Digital 3D Geometry Processing
Exercise 10 – Parametrization and Minimal Surfaces

Handout date: 19.11.2018

Submission deadline: 29.11.2018, 23:00 h

What to hand in

A .zip compressed file renamed to `Exercise`n`-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. Indicate what fraction of the total workload each project member contributed.
- 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 Parameterization

In this exercise you will implement the *convex combination maps*. A discrete mapping is called a *convex combination mapping* if it satisfies the linear equations

$$\sum_{v_j \in N_1(v_i)} w_{ij}(\mathbf{u}(v_j) - \mathbf{u}(v_i)) = 0 \tag{1}$$

and has positive weights that sum to one, i.e.

$$w_{ij} > 0 \qquad \wedge \qquad \sum_{v_j \in N_1(v_i)} w_{ij} = 1 \tag{2}$$

1.1 Boundary Initialization

The first step you need to do is to map the boundary vertices to a unit circle in the XY plane. Distribute the boundary vertices on the circle according to the boundary edge lengths. Initialize the texture coordinates for all the interior vertices to a center of the

circle. Complete the function `map_suface_boundary_to_circle()` inside `mesh_processing.cpp`. Store the texture coordinates in a vertex property `v:texture`. To see it in the viewer, enable the `Texture` view mode and click on the `Texture Smooth > Mapping Boundary` button. The outcome for the provided `Max Head` and `Max Face` meshes are shown in Figure1. Note that this only works for disk-topology mesh.

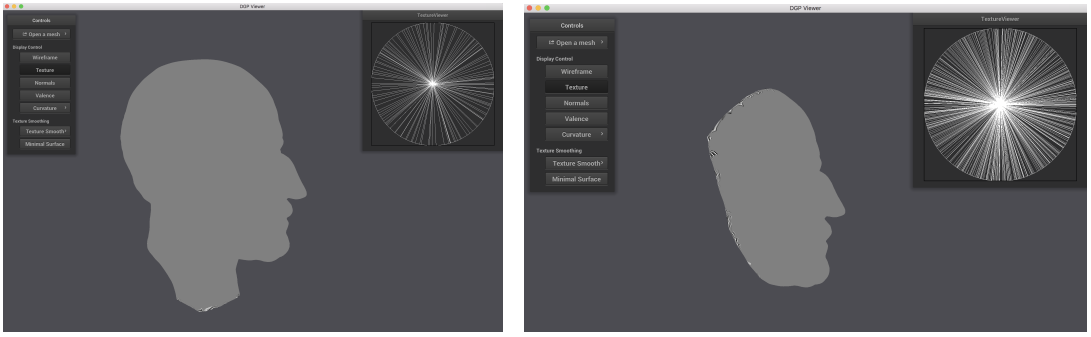

Figure 1: Boundary initialization in the Texture view mode.

1.2 Iterative Solver

One way of solving the linear system for texture mapping is by simply iterating the following for all vertices:

$$\forall v_i \in S \quad : \quad \mathbf{u}(v_i) \leftarrow \frac{1}{\sum_{v_j \in N_1(v_i)} w_{ij}} \sum_{v_j \in N_1(v_i)} w_{ij} \mathbf{u}(v_j).$$

Complete the function `iterative_solve_textures()`. Make sure that the cotan weights you are using satisfy the conditions from equations (2). To see the result of one iteration in the viewer click on the `Texture Smooth > Iterative Solve` button. The result after running 10 iterations of the interactive smoothing is shown in Figure2.

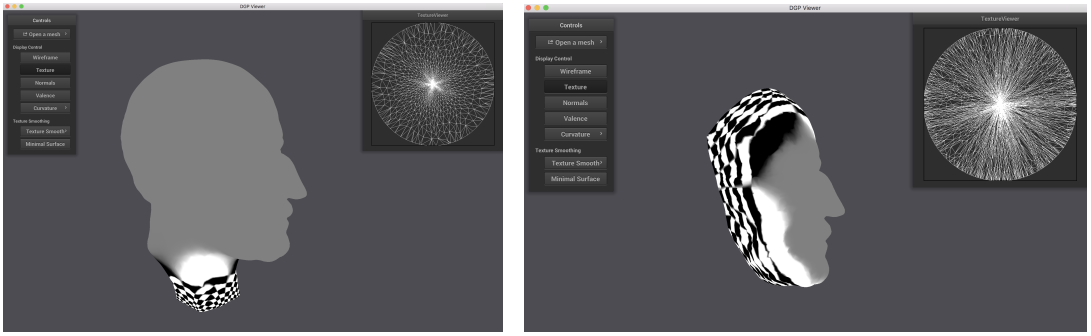

Figure 2: Texture view mode after 10 iterations of iterative solve.

1.3 Direct Solver

Another way of solving the linear system $\mathbf{Lu} = \mathbf{b}$ is to do the direct solve. Form the $\mathbf{L}$ matrix with non-boundary cotan weights and store the boundary conditions in matrix $\mathbf{b}$. Complete the function `direct_solve_textures()`. The result is shown in Figure3.

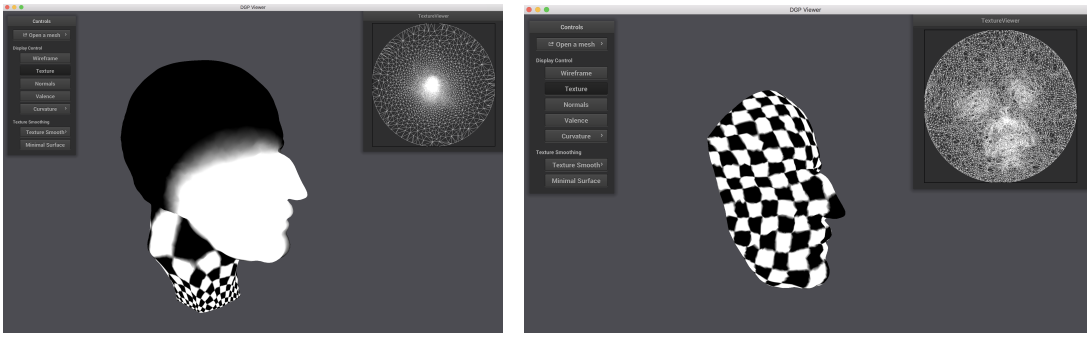

Figure 3: The result of the direct solver.

2 Minimal Surfaces

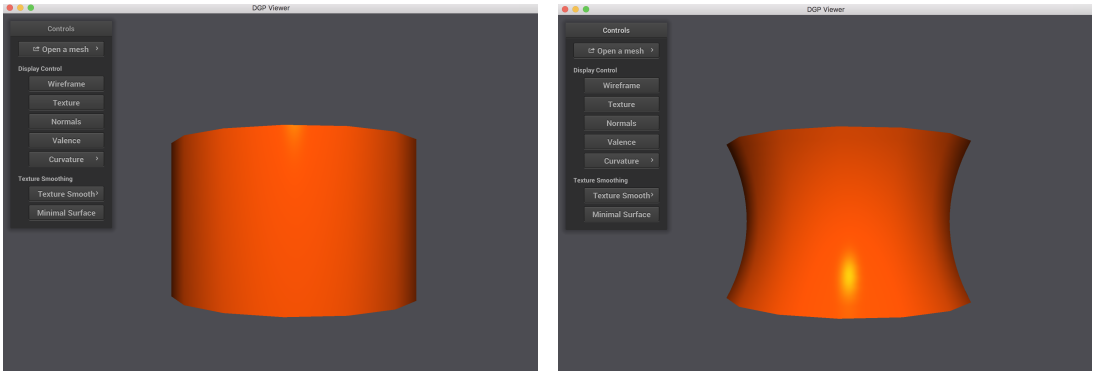

Figure 4: The initial *cylinder1* model and its minimal surface variant when keeping the lower and upper circle boundaries fixed.

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 $\mathbf{LX} = 0$ 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 4.

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?

Does the same effect happen if you replace the cotan Laplacian with the uniform Laplacian? Elaborate your answer.

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