

Tutte Parameterization Report

Features and Implementations

Minimal surface

I implemented a helper class *Laplace_solver* under the file *utils_laplacian_solver.h (.cpp)*, which takes on the following tasks:

- detect boundary.
- set weights, obtain the sparse matrix and target vectors.
- solve the Laplace equation.
- allow us to look up the boundary.

we call it in *node_boundary_mapping.cpp* to detect boundaries, and in *node_min_surf.cpp* to build and solve the Laplacian equation.

We choose the weight type via a bar in the minimal surface node, so far the bar represents integers, which is not very intuitive.

Boundary mapping

we implemented a map to a circle centered at (0.5, 0.5) with radius 1.5.

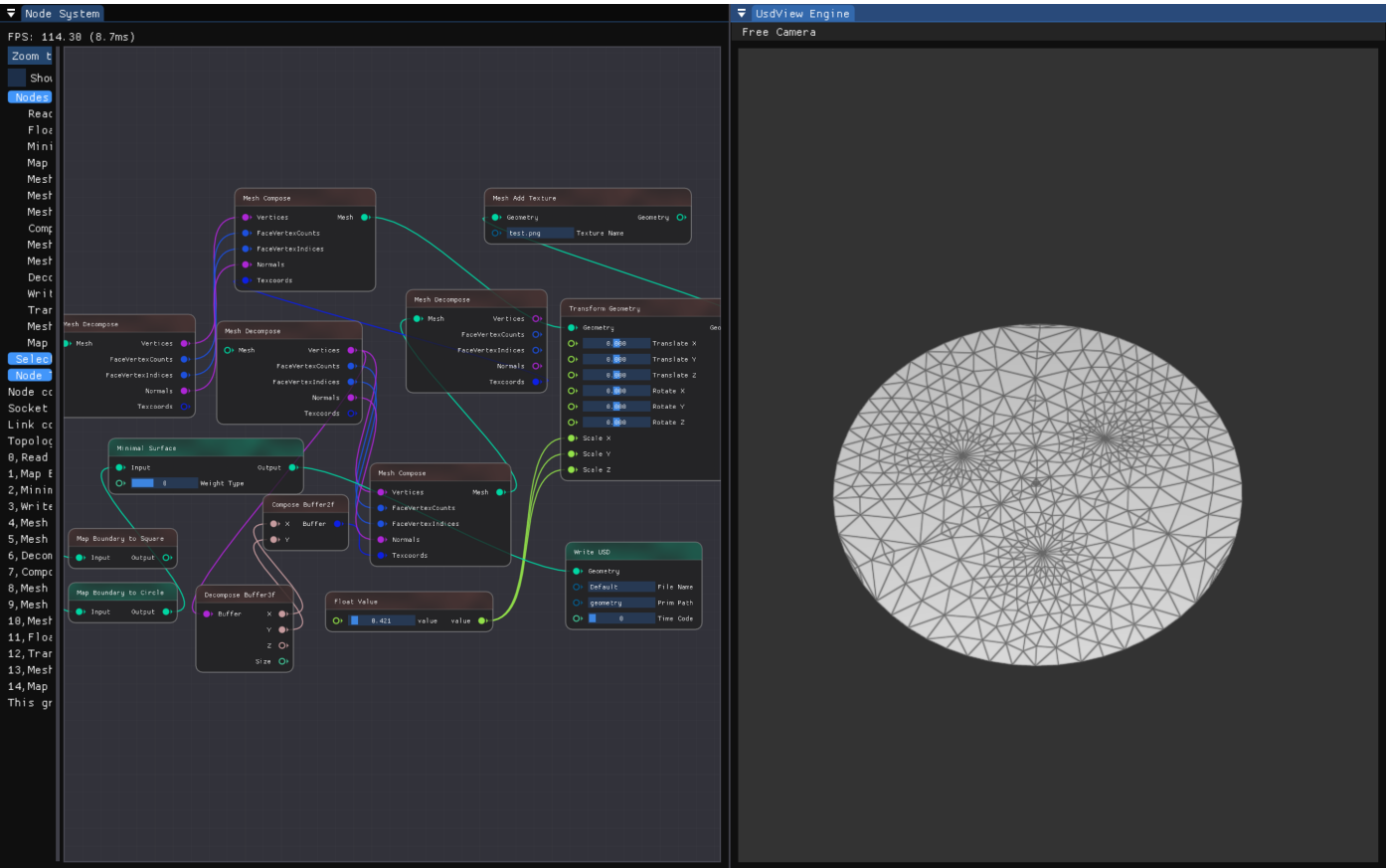
And a map to $[0, 1] \times [0, 1]$. The map of the unit square preserves the four corners: once the mapped boundary vertex is close enough (*when the angular difference is less than half of the angle increment*) to a corner, we forcefully set it to the corner.

Texture mapping

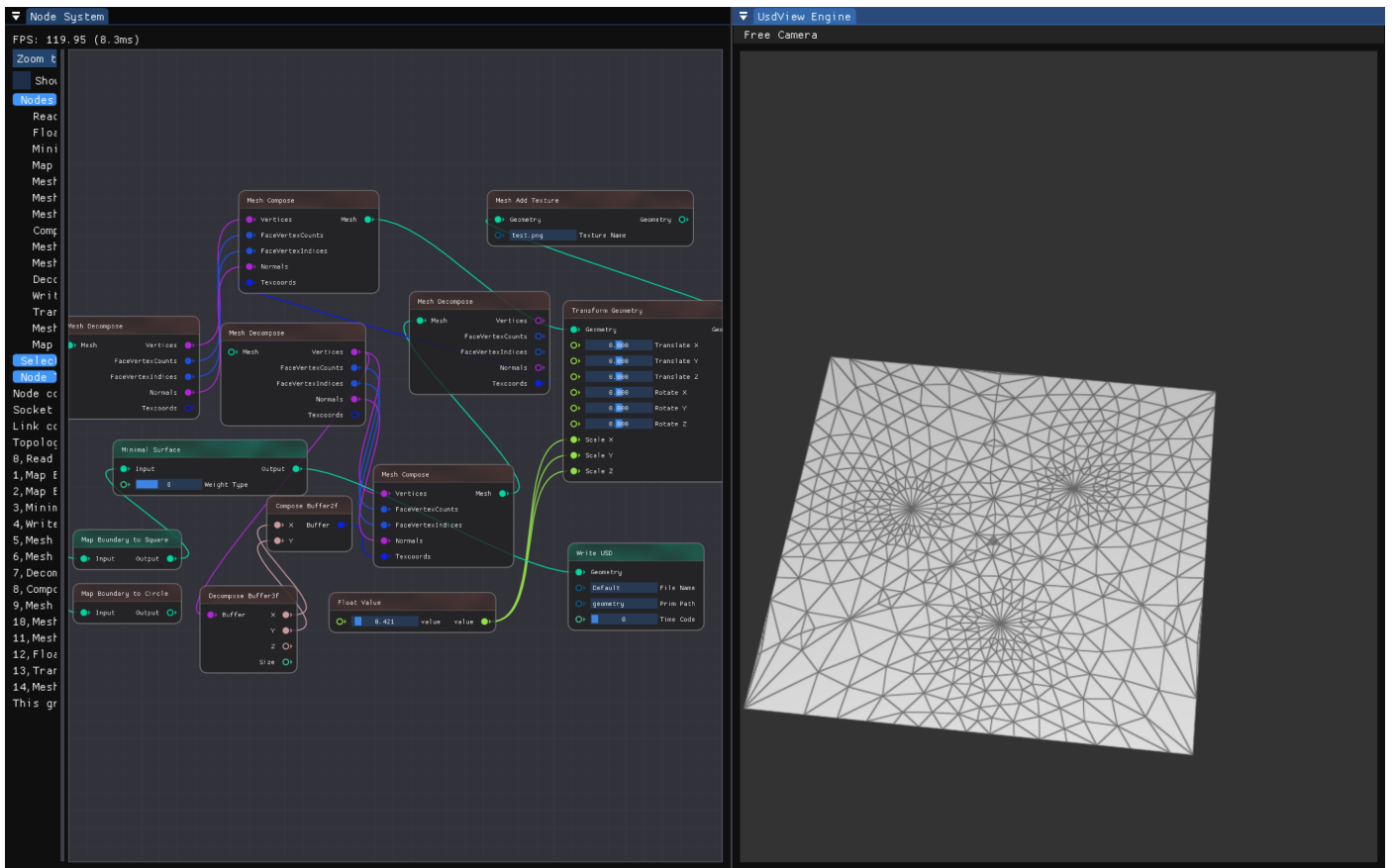
This is achieved via node programming, see the *blueprint.json*. It should be straightforward/standard.

Results

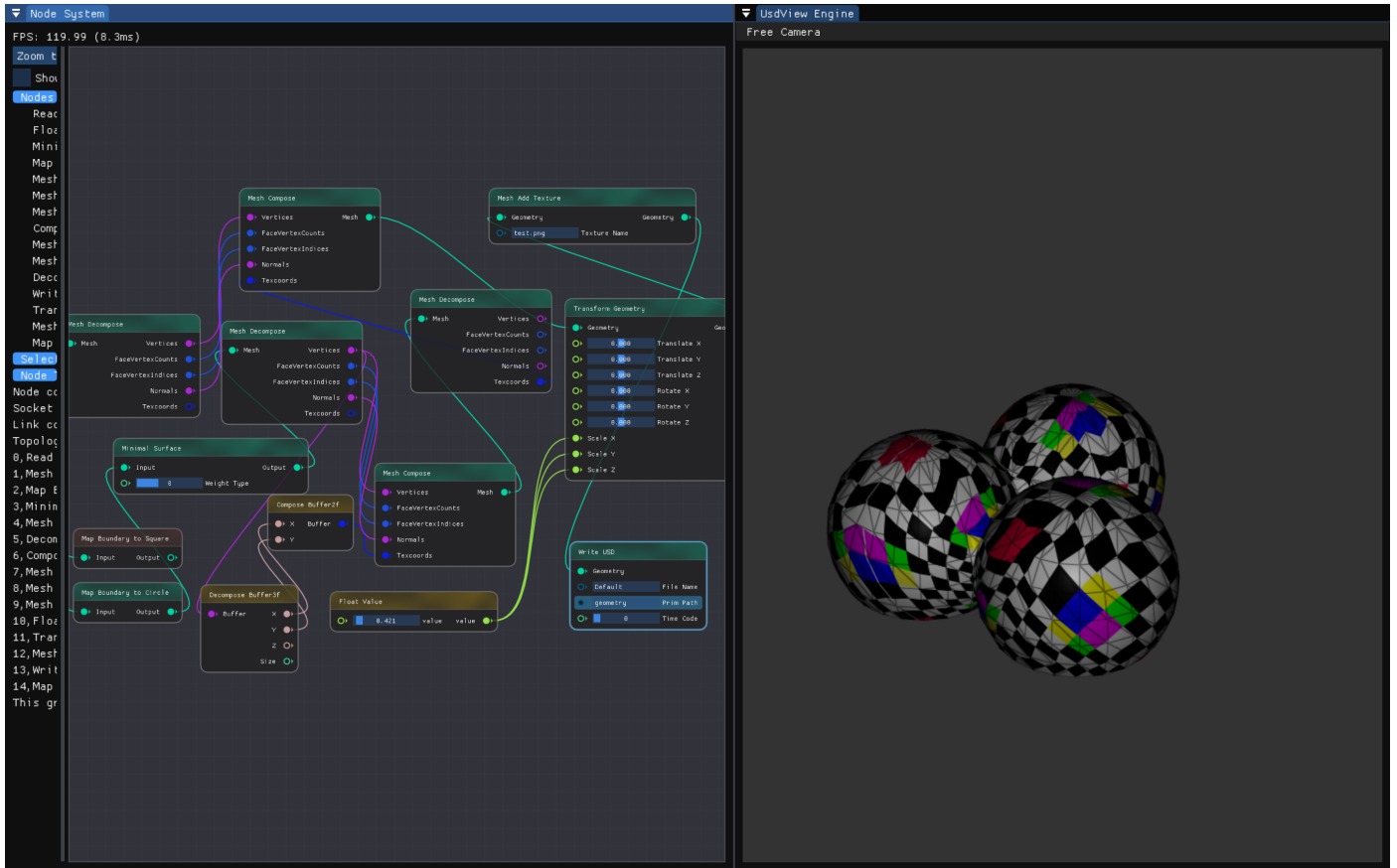
Circle boundary:



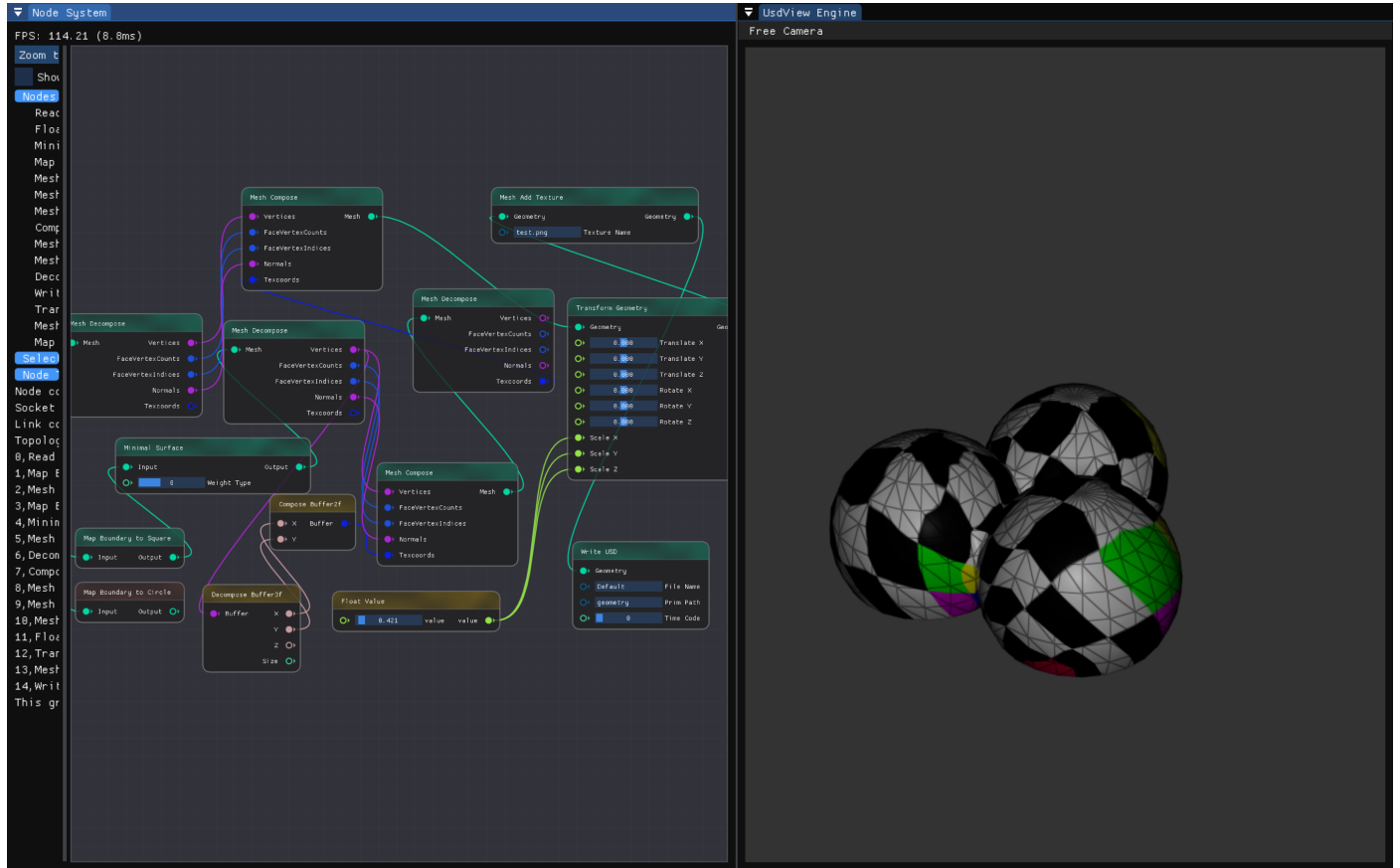
Square boundary:



Circle texture:



Square texture:



Issues and Future Work

- Only uniform weights work now, we need to fix the bug in `Laplace_solver::set_cotangent_weights()`.
- Package weight type selecting as several nodes, instead of passing it in as an integer selection in the minimal surface node.