Analysis of Lawson's Oriented Walk in Regular and Random Delaunay Triangulations

Narendher Rao Takkarsu * Ge Xia† Binhai Zhu‡

Point location in Delaunay triangulations is one of the classical problems of Computational Geometry. There has been some related research done in the past, partly due to that Delaunay triangulation is a popular planar subdivision used in real applications. Several algorithms for point location in Delaunay triangulations have been proposed, with or without additional data structures. The latter use no additional data structures [1, 3, 4, 5, 6, 7] and can be classified as sublinear geometric algorithms (for random Delaunay triangulations). Lawson's Oriented Walk (Lawson's Walk for short, which starts with a random edge $e = \overline{ab}$ and in $\triangle abc$, where c, q are on the same side of e, it picks one of the other two edges if it separates $\triangle abc$ and the query point q, and then repeats this process until the triangle containing q is found) [6] and the Jump and Walk [3, 7] are two of the most famous such sublinear geometric algorithms. Jump and Walk has been used in important software packages like CGAL, Triangle and X3D Grid Generation System. However, a formal proof of Lawson's Walk's performance in random Delaunay triangulations is still elusive.

On the other hand, various walking schemes of the straight walk (which starts from a random edge $e = \overline{ab}$ and walks from the mid-point of it to q, triangle by triangle, until the triangle containing q is found), like orthogonal walk, have been practically investigated [4]. In theory, the performance of the straight walk (and with random sampling to have a closer edge to start with, i.e., Jump and Walk) in 2-d random Delaunay triangulations has been well analyzed. The straight walk takes an expected $O(\sqrt{n})$ time in a random Delaunay triangulation of n vertices; and the Jump and Walk algorithm would take an expected $O(n^{1/3})$ time [3, 2]. This motivated us to analyze the performance of Lawson's Walk on 2-d random Delaunay triangulations, which is still open.

We plan our research in two ways. First of all, in some previous failed attempt to prove the performance of Lawson's Walk, an inductive idea was used. So, we first try to investigate whether

*Department of Computer Science, Montana State University, Bozeman, MT 59717, USA. Email: narendherra.takkarsu@cs.montana.edu.

†Department of Computer Science, Lafayette College, Easton, PA 18042. Email: gexia@cs.lafayette.edu.

‡Department of Computer Science, Montana State University, Bozeman, MT 59717, USA. Email: bhz@cs.montana.edu.

this inductive idea can be applied to a regular Delaunay triangulation. We show, mainly through induction, that in such a large grid with n vertices, Lawson's Walk does take $O(\sqrt{n})$ time. On the other hand, the difficulty of analyzing Lawson's Walk in random Delaunay triangulations is that its trajectory could be very different from the corresponding straight walk. Therefore, we try to test empirically how far away Lawson's Walk can deviate from the straight walk. We summarize our theoretical results as follows:

Theorem 1 *Let R be a regular triangular grid with n vertices over a regular triangular domain. Let $s \in R$ be the first triangle visited by Lawson's Walk and let $u \in R$ be the triangle containing the query point q (wlog, assuming q is always a center of some regular triangle in R). Then Lawson's Walk returns a shortest path from the center of s, $c(s)$, to q in the dual hexagonal mesh H.*

Corollary 1 *In a regular triangular mesh R of n vertices within a regular triangular domain, Lawson's Walk takes $O(\sqrt{n})$ time to locate a query point q.*

Proof. We show that that the regular hexagonal mesh H is a planar spanner, i.e., the length of a shortest path between two vertices x, y on H, $\delta(x, y)$, is at most 1.5 times the Euclidean distance $d(x, y)$. As the segment xy crosses $O(\sqrt{n})$ regular triangles in R, the shortest path from x to y on H also crosses $O(\sqrt{n})$ regular triangles in R — one unit length of $\delta(x, y)$ on H implies one extra regular triangle visited in R by Lawson's Walk. □

References

[1] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24:162–166, 1981.

[2] L. Devroye, C. Lemaire and J-M. Moreau. Expected time analysis for Delaunay point location. *Comp. Geom. Theory and Appl.*, 29(2):61–89, 2004.

[3] L. Devroye, E. P. Mücke, and B. Zhu. A note on point location in Delaunay triangulations of random points. *Algorithmica*, Special Issue on Average Case Analysis of Algorithms, 22(4):477-482, 1998.

[4] O. Devillers, S. Pion, and M. Teillaud. Walking in a triangulation. *International J. of Foundations of Computer Science*, 13(2):181–199, 2002.

[5] P. J. Green and R. Sibson. Computing Dirichlet tessellations in the plane. *The Computer Journal*, 21:168–173, 1978.

[6] C. L. Lawson. Software for C^1 surface interpolation. In J.R. Rice, editor, *Mathematical Software III*, pages 161-194. Academic Press, NY, 1977.

[7] E. P. Mücke, I. Saias and B. Zhu. Fast randomized point location without preprocessing in two and three-dimensional Delaunay triangulations. *Comp. Geom. Theory and Appl.*, Special Issue for SoCG'96, 12(1/2):63-83, 1999.