ELSEVIER

SCIENCE DIRECT®

Computational Geometry 29 (2004) 19–22

Computational Geometry
Theory and Applications

www.elsevier.com/locate/comgeo

Expected asymptotically optimal planar point location

John Iacono [1]

Department of Computer and Information Science, Polytechnic University, Brooklyn, NY 11201, USA

Available online 18 May 2004

Communicated by I. Streinu

Abstract

Given a fixed distribution of point location queries among the triangles in a triangulation of the plane, a data structure is presented that achieves, within constant multiplicative factors, the entropy bound on the expected point location query time. The data structure is a simple variation of Kirkpatrick's classic planar point location structure [D.G. Kirkpatrick, SIAM J. Comput. 12 (1) (1983) 28–35], and has linear construction costs and space requirements.

Keywords: Point location; Distribution-sensitive data structures

1. Introduction

Point location is a fundamental problem in computational geometry. In a point location problem, a subdivision of space is provided, and a data structure is constructed such that, for each query point, a pointer to the triangle in the subdivision in which the query point is in is returned. This problem has been well studied, and different restrictions on the dimension and nature of the subdivision have yielded numerous data structures. In two dimensions, data structures have been created (e.g. [8] using [5] for preprocessing) which have attained the best possible asymptotic query time of $O(\log n)$, and construction time of $O(n)$, for a planar polygonal subdivision, where n is the size of the subdivision.

Given a triangulation of the plane, T, into n triangles $t_1, \ldots, t_n$, if the probability of accessing triangle t_i on every point location query is $p(t_i)$ then the lower bound for the expected cost per operation is given by the entropy: $\sum_{i=1}^{n} p(t_i) \log 1/p(t_i)$. Thus, for non-uniform access distributions, expected $o(\log n)$ point location queries are possible.

E-mail address: jiacono@poly.edu (J. Iacono).

[1] Part of this work was done while the author was at Rutgers, The State University of New Jersey–New Brunswick. Research partially supported by NSF grant CCR-9732689.

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2004.03.010

The main result of this paper is a data structure that achieves the entropy bound, within a constant multiplicative factor, when the access probabilities are known. Our data structure, which is a variant of Kirkpatrick's [8], can be constructed in $O(n)$ time and space, and has a worst-case cost of $O(\log n)$ per point location query.

2. Previous results

Goodrich, Orletsky and Ramaiyer [6] provided a data structure for the point location problem that achieves the same asymptotic runtime as our structure over a point location query sequence, with one main advantage. Their structure is online, and thus does not require initialization with distributional information. As their structure is based on splay trees, the runtimes of individual point location queries are amortized and may take linear time. The disadvantage of the structure is that the runtime is relative not to the user provided subdivision of the plane, but rather to one constructed by passing a vertical line through every point in the original subdivision. This restriction can cause their structure to perform asymptotically worse than ours over certain classes of subdivisions and point location query distributions.

Three other optimal planar point location data structures were developed by Arya, Malamatos and Mount, which improved upon the earlier work of Arya et al. [1]. The first two are complex structures that achieve the entropy bound with a constant of 1 plus lower order terms. They differ in the amount of space used. The result of [2] used $O(n \log n)$ space, which is then improved in [3] to $O(n \log^* n)$. Their third structure, [4], achieves the entropy bound within constant factors through a modification of the randomized point location methods of Mulmuley [9] and Seidel [10]. This randomized structure is simpler to implement.

In comparison, our result achieves the entropy bound for point location in a triangulation, within constant multiplicative factors, using linear space and preprocessing time, without any unnatural assumptions or restrictions. While our result is the only deterministic one with linear time construction, the main advantage of our result is its simplicity. Given code for Kirkpatrick's method, our structure could be implemented with only minor changes. However, the constants used by our structure, as presented, differ from the best results of the previous paragraph by almost three orders of magnitude. While we have made no effort to minimize the constants, such minimization will not bring the constants down to a reasonable level.

We would like to note that [3,4], and the preliminary version of this result, [7], were produced independently and appeared simultaneously at the same conference.

3. Description of the structure

The data structure is a variant of Kirkpatrick's planar point location structure [8]. In order to create the data structure, a triangulation T of the plane into triangles $t_1, t_2, \ldots, t_n$, and an associated set of values $x_1, x_2, \ldots, x_n$, are provided, where x_i denotes the probability that a query point lies in triangle t_i. We assume that the triangulation has been augmented so that it contains only three exterior vertices. The data structure consists of a series of triangulations $T_0, T_1, \ldots, T_m$. Each triangle in a triangulation T_k contains links to those triangles in T_{k+1} and T_{k-1} that intersect it. A triangle is said to be terminal if it is found in the original triangulation T. Every triangle has an associated weight. The sum of weights in each triangulation is 1.

The final triangulation T_m is T, all other triangulations T_k are derived from T_{k+1} by removing an independent set of internal vertices, and arbitrarily retriangulating the resultant non-triangular polygons. The weight of the triangle t_i in T_m is x_i, for all i. The weights of newly formed nonterminal triangles is an arbitrary redistribution of the weights of the triangles incident to the vertex whose removal caused the creation of these triangles. The weight of a triangle in T_k which remains from T_{k+1} remains the same. The key to making this data structure work is carefully picking the right independent set of vertices for removal.

Suppose T_k has n_k triangles and v_k vertices. In a triangulation it follows from Euler's formula that the average degree of a vertex is at most 6. Thus, at most $v_k/4$ vertices have degree higher then 24. Since the average weight of a triangle is $1/n_k$, at most $n_k/24$ of the triangles have weights greater than $24/n_k$. As each triangle is incident to 3 vertices, at most

$$\frac{3n_k}{24} = \frac{n_k}{8} = \frac{2v_k - 4}{8} \leqslant \frac{v_k}{4}$$

vertices have a triangle incident whose weight is at least $24n_k$. Therefore, at least $v_k/2$ vertices have degree at most 24 and do not have a triangle incident whose weight is more than $24/n_k$. We then may, by simply using a greedy $O(n_k)$ algorithm, pick an independent set of size $(n_k - 3)/50$ of independent internal vertices for removal with both of these properties. The -3 term is caused by the fact that the removal of the external three vertices is forbidden. The removal of this set from T_{k+1}, and the retriangulation of the resultant non-triangular triangles, is the method by which T_k is derived from T_{k+1}. Since each of the removed vertices has a degree of at most 24, the triangulation of the polygon formed by the removal of a vertex can be done in constant time, and each newly formed triangle in T_k will intersect at most 24 triangles in T_{k+1}. The total time to create T_k from T_{k+1} is $O(n_{k+1})$. The process of removing vertices to create new triangulations is stopped when all of the interior vertices are removed, as it is always possible to remove at least one interior vertex with degree at most 24 that does not have a triangle incident whose weight is more than $24/n_k$. Thus $n_0 = 3$.

Point location on a query point p is performed by locating the triangle in each triangulation $T_0, T_1, \ldots$ that p lies in, until p is found to be in a terminal triangle of some triangulation T_j. For each step in this process, if the triangle that p lies in t_k is known to be τ, the triangle that p lies in T_{k+1} may be determined by looking at the at most 24 triangles in T_{k+1} that intersect with, and thus are linked to, τ. Thus point location on a point in a triangle that is terminal in triangulation T_j takes time $O(j)$.

4. Analysis

Theorem 1. *For a triangulation T of the plane into n triangles, $t_1, t_2, \ldots, t_n$, and a set of n positive values $x_1, x_2, \ldots, x_n$, $\sum_{j=1}^{n} x_j = 1$, our structure can be built in time and space $O(n)$ and can answer a point location query for a point in t_i in deterministic time $O(\min(\log n, \log(1/x_i)))$.*

Proof. Since $(49/50)(v_{k+1} - 3) \leqslant v_k - 3$, and $v_0 = 3$, it can be seen that $v_k - 3 \geqslant (50/49)^k$, and thus $m = O(\log n)$. Since only vertices of weight at most $24/n_k$ are chosen for the independent set in processing T_k, t_i is guaranteed to remain a terminal triangle in all triangulations $T_j, \ldots, T_m$ where $x_i > 24/n_j = 24/(2v_j - 4)$. Thus t_i will be terminal in triangulations where $v_j \geqslant 12/x_i + 2$. Since $(50/49)^j + 3 = 12/x_i + 2$ when $j = \log_{50/49}(12/x_i - 1)$, setting $j = \log_{50/49}(12/x_i - 1)$ guarantees

that $v_j \geqslant 12/x_i + 2 = (50/49)^j + 3$ and thus t_i is terminal is all triangulations $T_{\lceil \log_{50/49}(12/x_i - 1)\rceil}, \ldots, T_m$. Therefore the time to perform a point location query on a point in t_i is $O(\log \min(\log n, 1/x_j))$. □

Because each level can be constructed in linear time, and the size of each level is a geometric series from 3 to n, the total construction cost is $O(n)$. For the same reason, the space used is $O(n)$.

The main result is an immediate consequence of this theorem and can be stated as a corollary:

Corollary 1. *If a sequence of point location queries on a triangulation are drawn independently from a known fixed distribution, our data structure, which uses linear space and preprocessing time, is expected to execute the sequence of point location queries optimally within constant factors.*

References

[1] S. Arya, S. Cheng, D. Mount, H. Ramesh, Efficient expected-case algorithms for planar point location, in: Proc. 7th Scandinavian Workshop on Algorithm Theory, Lecture Notes in Comput. Sci., vol. 1851, 2000, pp. 353–366.
[2] S. Arya, T. Malamatos, D.M. Mount, Nearly optimal expected-case planar point location, in: Foundations of Computer Science, 2000, pp. 208–218.
[3] S. Arya, T. Malamatos, D.M. Mount, Entropy-preserving cuttings and space efficient planar point location, in: Symposium on Discrete Algorithms, 2001, pp. 256–261.
[4] S. Arya, T. Malamatos, D.M. Mount, A simple entropy-based algorithm for planar point location, in: Symposium on Discrete Algorithms, 2001, pp. 262–268.
[5] B. Ghazelle, Triangulating a simple polygon in linear time, Discrete Comput. Geom. 6 (5) (1991) 485–524.
[6] M.T. Goodrich, M. Orletsky, K. Ramaiyer, Methods for achieving fast query times in point location data structures, in: Symposium on Discrete Algorithms, 1997, pp. 767–776.
[7] J. Iacono, Optimal planar point location, in: Symposium on Discrete Algorithms, 2001, pp. 340–341.
[8] D.G. Kirkpatrick, Optimum search in planar subdivisions, SIAM J. Comput. 12 (l) (1983) 28–35.
[9] K. Mulmuley, A fast planar partition algorithm, SIAM J. Comput. 6 (1977) 235–239.
[10] R. Seidel, A simple and fast incremental algorithm for computing trapezoidal decompositions and for triangulating polygons, Computational Geometry 1 (1991) 51–64.