction to Computer Graphics

Assignment 4

Mesh Fairing

Due November 6, 2018 @ 11:59 PM

The Assignment

The assignment is to write a program that performs [Mesh Fairing](#). Your program will read a file in a simplified obj format and write an obj file of a smoothed mesh. Your program should support the following usage:

```
smoothing input.obj output.obj lambda dt iterations
```

The output to the right is from

```
smoothing bunny.obj sbunny.obj 1 1 50
```

applied to [bunny.obj](#) and viewed in [MeshLab](#).

Your program should perform iterations of [Laplacian smoothing](#). For the basic assignment use the "umbrella operator" to approximate the Laplacian. Initialize a vec3 (laplacian) and a double (m) for each vertex for accumulating estimates of the Laplacian and the number of incident edges. Then loop over all triangles updating the accumulators for each vertex in the triangle (this will count each edge twice, but that is okay, since you will divide by the count). Then loop over the vertices updating their position (x) to

```
x += lambda*dt* (laplacian / m)
```

An executable of my program is at:

```
~adamb/public/Smoothing/smoothing
```

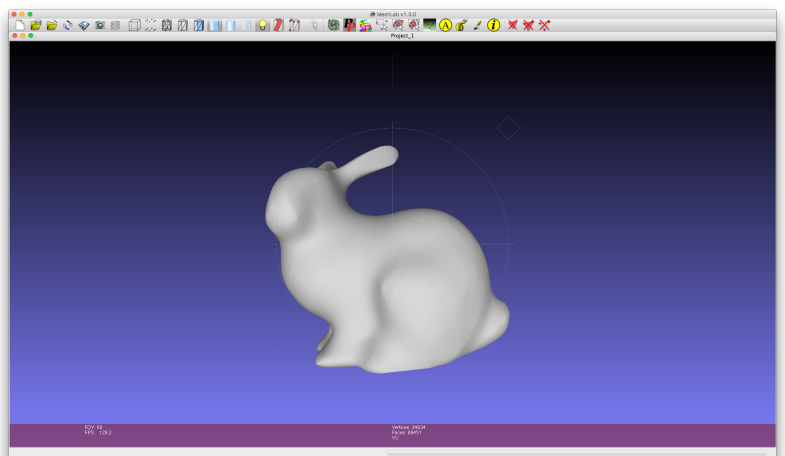
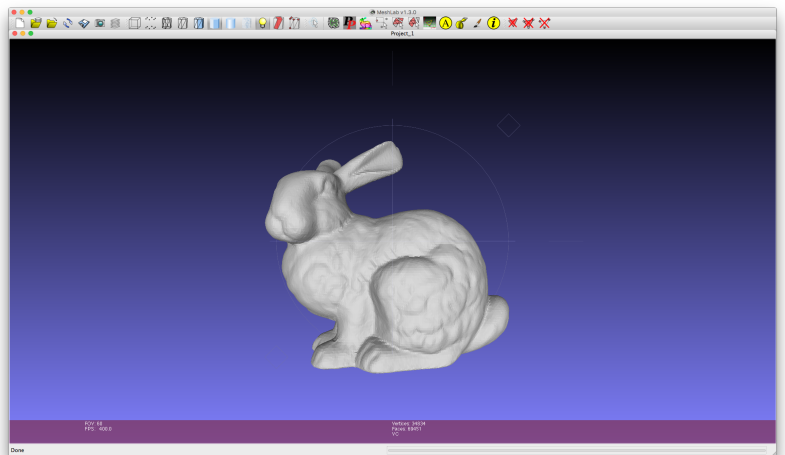
Extra Credit

For 50 additional points, implement [implicit integration](#) (command line option -i eps). eps is a small constant error threshold used by the linear solver. A simple LinearSolver package is in ~adamb/public/LinearSolver. [implicit.cpp](#) is some sample code to use the conjugate gradient solver assuming a Laplacian operator has been defined.

For 15 additional points, use [cotangent weights](#) (command line option -c).

For 15 additional points, implement [global volume preservation](#) (command line option -v).

For 25 additional points, use the biharmonic operator (command line option -b).



For 50 additional points, implement Loop subdivision (command line option -s number_of_subdivision_passes).

What to turn in

Turn in this assignment [electronically](#) by pushing your source code to your proj4 GIT directory by 11:59 PM on the day of the deadline. We will be looking for multiple checkins documenting your development process.

As always, double check that you have submitted **everything** we need to build and run your submission, but **no** generated files (.o's, executables, or images). Be sure to include a Makefile that will build your project when we run 'make', and a `readme.txt` file telling us about your assignment. Do not forget to tell us what (if any) help did you receive from books, web sites or people other than the instructor and TA.

(This page: www.csee.umbc.edu/~adamb/435/proj4.html)