1 Introduction

The computation of the Delaunay triangulation of a set of points and using that triangulation to query those triangles of certain points is one of the classical problems of computational geometry. We propose an alternative and possibly efficient way to query those points in 2D using a 3D tetrahedral mesh structure.

2 Related Work

2.1 The Delaunay Hierarchy

Devillers mentions a method to query and update points using a hierarchical structure of Delaunay triangulations. Each level of hierarchy contains a subset of the level below. Algorithm uses a "walk" algorithm to find the edge closest to the query point then starts another traversal in the level below from the found edge until the base triangulation is reached [1]. Computational Geometry Algorithms Library (CGAL) has an implementation of this method built in [2].

2.2 TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator

A paper about the implementation of a C++ program which produces quality Delaunay tetrahedral mesh. We will probably utilize this as we generate our 3D mesh structure [3].

2.3 Fast Tetrahedral Mesh Traversal for Ray Tracing

Aman et al. propose a fast and robust tetrahedral mesh traversal algorithm for rays where they utilize the 3D mesh as an acceleration structure for ray tracing [4]. Memory compression and traversal scheme of this approach will be utilized to traverse our mesh structure instead of a "walk".
3 Proposed Methodology

Our aim is to approach point query problem as an "Ray Tracing" problem rather than a "Triangle walk" problem. Main idea is we will tetrahedralize the given 2D Delaunay triangulation to obtain a "hierarchy" like structure where we have less triangle face on the up-most layer.

The point queries for this structure will be as simple as casting and tracing a vertical ray through the mesh structure from the input origin point. Traversal and compression scheme can be optimized further than Aman et al.’s approach since rays are known to go axis parallel in the -z direction(see figure 1 b) ) .

![Figure 1: a) Example triangulated surface. b)Example query ray traversal.](image)

References


