Assignment Six: 3D Voronoi Diagram

David Gu

Yau Mathematics Science Center
Tsinghua University
Computer Science Department
Stony Brook University

gu@cs.stonybrook.edu

January 2, 2021

Convex Hull

The input to the 3D voronoi diagram algorithm is a set of 3D points

$$P = \{p_1, p_2, \ldots, p_n\}$$

The output is the 3D Delaunay triangulation of the point set P.

Input

The input points are randomly generated within the unit sphere.

Output

The Delaunay triangulation is represented as a tetrahedral mesh, using Dart data structure to store. The Vornonoi diagram is the dual structure to the Delaunay triangulation.

Vornoi Diagram

Algorithm Pipeline

- Lift each point $p_i = (x_i, y_i, z_i)$ to the parabola,

$$q_i = (x_i, y_i, z_i, w_i), \quad w_i = 1/2(x_i^2 + y_i^2 + z_i^2),$$

- Compute the 4D convex hull of $\{q_1, q_2, \ldots, q_n\}$;
- Project the convex hull to obtain the 3D Delaunay triangulation of $\{p_i\}$;
- Compute the circum-sphere for each tetrahedron;
- Each Delaunay edge is dual to a Voronoi face;
- Each Delaunay vertex is dual to a Voronoi tetrahedron;

Circum Sphere

W. H. Beyer Formula

Given four points on the sphere (x_k, y_k, y_k), $k = 1, 2, 3, 4$, the equation for the sphere is

$$(x-a)^2 + (y-b)^2 + (z-c)^2 = R^2,$$

represented as

$$\begin{vmatrix} x^2+y^2+z^2 & x & y & z & 1 \\ x_1^2+y_1^2+z_1^2 & x_1 & y_1 & z_1 & 1 \\ x_2^2+y_2^2+z_2^2 & x_2 & y_2 & z_2 & 1 \\ x_3^2+y_3^2+z_3^2 & x_1 & y_3 & z_3 & 1 \\ x_4^2+y_4^2+z_4^2 & x_4 & y_4 & z_4 & 1 \end{vmatrix} = 0.$$

Dart Data Structure

EdgeVolumeIterator

Go through all the tetrahedra around one edge,

$$Dart \rightarrow beta(2) \rightarrow beta(3).$$

VertexEdgeIterator

Go through all the edges incident to the vertex.

Example

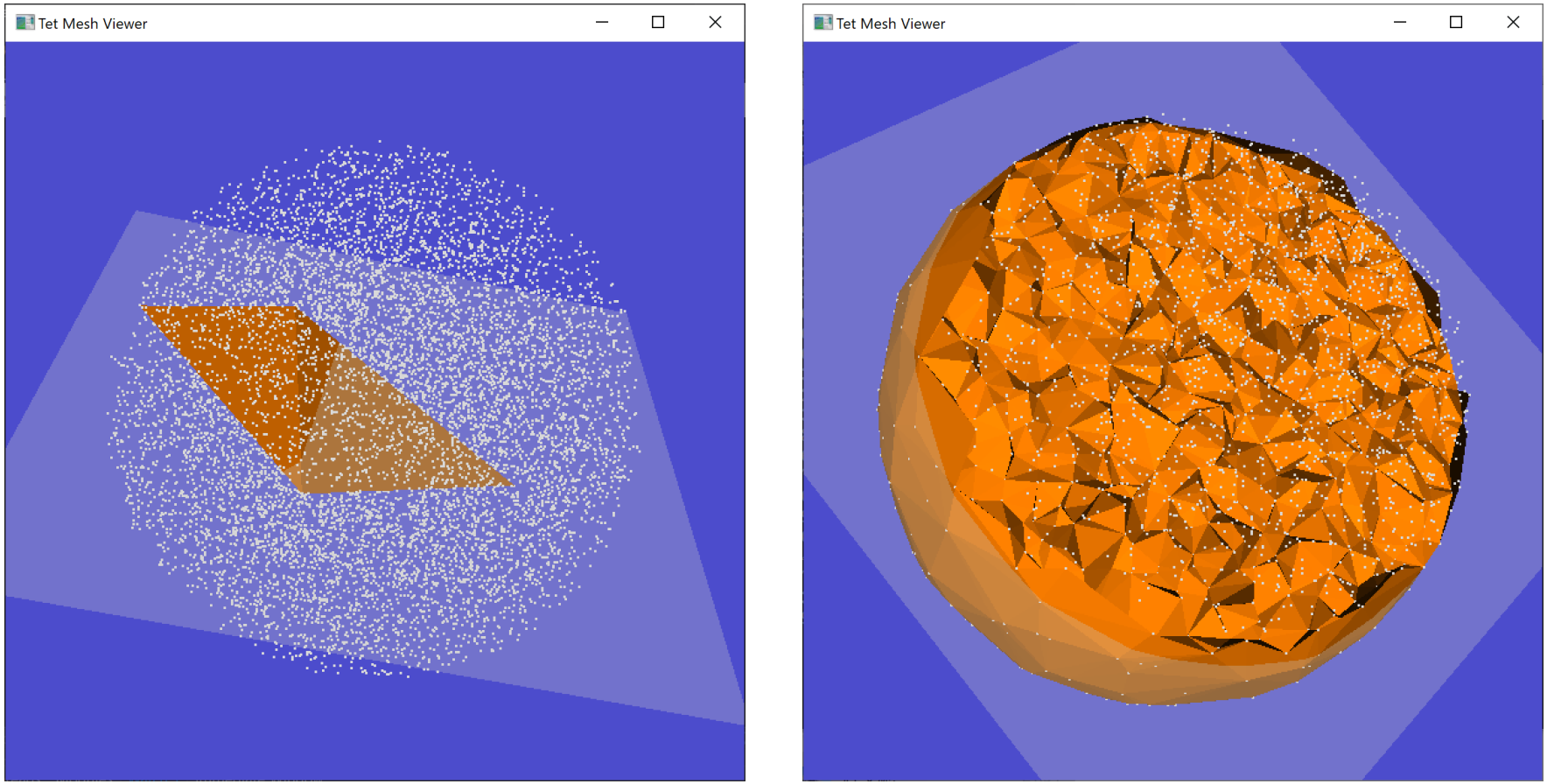

Figure: 3D Delaunay triangulation.

Example

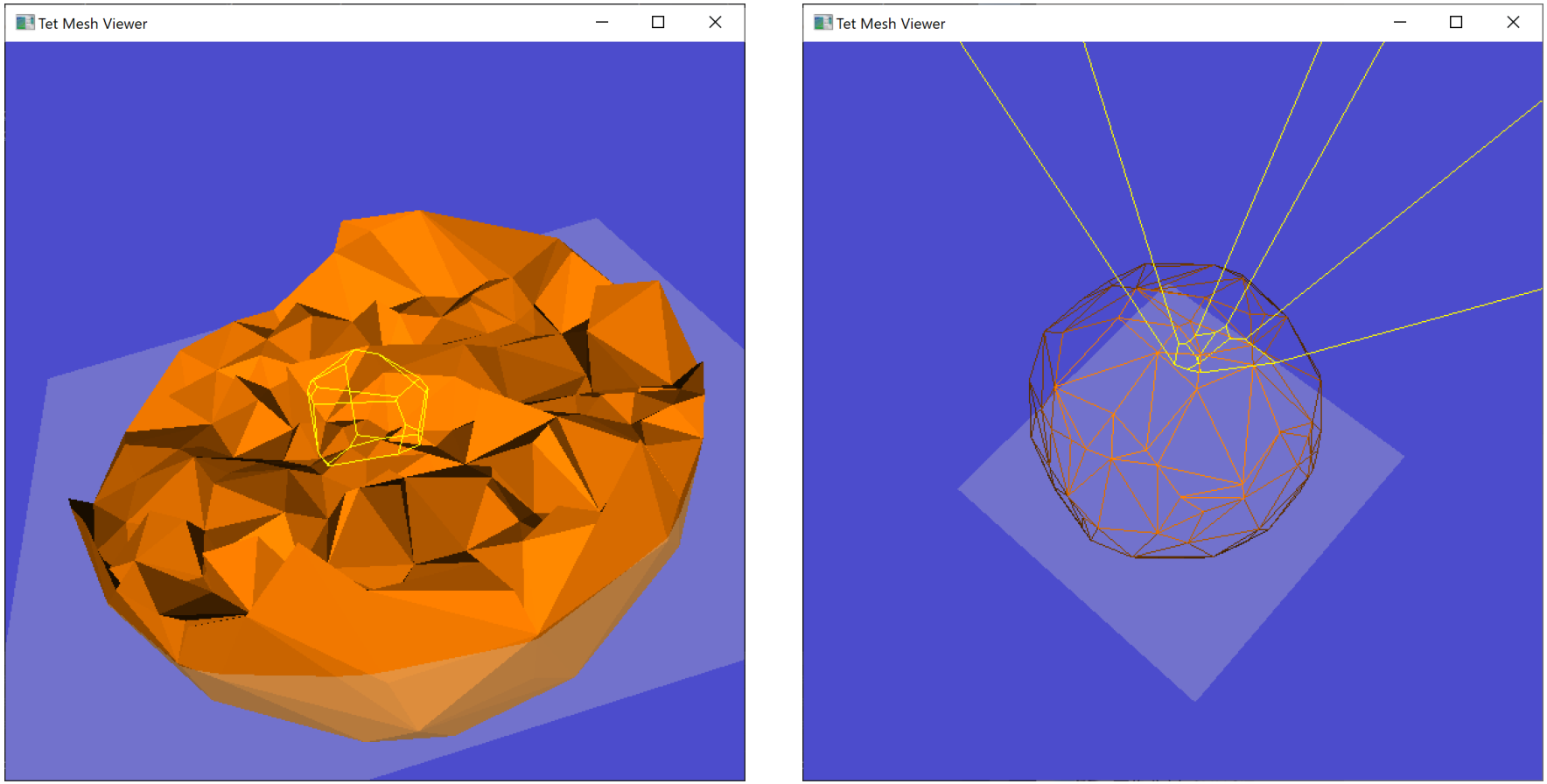

Figure: Voronoi Diagram.

Example

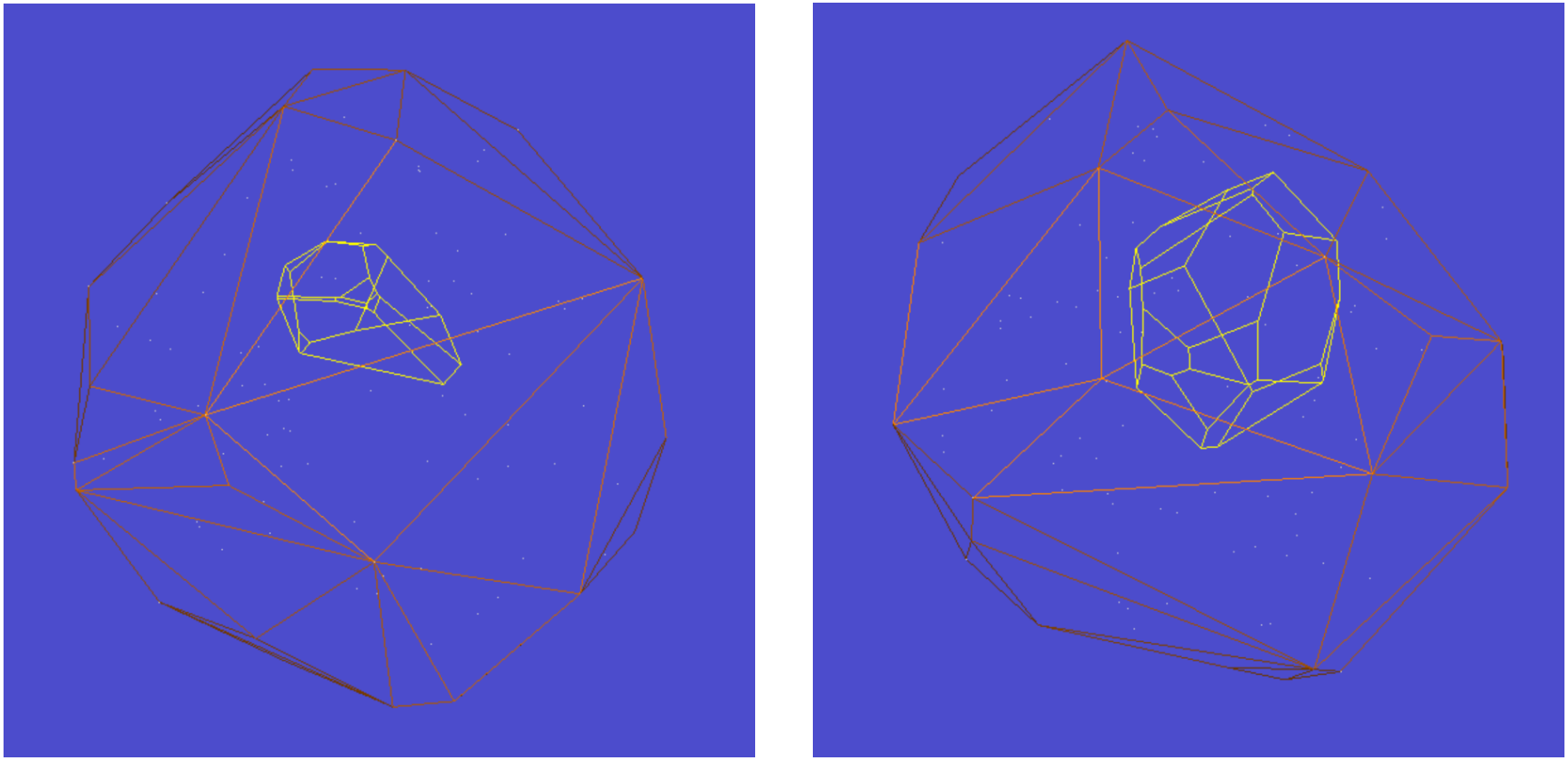

Figure: Voronoi diagram.

Example

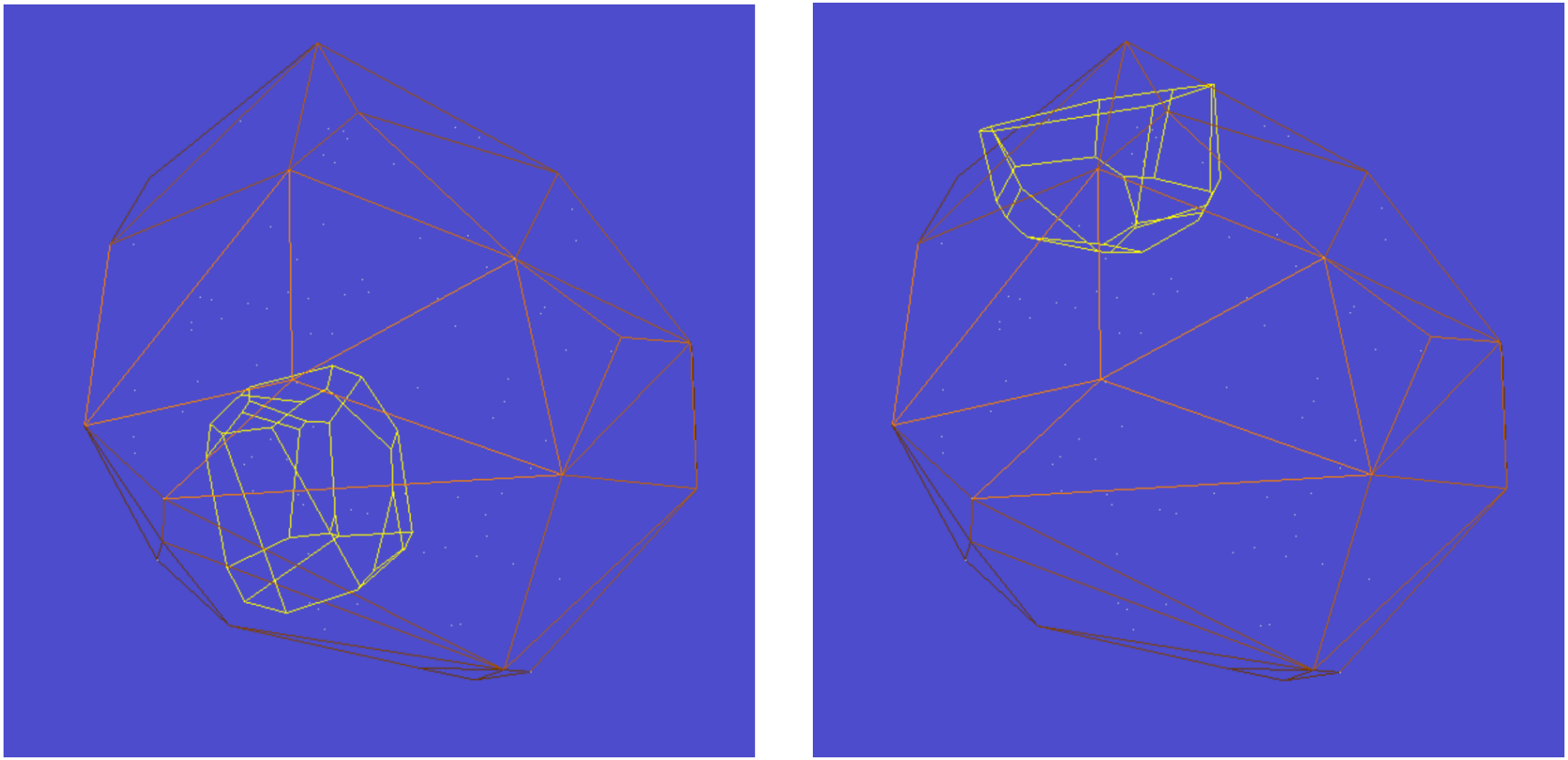

Figure: Vornonoi diagram.

Instruction

Dependencies

1. 'DartLib', a general purpose mesh library based on Dart data structure.
2. 'freeglut', a free-software/open-source alternative to the OpenGL Utility Toolkit (GLUT) library.

Directory Structure

- 3rdparty/DartLib, header files for mesh;
- convex_hull/include, the header files for convex_hull;
- convex_hull/src, the source files for convex_hull;
- CMakeLists.txt, CMake configuration file;

Configuration

Before you start, read README.md carefully, then go three the following procedures, step by step.

1. Install [CMake](https://cmake.org/download/).
2. Download the source code of the C++ framework.
3. Configure and generate the project for Visual Studio.
4. Open the .sln using Visual Studio, and complie the solution.
5. Finish your code in your IDE.
6. Run the executable program.

Configure and generate the project

1. open a command window
2. cd Assignment_6_skeleton
3. mkdir build
4. cd build
5. cmake ..
6. open OTHomework.sln inside the build directory.

Finish your code in your IDE

- You need to modify the file: PowerDiagram3D.cpp, DelaunayTMesh.h;
- search for comments "insert your code"
- Modify functions:
  1. $CPowerDiagram3D :: \_tet\_circumsphere(CDelaunayTMesh :: CVolume * pT, CPointcenter, doubleradius)$
  2. $CDelaunayTMesh :: CDualCellvertex\_dual\_cell(V * pV)$
  3. $CDelaunayTMesh :: CDualFacevertex\_dual\_face(E * pE)$

Finish your code in your IDE

Try your best to improve the efficiency.