ISL Take-Home Assignment

Overview

Task: find identically-colored connected components in a triangle mesh with Open3D.

You'll implement a function to return a list of identically-colored connected components. An identically-colored connected component consists of spatially connected vertices with the same color. In this question, a connected component is represented by a list of vertex indices.

Examples

Example mesh

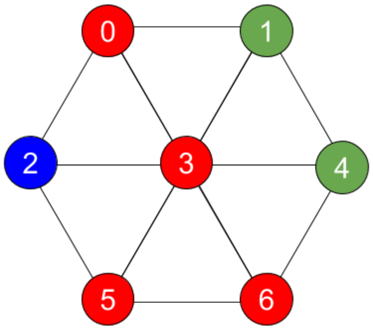

Figure 1: An example triangle mesh with colored vertices.

Given the mesh as show in Figure 1:

- There are 7 vertices: `[0, 1, 2, 3, 4, 5, 6]`,
- There are 6 triangles: `[(0, 2, 3), (0, 3, 1), (1, 3, 4), (2, 5, 3), (3, 5, 6), (3, 6, 4)]`.
- Each vertex has a color: `[red, green, blue, red, green, red, red]`.
- Expected output of `IdenticallyColoredConnectedComponents()`: `[[0, 3, 5, 6], [1, 4], [2]]`.

We then write the results into `result.txt`, where each line in the text file represents one connected component. In each line, vertex indices are separated by space:

```
0 3 5 6
1 4
2
```

Listing 1: `result.txt`

Example C++ invocation (Solution.cpp)

```
#include "Open3D.h"

int main() {
    using namespace open3d;

    // Read triangle mesh "test_mesh.ply"
    t::geometry::TriangleMesh mesh;
    ...

    // Then get the connected components
    auto connected_components = mesh.IdenticallyColoredConnectedComponents();

    // Print connected_components in the specified format
    ...

    return 0;
}
```

Listing 2: Example Solution.cpp

Example Python invocation (solution.py)

```
import open3d as o3d

# Read triangle mesh "test_mesh.ply"
mesh = ...

# Then get connected components
connected_components = mesh.identically_colored_connected_components()

# Print connected_components in the specified format
...
```

Listing 3: Example solution.py

Tasks

You're expected to:

1. Write C++ function:
   `open3d::t::geometry::TriangleMesh::IdenticallyColoredConnectedComponents`.
2. Write Python binding:
   `open3d.t.geometry.TriangleMesh.identically_colored_connected_components`.
3. Write `examples/cpp/Solution.cpp` to read the input mesh `test_mesh.ply`, find identically-colored connected components. Change the build system so that an executable can be build.
4. Write `examples/python/solution.py` to read the input mesh `test_mesh.ply`, find identically-colored connected components and print results.
5. Output the result of task 3 or 4 (their results shall be the same) to `examples/result.txt`.
6. Write C++ and Python unit tests integrated with Open3D's unit test system.
7. Document your code, the algorithm used, how to build and run, and etc.

Reference

Here are some links for your reference:

- Open3D repository: `https://github.com/intel-isl/Open3D`.
- Open3D docs: `http://www.open3d.org/docs`.
- Pybind11: `https://github.com/pybind/pybind11`.

Submission

To submit your code: push the code to a **private** git repository, share the repository with us and ping us via email. Include all the necessary files, including `examples/result.txt`.