Walking In A Triangulation

(Notes from the above paper)

by Ashwin Nanjappa

Web: http://www.comp.nus.edu.sg/~ashwinna/

1. Introduction

Point location is a fundamental problem in computational geometry. Given a triangulation $\mathcal{T}$ of n vertices in the plane and a point p, find the triangle of $\mathcal{T}$ containing p.

2. Framework

Let S be a set of n points in $\mathbb{R}^d$, $d = 2, 3$.

Given such a triangulation $\mathcal{T}$ of S, we study strategies to reach a query point p starting from a given starting vertex q of $\mathcal{T}$, walking in $\mathcal{T}$ by using adjacency relations between the simplices of $\mathcal{T}$.

Predicates are simple geometric questions.

The *orientation predicate* is defined over $d+1$ points by the sign of a d dimensional determinant, expressed below for 2 and 3 dimensions:

$$\text{orientation}(\alpha, \beta, \gamma) = \text{sign}(\det\begin{pmatrix} \beta_x - \alpha_x & \gamma_x - \alpha_x \\ \beta_y - \alpha_y & \gamma_y - \alpha_y \end{pmatrix})$$

$$\text{orientation}(\alpha, \beta, \gamma, \delta) = \text{sign}(\det\begin{pmatrix} \beta_x - \alpha_x & \gamma_x - \alpha_x & \delta_x - \alpha_x \\ \beta_y - \alpha_y & \gamma_y - \alpha_y & \delta_y - \alpha_y \\ \beta_z - \alpha_z & \gamma_z - \alpha_z & \delta_z - \alpha_z \end{pmatrix})$$

`neighbour(t through pq)` returns the triangle sharing edge `pq` with the triangle `t`.

3. Straight Walk

3.1 2 Dimensions

Traverse all the triangles of the triangulation $\mathcal{T}$ that are intersected by the line segment originating from a given vertex q of $\mathcal{T}$ and ending at the query point p.

The algorithm first performs an *initialization step* where we turn around on all the triangles incident on q until a triangle intersected by the ray qp is found.

After this we keep going through the edges of triangles that qp cuts through.

4. Orthogonal Walk

Orientation predicate is costly, its cost can be reduced by moving along one of the coordinate axes. This is the basic idea behind the *orthogonal walk*.

5. Visibility And Stochastic Walks

5.1 Description

Visibility Walk (Zigzag Walk): Starting from the triangle incident with q, the first edge e is tested. If it separates t from p, then the next triangle is the neighbour of t through e. The 3 edges of a triangle are tested in this way (if the former fails). When all 3 edges of a triangle fail, we have found the triangle that contains p.

Non-Delaunay triangulations can lead visibility walk into cycles.

Stochastic walk replaces the access to the *first edge* of t by a *random edge* of t. Thus, it prevents cycles.

Remembering stochastic walk adds in some memory to each triangle, it remembers which edge was recently used in a walk.

7. Conclusion And Open Problems

The best to use is the stochastic visibility walk. Next is orthogonal walk.

References

[1]

□