References

[1] Y. F. Xu. Minimum weight triangulation of a planar point set. Ph. D. thesis. Institute of Applied Mathematics, Academia Sinica, 1992.

[2] A. Borodin, N. Linial and M. Sakes, An optimal on-line algorithm for metrical task systems, Proc. 19th STOC, 1987, 373-382.

[3] S. Irani, Coloring inductive graphs on-line, Proc. 31st FOCS, 1990.

[4] A. A. Melkman, On-line construction of the convex hull of a simple polyline, Inform. Proc. Lett. Vol. 25, 1987, 11-12.

[5] F. Preparata and M. I. Shamos, Computational Geometry-An Introductibin, Springer-Verlag, 1985.

求平面点集独立线段的一个联机算法

徐寅峰 （陈丽华）
（西安交通大学管理学院） （北京科技大学）

提 要

令 S 为一个有限平面点集合，线段 L(p,q)，p，q∈S 称为 S 的一个独立线段当且仅当不存在两个端点在 S 中的线段与 L(p,q)相交。本文给出了时间复杂度为 $O(n^2\log n)$ 和修正时间复杂度为 $O(n\log n)$ 的联机算法。这一算法与已知的脱机算法具有相同的时间复杂度。

第12卷 第4期
1995年12月

工 程 数 学 学 报
JOURNAL OF ENGINEERING MATHEMATICS

Vol. 12 No. 4
Dec. 1995

An On-line Algorithm for Finding Stable Line Segments of a Planar Point Set*

Xu Yinfeng
(The School of Management, Xi' an jiaotong University, Xi' an710049)

Chen Lihua
(Beijing University of Science and Technology, Beijing100083)

Abstract

Let S be a set of finite planar points, a line segment $L(p,q)$ with $p,q \in S$ is called a stable line segment of S, if there is no line segment with two endpoints in S intersects L(p,q). In this paper, an on—line algorithm with $O(n^2\log n)$ time and $O(n\log n)$ update time is also given.

Keywords on-line algorithm; stable line segment, convex hull
Classification 68Q25, 52C25

1 Introduction

For S, a set of n points in a plane, a line segment denoted by $L(p,q)$ with endpoints p,q is called a stable line segment of S, if there is no line segment with two endpoints in S intersects $L(p,q)$. Another geometrical interpretation of a stable line segment $L(p,q)$ is the point p and point q always see each other, no matter whtat line segments with endpoints in S are added. In this case, the pair of points $\{p,q\}$ can be also thought of as a strongly visibility pair of points. Clearly, the set of stable line segments of S is very important in the study of some visible problems with S and as shown in [1], the set of all stable line segments of S also plays an important role in the design of algorithms that produce minimum weight triangulations of S. For the convenience of discussions, we assume that S is a simple point set(that is , no three points in S are collinear). The set of all stable line segments of S is denoted by SL(S). In this paper, an on-line algorithm with $O(n^2\log n)$ time and $O(n\log n)$ update time is given.

* 本文1993年8月27日收到。

2 An On-line Algorithm

The off-line algorithm examined in [1] requires all of the data to be present before any processing begins. In many geometric applications, particularly those that run in real-time, this condition connot be met and some computations must be done as the points are being received. In general, an algorithm that cannot look ahead at its input is refered to as on-line, while one that operates on all the data collectively is termed off-line. A general feature of on—line algorithms is that no bound is placed on the update time, or equivalently, a new item(point) is input on request as soon as the update relative to the previously input item has been completed. We shall refer to the time interval between two consecutive inputs as the interarrival delay. A more demanding case of on-line applicatins occurs whtn the interarrival delay is outside the control of the algorithm. In other words, inputs are not acquired on request by the algorithm, but occur with their own independent timing; we shall assume, however, that they be evenly spaced in time. For this situation, the update must be completed in time no greater than the constant interarrival delay. Algorithms for this mode of operation are appropriately called in real-time. It should be pointed out that frequently known on-line algorithms are globally less efficiently than the corresponding off—line algorithm (some price must be paid to acquire the on-line property).

The design of on-line of a on-line problem is a current hot topics in optimization theory. For a lot of problems, the corresponding on-line problems and on-line algorithms were given in [2],[3],[4],[5]. In the following, we will give an on-line algorithm that produces SL(S) for a simple point set S.

Problem. Given a sequence of N points in the plane, $p_1, p_2, \cdots, P_N$, (Suppose that $S = \{p_i : i = 1, \cdots, N\}$ is a simple point set), find their SL(S) in such a way that p_i is processed we have $SL(\{p_1, p_2, \cdots, p_i\})$.

Lemma 2.1 Let L be a set of n line segments with endpoint set S, l be a line segment with two endpoints in S. To determine whether l is in L can be done in O(logn) time and O(n) storage with O(nlogn) preprocessing time.

Proof. Let $S = \{p_1, p_2, \cdots, p_n\}$. For the convenience of discussion, we assume that for $L(p_i, p_j) \in L$, satisfy $i < j, L(p_i, p_j) < L(p_k, p_l)$ if i<k or i=k, j<l. Sort the n line segments in L lexicographically and let $L = \{l_1, l_2, \cdots, l_n\}, l_i < l_{i+1}, i = 1, \cdots, n\text{-}1$. The sorting process can be done in O(nlogn) time and O(n) storage[PS85]. Since L is ordered, we can test whether l is in L in $O(logn)$ time. This ends the proof.

From Lemma 2.1, we have

Lemma 2.2 Let L_1 and L_2 be two line segments sets. $|L_1| = m_1, |L_2| = m_2$, then the intersection of the two sets, $L_1 \cap L_2$, can be found in $O(m_2 log m_1)$ time with $O(m_1 log m_1)$ preprocessing time and $O(m_1 + m_2)$ storage.

For investigating on-line SL(S) algorithm for a simple point set S, we need the following three lemmas.

Lemma 2.3[5] The constained triangulation on a set of n points(i.e., a set of triangulation edges may be prespecified in the problem statement) can be algorithmically constructed in $O(n\log n)$ time and $O(n)$ storage.

Lemma 2.4[5] Point location in an N-vertex planar subdivision can be effected in $O(logN)$ time using $O(N)$ storage, given $O(NlogN)$ preprocessing time.

Lemma 2.5[1] To determine whether a line segment with two endpoints in S is in $SL(S)$ can be answered in $O(n\log n)$ time and $O(n)$ storage.

For a point sequence $p_1, p_2, \cdots, p_n$, let $S_i = \{p_1, p_2, \cdots, p_i\}$, $L_{p_i}(S_{i-1}) = \{L(p_k, p_i): k = 1, \cdots, i-1\}$, L_{P_i} be the set of line segments with one endpoint p_i and in $SL(S_i)$. Suppose that we have $SL(S_i)$ and p_i, we will show how to find $SL(S_i)$. First, we construct a triangulation of S_{i-1}, $T(S_{i-1})$, this can be done in $O(ilogi)$ time from Lemma 2.3, since S_i is a simple point set. From Lemma 2.4, we can decide the location of point p_i in the subdivision consisting of $T(S_{i-1})$ in $O(logi)$ time. There are two cases, 1) p_i locates in some triangle of $T(S_{i-1})$, let $\triangle_{q_1q_2q_3}$ be the reiangle $q_1q_2q_3 \in S_{i-1}$. 2) p_i locates outside the convex hull, $CH(S_{i-1})$, $of\ S_{i-1}$.

Case 1. We know that in this case $L(q_i, p_j), j = 1, 2, 3$ are the only three line segments that may be in $SL(S_i)$ with one endpoint p_i and from Lemma 2.5 we know that O(ilogi) time is enough to test whether $L(q_j, p_i), j = 1, 2, 3$ are in $SL(S_i)$. Since p_i is a new point, some line segements in $SL(S_{i-1})$ may be vanished in $SL(S_i)$. It is obviolus that if l is in $SL(S_{i-1})$ and is not in $SL(S_i)$, then l must intersect a line segement $L(q, p_i)$ for some $q \in S_{i-1}$ ***i.e.***, l must intersect a line sebment in $Lp_i(s_{i-1})$. From Lemma 2.3, we can constuct a triangulation of S_i, $T^*(S_i)$, such that $L_{p_i}(S_{i-1}) \subseteq T^*(S_i)$. Clearly, $T^*(S_i) \cap SL(S_{i-1})$ is just the set of line segments that be in both $SL(S_{i-1})$ and $SL(S_i)$. From Lemma 2.2, we know that $T^*(S_i) \cap SL(S_{i-1})$ can be found in O(ilogi) time. Add. L_{p_i} to $T^*(S_i) \cap SL(S_{i-1})$, we have $SL(S_i)$, since $SL(S_i) = T^*(S_i) \cap SL(S_{i-1}) \cup L_{p_i}$.

Case 2. Let C_{i-1} be convex hull of S_{i-1} it is shown in [5] that we can find two supporting lines (a supporting line of a convex polygon P is a straight line l passing through a vertex of P and such that the interior of P lies entirely on one side of l) from p_i to C_{i-1} in $O(\log i)$ time and $O(i)$ storage. So we can find L_{p_i} in at most $O(i)$ time from the fact that L_{p_i} consists of the line segments with one endpoint p_i and the other endpoints on the convex hull of S_{i-1} and between the two supporting points that faced to p_i (Since $|L_{p_i}|$ may be the same order of $O(i)$). The next step is the same as in Case 1.

From the discussion in case 1 and case 2, we know that the computations from $SL(S_{i-1})$ to $SL(S_i)$ can be done in $O(i\log i)$ time. So we can design an on-line algorithm that produces $SL(S)$ in at most $O(n^2logn)$ time and $O(n)$ storage, since

$$\sum_{i=3}^{n} O(i\log i) = O(n^2\log n)$$

Theorem The $SL(S)$ of a simple point set S consists of a point sequence $p_1, p_2, \cdots, p_n$ can be found on-line in $O(n^2\log n)$ time with O(nlogn) update time, that is, in real-time.

References

[1] Y. F. Xu. Minimum weight triangulation of a planar point set. Ph. D. thesis. Institute of Applied Mathematics, Academia Sinica, 1992.

[2] A. Borodin, N. Linial and M. Sakes, An optimal on-line algorithm for metrical task systems, Proc. 19th STOC, 1987, 373-382.

[3] S. Irani, Coloring inductive graphs on-line, Proc. 31st FOCS, 1990.

[4] A. A. Melkman, On-line construction of the convex hull of a simple polyline, Inform. Proc. Lett. Vol. 25, 1987, 11-12.

[5] F. Preparata and M. I. Shamos, Computational Geometry-An Introductibin, Springer-Verlag, 1985.

求平面点集独立线段的一个联机算法

徐寅峰　　（陈丽华）
（西安交通大学管理学院）　　（北京科技大学）

提　要

令 S 为一个有限平面点集合，线段 L(p,q),p,q∈S 称为 S 的一个独立线段当且仅当不存在两个端点在 S 中的线段与 L(p,q)相交。本文给出了时间复杂度为 $O(n^2\log n)$ 和修正时间复杂度为 $O(n\log n)$ 的联机算法。这一算法与已知的脱机算法具有相同的时间复杂度。