Laboratoire d’Informatique Graphique et Géométrique

Digital 3D Geometry Processing
Exercise 5 – Implicit Smoothing and Minimal Surfaces

Handout date: 24.10.2016

Submission deadline: 03.11.2016, 23:00 h

What to hand in

A .zip compressed file renamed to `Exercisen-GroupMemberNames.zip` where n is the number of the current exercise sheet. It should contain:

- Hand in **only** the files you changed (headers and source). It is up to you to make sure that all files that you have changed are in the zip.
- A `readme.txt` file containing a description on how you solved each exercise (use the same numbers and titles) and the encountered problems.
- Other files that are required by your `readme.txt` file. For example, if you mention some screenshot images in `readme.txt`, these images need to be submitted too.
- Submit your solutions to Moodle before the submission deadline. Late submissions will receive 0 points!

1 Implicit Smoothing

Implement the implicit smoothing algorithm, using the method presented in class, which is based on solving the following equation:

$$(D^{-1} - \lambda M)P^{(t+1)} = D^{-1}P^{(t)}.$$

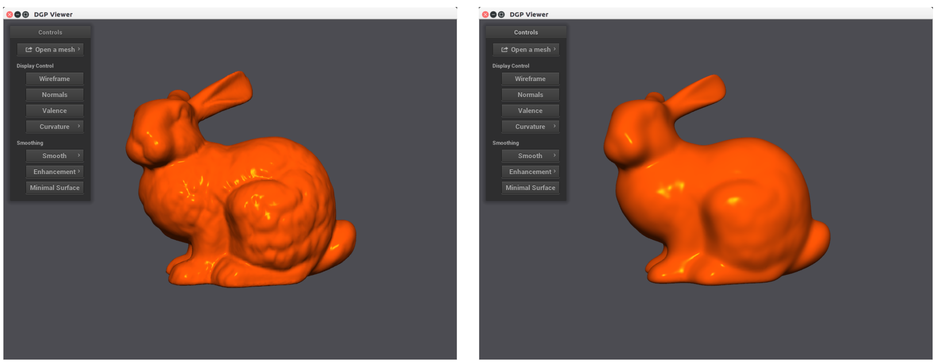

Figure 1: The Stanford bunny and its smoothed version using implicit smoothing.

This method is called via the `Smooth -> Implicit Smoothing` button and the code sits in the `implicit_smoothing(...)` method inside `mesh_processing.h/cpp`. The effect on the bunny mesh is shown in Figure 1.

For curves we saw that under curvature flow any curve goes to a convex shape and then converges to a point (Gage-Hamilton-Grayson theorem, see this webpage `https://en.wikipedia.org/wiki/Curve-shortening_flow#Gage.E2.80.93Hamilton.E2.80.93Grayson_theorem`).
Do you experience an analogous behavior for surfaces? Experiment with various meshes, time steps, and number of iterations. Briefly comment on your observations (no formal proof expected).

2 Minimal Surfaces

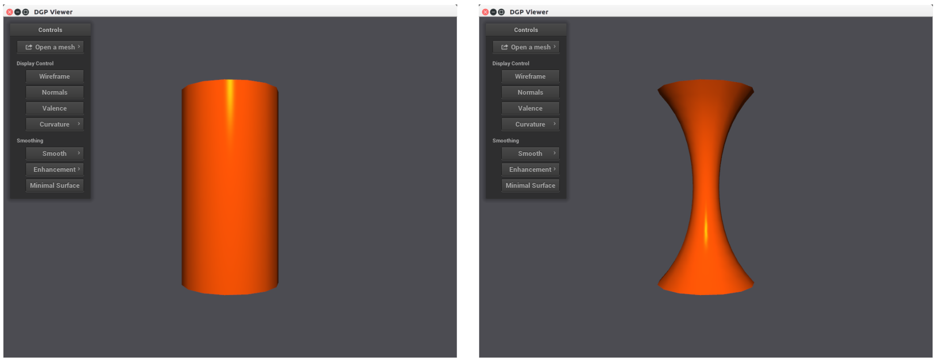

Figure 2: The initial cylinder model and its minimal surface variant when keeping the lower and upper circle boundaries fixed.

In this exercise we will implement a technique for obtaining the minimal surface given an initial mesh and boundary constraints. The minimal surface is the solution to the $LX = 0$ equation. For this exercise the boundary conditions are that the vertices on the boundary of the mesh are kept fixed. This is done by modifying the L matrix accordingly.
The implementation needs to be done in the `minimal_surface()` method inside `mesh_processing.h/cpp` and is called by pressing the `Minimal Surface` button. An example result is shown in Figure 2.

Iterate your method on the three provided cylinders. One of them shows a behavior that is different from the other two. Can you explain what is happening? Is the result consistent with the goal of the minimal surface optimization?

Please include screenshots of your results, accompanied by explanations in the `readme.txt` file.