HW5 Report - ARAP

Features and Implementations

Helper classes

parameterizer.h/cpp: this serves as the parent class without any actual implementation, ARAP, or ASAP class can both inherit this class.
parameterizer_arap.h/.cpp:

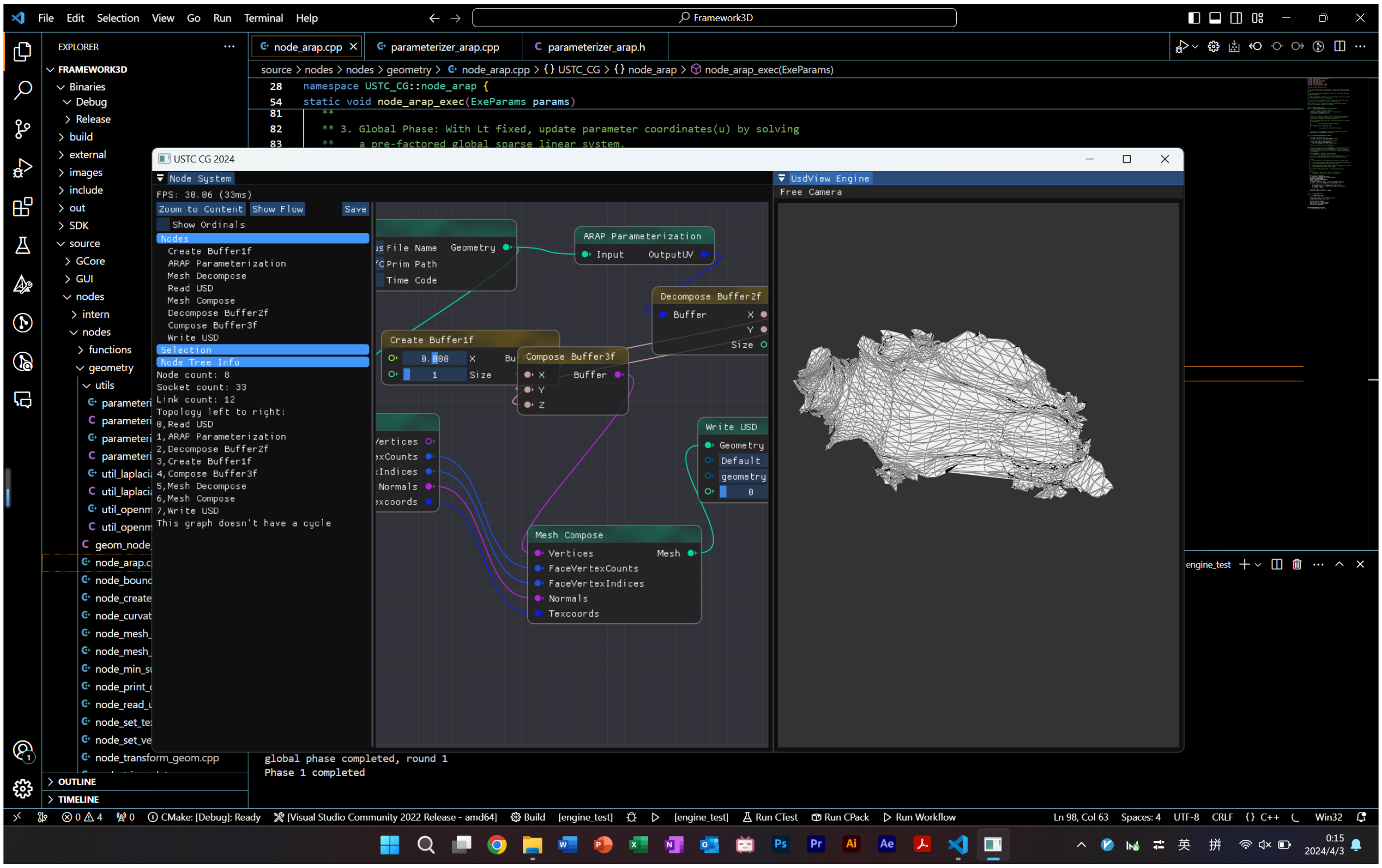
- We have public functions: *void initialize(shared_ptr< OpenMesh >)* to set up the matrix, decompose it with the solver, and execute the first kickoff boundary mapping; *void iterate(int num_iteration)* which does the heavy lifting as the name suggests; then *std::map< int, Vector2d> get_parameterization()*, which simply grants us access to the private field *current_position*.
- We call these three in order in *node_arap.cpp*.

As for private functions: we have

- *global_phase(int phase_count)* which solves the equation, it calls *assemble_targets()* first, then solves the equation with the private field solver and the matrix A initialized in *initialize()*.
- *local_phase()*, it constructs the 2×2 cross-variance matrices $S_t(u)$, which are stored in a private field *std::map< std::pair< int, int>, Matrix2d> L*, i.e. a mapping from half edges to matrices. Then we call *SVD()* to further update L.
- Finally, *iterate()* is just calling *local_phase()* and *global_phase()* in a for loop.

Results

All the following results use only 1 iteration.
Cow:

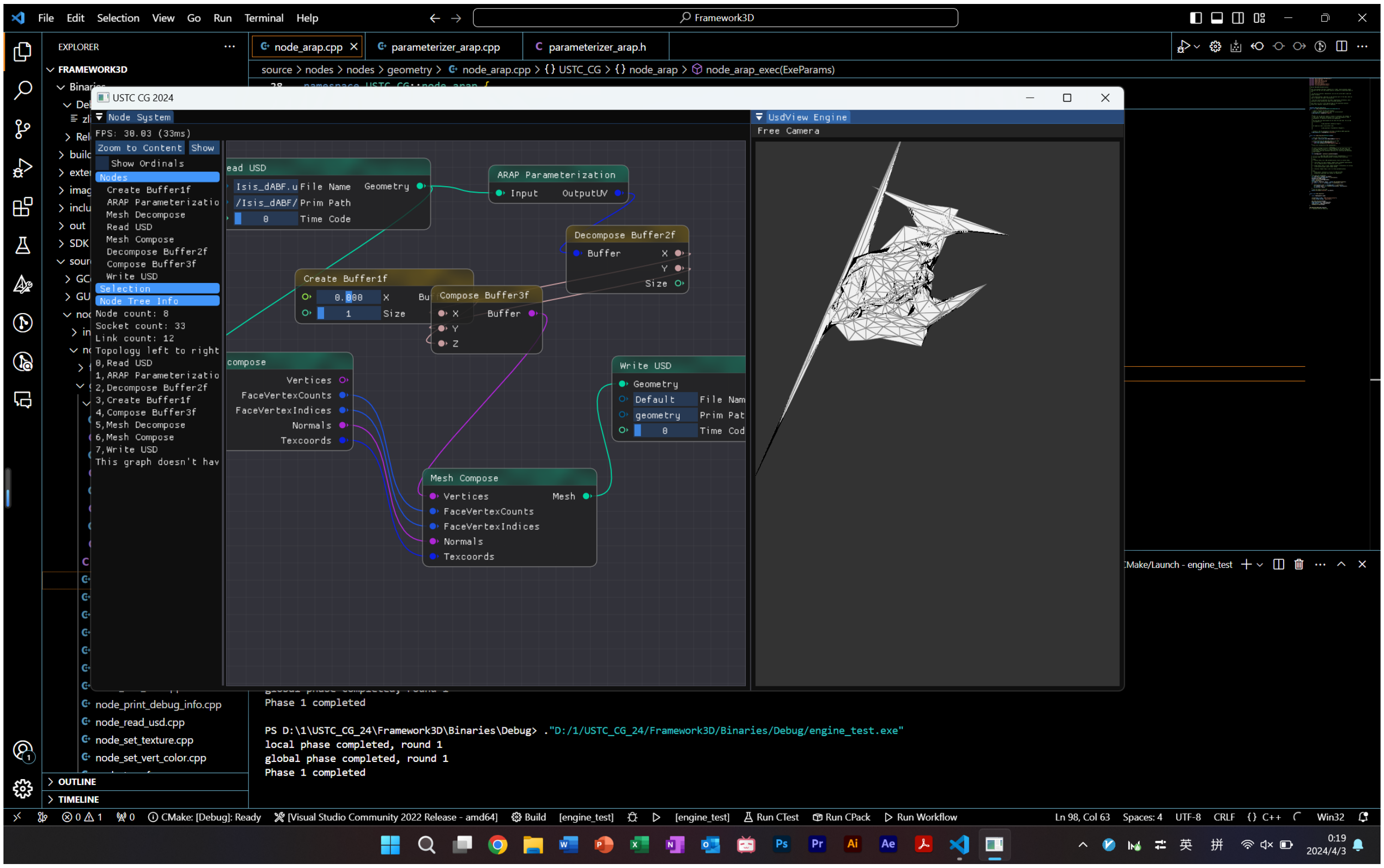


```
namespace USTC_CG::node_arap {
static void node_arap_exec(ExeParams params)
    **
    ** 3. Global Phase: With Lt fixed, update parameter coordinates(u) by solving
    **    a pre-factored global sparse linear system.
```

```
FPS: 30.06 (33ms)
Nodes
  Create Buffer1f
  ARAP Parameterization
  Mesh Decompose
  Read USD
  Mesh Compose
  Decompose Buffer2f
  Compose Buffer3f
  Write USD
Selection
Node Tree Info
Node count: 8
Socket count: 33
Link count: 12
Topology left to right:
0,Read USD
1,ARAP Parameterization
2,Decompose Buffer2f
3,Create Buffer1f
4,Compose Buffer3f
5,Mesh Decompose
6,Mesh Compose
7,Write USD
This graph doesn't have a cycle
```

```
global phase completed, round 1
Phase 1 completed
```

Isis:

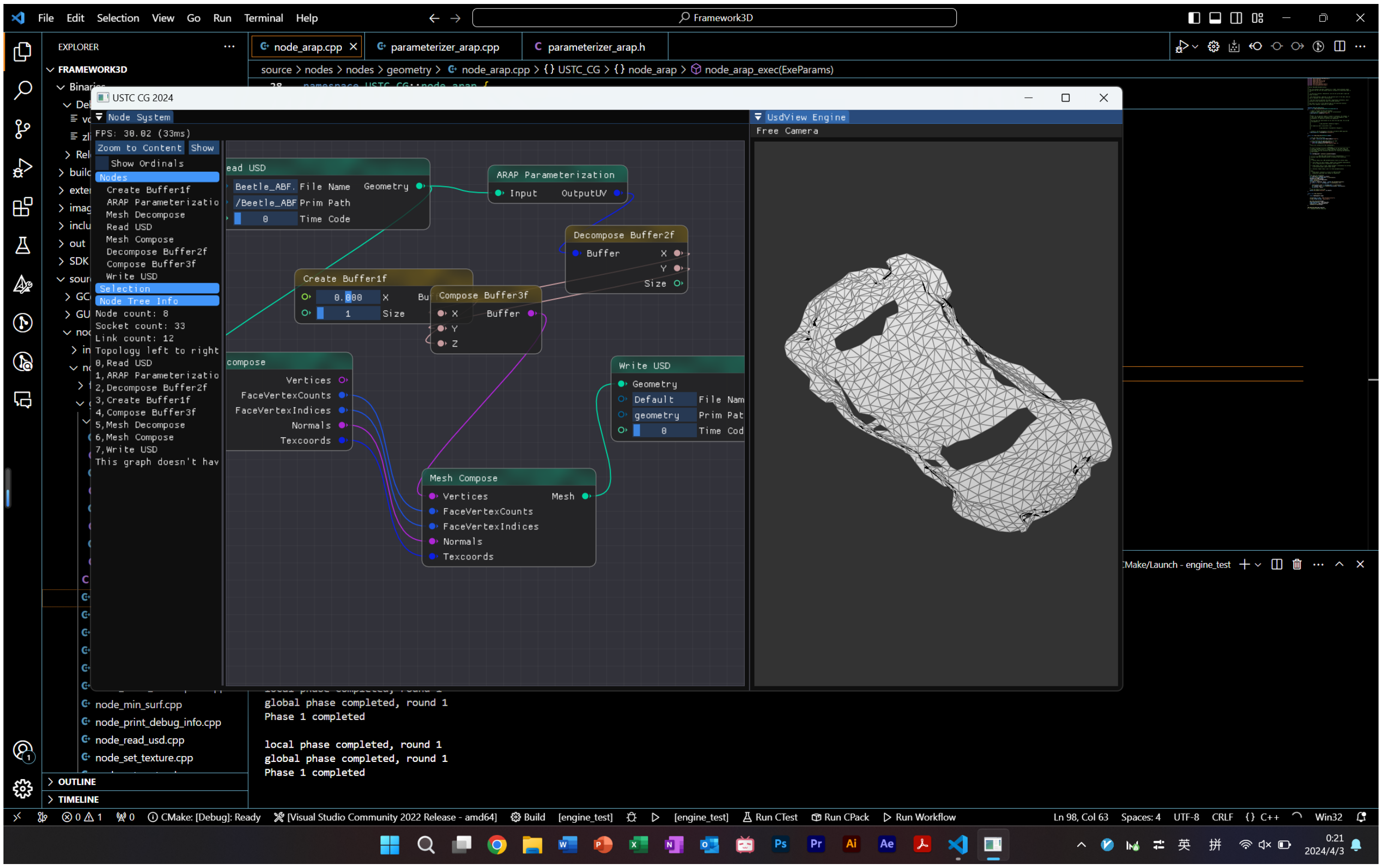


```
FPS: 30.03 (33ms)
Node count: 8
Socket count: 33
Link count: 12
Topology left to right
0,Read USD
1,ARAP Parameterizatio
2,Decompose Buffer2f
3,Create Buffer1f
4,Compose Buffer3f
5,Mesh Decompose
6,Mesh Compose
7,Write USD
This graph doesn't hav
```

```
Phase 1 completed

PS D:\1\USTC_CG_24\Framework3D\Binaries\Debug> ."D:/1/USTC_CG_24/Framework3D/Binaries/Debug/engine_test.exe"
local phase completed, round 1
global phase completed, round 1
Phase 1 completed
```

Beetle:



```
FPS: 30.02 (33ms)
Node count: 8
Socket count: 33
Link count: 12
Topology left to right
0,Read USD
1,ARAP Parameterizatio
2,Decompose Buffer2f
3,Create Buffer1f
4,Compose Buffer3f
5,Mesh Decompose
6,Mesh Compose
7,Write USD
This graph doesn't hav
```

```
global phase completed, round 1
Phase 1 completed

local phase completed, round 1
global phase completed, round 1
Phase 1 completed
```

Issues and Future Work

- I haven't implemented the fix points, now only the first iteration gives a decent result. I might have to check if the mapping of vertices is in order because the mesh appears to be messy.

- do some node programming: make it easier to select the initial boundary mapping method, enable num_iteration selection in the node, etc.