# On Stable Line Segments in Triangulations

Andranik Mirzaian [1] Cao An Wang [2] and Yin-Feng Xu [3]

## 1 Overview

Let $S$ be a set of $n$ points in the plane and $E$ denote the set of all the line segments with endpoints in $S$. A line segment $\overline{pq}$ with $p, q \in S$ is called a **stable line segment** of all triangulations of $S$, if no line segment in $E$ properly intersects $\overline{pq}$. The intersection of all possible triangulations of $S$ then is the set of all stable line segments in $S$, denoted by $SL(S)$.

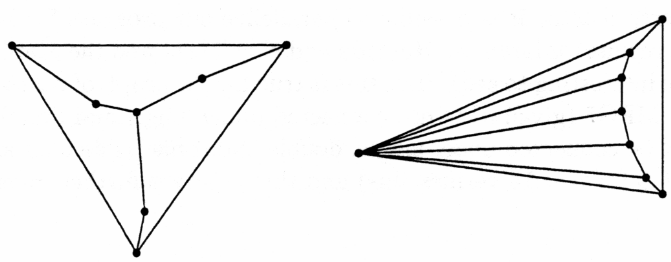
As a combinatorial problem, various properties of stable line segments of a set of planar points have been investigated in [Xu92]. It is shown that the maximum number of stable line segments in $S$ is $2(n-1)$. There is an interesting relationship between stable line segments and so-called extreme line segments $EL(S)$ [Ed86]. A line segment $\overline{pq}$ with $p, q \epsilon S$ is called an extreme line segment if $\{\overline{pq}\} = E \cap H$ for some open half-plane $H$ [Ed86]. Then, we have that

$$CH(S) \subseteq EL(S) \subseteq SL(S).$$

A more important property is the relationship between $SL(S)$ and so-called $k$-optimal triangulations. Let $T(S)$ denote a triangulation of $S$. $T(S)$ is called a **$k$-optimal triangulation** for $4 \leq k < n$, denoted by $LOT_k(S)$, if every $k$-sided simple polygon drawn from $T(S)$ is optimally triangulated by some edges of $T(S)$.

Let $SL_k(S)$ denote the intersection of all possible $LOT_k(S)$'s (i.e., the set of edges that are in every $LOT_k(S)$). Let $MWT(S)$ denote a minimum weight triangulation of $S$. Then, we have that

$$SL(S) \subseteq SL_4(S) \subseteq \cdots \subseteq SL_k(S) \cdots \subseteq SL_{n-1}(S) \subseteq MWT(S).$$



(a) (b)

Figure 1:

In some special cases of $S$, $SL(S)$ forms a connected graph as shown in Figure 1. Thus, an $MWT(S)$ can be constructed in polynomial time using the dynamic programming algorithm proposed in [Gi79, Kl80].

So far the structure properties of $SL(S)$ have been thoroughly studied, but not its algorithmic issue.

[1] andy@cs.yorku.ca, Department of Computer Science, York University, North York, Ontario, Canada M3J 1P3. His work is partially supported by NSERC.

[2] Department of Computer Science, Memorial University of Newfoundland, St.John's, NFLD, Canada A1C 5S7. His work is suported by NSERC and a special fund from the Dean of Science, Memorial University of Newfoundland.

[3] School of Management, Xi'an Jiaotong University, Xi'an, 710049, P.R. China. The work was done while this author was visiting Departments of Computer Science, at Memorial University of Newfoundland and York University.

A recent result on finding a subgraph $LOT(S)$ of $SL_4(S)$ [DM96] implies an $O(n^4)$ time and $O(n^3)$ space algorithm for finding $SL(S)$ since it is not difficult to show that

$$SL(S) \subseteq LOT(S) \subseteq SL_4(S).$$

In this paper, we shall propose two algorithms for computing $SL(S)$. One is an $O(n^2 \log n)$ time and $O(n)$ space algorithm and the other is an $O(n^2)$ time and $O(n^2)$ space algorithm.

## 2 Introduction

A triangulation of a planar point set $S$ is defined as a maximal set of non-crossing line segments which have both endpoints in $S$. A minimum weight triangulation of $S$ (denoted $MWT(S)$) is a triangulation among all possible triangulations over $S$ such that the sum of its total edge lengths is minimal. To compute an $MWT$ of a point set is an outstanding open problem, whose complexity status is unknown since 1975 [SH75, GJ79]. An $O(n^3)$ time dynamic programming algorithm for constructing an $MWT$ of a simply polygon was given independently in [Gi79, Kl80]. Based on the above mentioned dynamic programming algorithm, Anagnostou and Corneil [AC93] designed an $O(n^{3k+1})$ time algorithm for computing an $MWT$ of a point set with k nested convex polygons, and later Meijer and Rappaport [MR92] improved the time complexity to $O(n^k)$ when each of the $k$ nested polygons degenerated into a straight line segment. Xu and others [Xu92, CGJ95] showed that if a subgraph of an $MWT$ with k connected components is given, then an $MWT$ can be found in $O(n^{k+2})$ time. Up to now, none of the existing algorithms for finding an $MWT$ of a general point set achieves polynomial time bound. An alternative direction is to identify a subset of line segments in $E$ belonging to an $MWT$. The advantage of this direction is two-fold. The more such line segments are identified, the more likely the resulting subgraph will connect all the points in $S$. Then, the ultimate solution can be found in $O(n^{k+2})$ time by using dynamic programming. On the other hand, it was shown in [XZ96] that finding more line segments within an $MWT$ can improve the performance of some heuristics.

Several investigations have reported on the subgraphs of an $MWT$, [BDE96, CX96, DM96, Ke94, Xu92, Xu96, YXY94]. A trivial case is the set of line segments in all triangulations of a given point set S (i.e., a set of stable line segments $SL(S)$). No detailed work was done on the algorithms for computing $SL(S)$. In the following section, we shall propose two algorithms for computing $SL(S)$.

## 3 Algorithmic Issues

Let $J$ denote the set of all triangulations of a point set $S$, then we have the following obvious facts:

**Fact 1.** $SL(S) = \cap_{T(S) \in J} T(S)$, and

**Fact 2.** $\overline{pq} \in SL(S)$ **iff** no line segment with endpoints in $S$ properly intersects $\overline{pq}$.

Note that the Delaunay triangulation of $S$, $DT(S)$, belongs to $J$. By Fact 1, we first construct the Delaunay triangulation $DT(S)$ and then test whether the line segments in $DT(S)$ are also in $SL(S)$. Note that the number of line segments in $DT(S)$ is linearly proportional to $n$, it is easy to design an $O(n^3)$ time algorithm by testing all possible intersections of the line segments with Delaunay edges.

With a more detailed geometric analysis, we can improve the time complexity from $O(n^3)$ to $O(n^2 \log n)$ and space complexity from $O(n^2)$ to $O(n)$ or time complexity to $O(n^2)$ and space complexity remains as $O(n^2)$.

## 3.1 Algorithm 1

**Lemma 1** *Let $\overline{pq}$ be a line segment, $\{p,q\} \cup S$ be a simple point set, $|S| = n$. To determine whether there is a line segment with two endpoints in $S$ that properly intersects $\overline{pq}$ can be answered in $O(n \log n)$ time and $O(n)$ space.*

**Proof** First, by a rigid motion we can transform point $p$ to the origin and point $q$ on the $x$-axis and denote its coordinates $(x^*, 0), x^* > 0$. This can be done in $O(n)$ time. In the new coordinate system, $S$ becomes $S'$, $p \to p'$ and $q \to q'$, $p' = (0,0)$ and $q' = (x^*, 0)$, and $r = (x(r), y(r))$ in $S'$. If no points in $S'$ are below (or above) $x$-axis, then no line segment with two endpoints in $S'$ intersects the line segment $L(p', q')$. If there are points with $y(p_i) > 0$ and $y(p_j) < 0$ for $p_i, p_j \in S'$, we divide $S'$ into two subsets

$$
\begin{aligned}
S'_+ &= \{p \mid y(p) > 0, p \in S'\} \\
S'_- &= \{p \mid y(p) < 0, p \in S'\}
\end{aligned}
$$

This step can be done in $O(n)$ time. For convenience of discussion, we assume there is no point $r$ in $S'$ with $x(r) = 0$ or $x(r) = x^*$, so we can divide $S'_+$ and $S'_-$ into $S'_+(-1), S'_+(0)$, $S'_+(+1)$ and $S'_-(-1), S'_-(0)$, $S'_-(+1)$ as follows:

$$
\begin{aligned}
S'_+(-1) &= \{p \mid x(p) < 0, p \in S'_+\} \\
S'_+(+0) &= \{p \mid 0 < x(p) < x^*, p \in S'_+\} \\
S'_+(+1) &= \{p \mid x(p) > x^*, p \in S'_+\} \\
S'_-(-1) &= \{p \mid x(p) < 0, p \in S'_-\} \\
S'_-(-0) &= \{p \mid 0 < x(p) < x^*, p \in S'_-\} \\
S'_-(+1) &= \{p \mid x(p) > x^*, p \in S'_-\}
\end{aligned}
$$

and clearly we have

$$
\begin{aligned}
S'_+ &= S'_+(-1) \cup S'_+(0) \cup S'_+(+1) \\
S'_- &= S'_-(-1) \cup S'_-(0) \cup S'_-(+1).
\end{aligned}
$$

This step can be completed in $O(n)$ time. As shown in Figure 2(A), if $S'_+(0) \neq \emptyset$ and $S'_-(0) \neq \emptyset$, then there is a line segment with one endpoint in $S'_+(0)$ and the other in $S'_-(0)$ which intersects $L(p', q')$ (Figure 2(B)). To determine whether or not $S'_-(0)$ and $S'_+(0)$ is empty needs only $O(n)$ time. So without lose of generality, we can assume that $S'_+(0) = \emptyset$.

Second, we sort points in $S'_+(-1)$ and $S'_+(+1)$ lexicographically by polar angle at $p'$ and $q'$ respectively. In the new polar coordinate system, $S'_+(+1)$ becomes $S'_+(+1)_{p'}$ and $S'_+(-1)$ becomes $S'_+(-1)_{p'}$, similarly $S'_+(+1)_{q'}$ and $S'_+(-1)_{q'}$. Let $|\ S'_+(+1)\ | = m_1, |\ S'_+(-1)\ | = m_2, \alpha_{p'}(r)$ denote the polar angle of $r$ from origin $p'$ and $\alpha_{q'}(r)$ denote the polar angle of $r$ from $q'$. We have

$$
\begin{aligned}
S'_+(+1)_{p'} &= \{p_i^+ \mid \alpha_{p'}(p_{i+1}^+) > \alpha_{p'}(p_i^+), i = 1, 2, \cdots, m_1 - 1\} \\
S'_+(+1)_{q'} &= \{q_i^+ \mid \alpha_{q'}(q_{i+1}^+) > \alpha_{q'}(q_i^+), i = 1, 2, \cdots, m_1 - 1\} \\
S'_+(-1)_{p'} &= \{p_i^- \mid \alpha_{p'}(p_{i+1}^-) > \alpha_{p'}(p_i^-), i = 1, 2, \cdots, m_2 - 1\} \\
S'_+(-1)_{q'} &= \{q_i^- \mid \alpha_{q'}(q_{i+1}^-) > \alpha_{q'}(q_i^-), i = 1, 2, \cdots, m_2 - 1\}.
\end{aligned}
$$

The above sorting step can be done in $O(n \log n)$ time [PS85]. Let $u \in S'_-$. We discuss whether there is a line segment with one endpoint u and another endpoint in $S'_+$ that crosses $\overline{pq}$ in following three cases.

Case 1: $u \in S'_-(0)$.

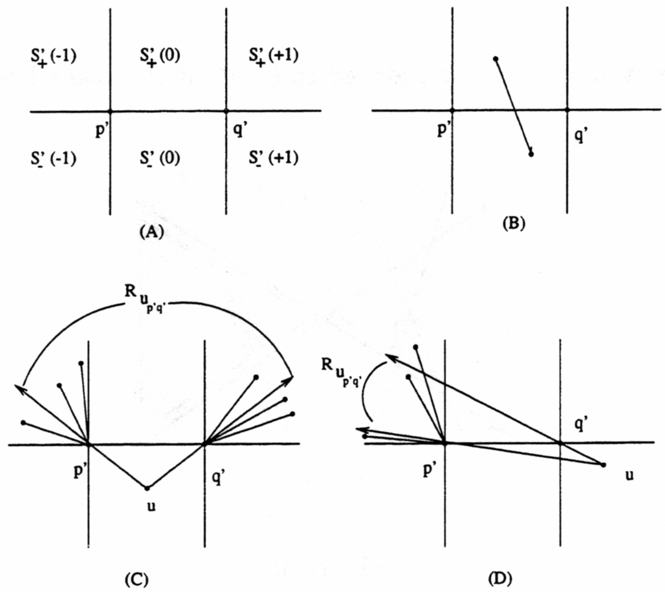



Figure 2:

Construct two rays $up'$ and $uq'$, let $\alpha_{up'}$ and $\alpha_{uq'}$ be the polar angles of $up'$ and $uq'$ in polar coordinate system with anchor points $p'$ and $q'$ respectively. Testing the rank of $\alpha_{up'}$ in $S'_+(+1)_{p'}$ and $\alpha_{uq'}$ in $S'_+(-1)q'$, can be done in $O(\log n)$ time binary search. This way we can find out whether there exists a point in $S'_+(+1) \cup S'_+(-1)$ lying in the angle region $R_{up'q'}$ between the two rays $up'$ and $uq'$. (Figure 2(C)).

Case 2: $u \in S'_-(+1)$.

Construct two rays $up'$ and $uq'$, let $\alpha_{up'}$ and $\alpha_{uq'}$ be the same as in Case 1. Test the ranks of $\alpha_{up'}$ and $\alpha_{uq'}$ in $S'_+(-1)_{p'}$ and $S'_+(-1)_{q'}$. (Note only points in $S'_+(-1)$ needs to be tested, since $u \in S'_-(+1)$). Thus, in $O(\log n)$ time we can find out whether there exists a point in $S'_+(-1)$ which lies in the angular region $R_{u_{p'q'}}$. (Figure 2(D)).

Case 3: $u \in S'_-(-1)$.

A similarly analysis as in Case 2.

It has shown in the above discussion that the total computations to determine whether a line segment with two endpoints in $S$ intersects $\overline{pq}$ take at most $O(n \log n)$ time. □

In the following, $LI(S, \overline{pq})$ denotes the above algorithm that answers whether or not there exists a line segment in $E$ that crosses $\overline{pq}$. By the above lemma, algorithm $LI(S, \overline{pq})$ takes $O(n \log n)$ time and $O(n)$ space.

**Lemma 2** *Let $\overline{pq}$ be a line segment, and let $\{p, q\} \cup S$ be a point set in general positions, $\mid S \mid = n$. Let $E$ denote all the line segments in $S$. Then, whether or not $\overline{pq}$ crosses an element of $E$ can be answered in $O(nlogn)$ time and $O(n)$ storage.*

**Theorem 1** *$SL(S)$ can be found in $O(n^2 \log n)$ time and $O(n)$ space, where $\mid S \mid = n$.*

**Proof** It is clear that $SL(S)$ must be contained in the Delaunay triangulation $DT(S)$. Thus, we start with $DT(S)$, which can be constructed in $O(n \log n)$ time and $O(n)$ space. Using algorithm $LI(S, \overline{pq})$ we test if an edge $\overline{pq}$ of $DT(S)$ belongs to $SL(S)$ in $O(n \log n)$ time and $O(n)$ space. The theorem follows since the number of edges in $DT(S)$ is $O(n)$.

□

### 3.2 Algorithm 2

The above time complexity bound can be reduced to $O(n^2)$ if the space bound increases to $O(n^2)$.

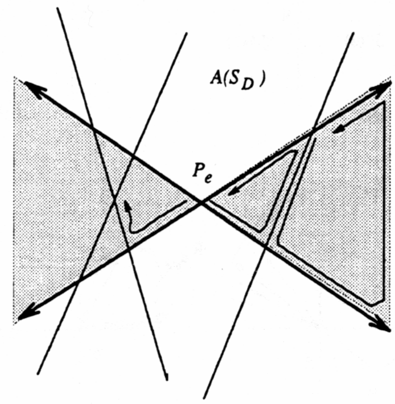



Figure 3:

**Algorithm 2**

- Find the arrangement for $n$ lines, where each line is the dual of a point of $S$ in the dual plane. Denote this arrangement as $A(S_D)$.
- Find $DT(S)$; For each Delaunay edge $e$ of $DT(S)$ DO.
  - Let $p_e$ be the intersection point of the dual lines of the endpoints of $e$. Let $W(p_e)$ be the double wedge determined by these two dual lines. Traverse the portion of $A(S_D)$ inside $W(p_e)$, starting at $p_e$. (Refer to Figure 3.)
  - If a vertex of $A(S_D)$ is found properly inside $W(p_e)$, then report '$e$ is not in $SL$';
  - Otherwise, report '$e$ is in $SL$'
- EndDo.

**Theorem 2** *$SL(S)$ can be found in $O(n^2)$ time and $O(n^2)$ space, where $|S| = n$.*

## 4 Concluding Remarks

We proposed an $O(n^2 \log n)$ time and $O(n)$ space algorithm and an $O(n^2)$ time and space algorithm for finding $SL(S)$. It is interesting to see whether $SL_4(S)$ can be computed in polynomial time.

## 5 References


[AC93] E. Anagnostou and D. Corneil "Polynomial time instances of the minimum weight triangulation problem," Computational Geometry: Theory and Application, 3(1993), 247-259.

[BDE96] P. Bose, L. Devroye and W. Evans, "Diamonds are not a Minimum Weight Triangulation's Best Friend," UBC CS TR-96-01, 1996.

[CGT95] S. W. Cheng, M. Golin and J. Tsang, "Expected case analysis of $\beta$-skeletons with applications to the construction of minimum weight triangulations," 7CCCG Conf. Proc., Quebec City, 1995, 279-284.

[CX96] S. W. Cheng and Y. F. Xu, "Approaching the largest $\beta$-skeleton within the minimum weight triangulation," Proc. 12th Ann. ACM Symp. Computational Geometry, Philadelphia, 1996.

[DM96] M. T. Dickerson and M. H. Montague, "The exact minimum weight triangulation," Proc. 12th Ann. ACM Symp. Computational Geometry, Philadelphia, 1996.

[Ed86] H. Edelsbrunner, "The number of extreme pairs of finite point-sets in Euclidean spaces," J. Combin. Theory, Ser. A. 43, 1996, 344-349.

[Gi79] P. D. Gilbert, "New results in planar triangulations," TR-850, Univ. of Illinois, Coordinated Science Lab, 1979.

[GJ79] M. Garey and D. Johnson, "Computers and Intractability, A Guide to the Theory of NP-completeness," Freeman, 1979.

[Ke94] J. M. Keil, "Computing a subgraph of the minimum weight triangulation," Computational Geometry: Theory and Applications 4(1994) 13-26.

[K180] G. Klincsek, "Minimal triangulations of polygonal domains," Ann. Discrete Math., 9(1980) 121-123.

[MR92] H. Meijer and D. Rappaport, "Computing the minimum weight triangulation of a set of linearly ordered points," Information Processing Letters, Vol. 42, pp.35-38, 1992.

[SH75] M. I. Shamos and D. Hoey, "Closest point problems," Proc. 16th IEEE Symp. Foundations of Computer Science, 1975, 151-162.

[Xu92] Y.F. Xu, "Minimum Weight triangulation problem of a planar point set," Ph.D. Thesis, Institute of Applied Mathematics, Academia Sinica, Beijing, 1992.

[Xu96] Y.F. Xu, "On stable line segments in all triangulations," JCU-Applied Math, 1996, to appear.

[XZ96] Y.F. Xu and D. Zhou, "Improved heuristics for the minimum weight triangulation," Acta Mathematics Applicatae Sinica, Vol. 11, No. 4, 1995, 359-368.

[YXY94] B.T. Yang, Y.F. Xu and Z.Y. You, "A chain decomposition algorithm for the proof of a property on minimum weight triangulations," Proc. 5th International Symposium on Algorithms and Computation (ISAAC '94), LNCS 834, Springer-Verlag, 1994, 423-427.