Pointed Binary Encompassing Trees: Simple and Optimal

Michael Hoffmann
Institute for Theoretical Comp. Sci.
ETH Zürich, CH-8092, Switzerland
hoffmann@inf.ethz.ch

Csaba D. Tóth
Department of Mathematics
MIT, Cambridge, MA 02139
toth@math.mit.edu

Abstract. For n disjoint line segments in the plane we can construct a binary encompassing tree such that every vertex is *pointed*, what's more, at every segment endpoint all incident edges lie in a halfplane defined by the incident input segment. Our algorithm runs in $O(n \log n)$ time which is known to be optimal in the algebraic computation tree model.

Introduction. Interconnection graphs of disjoint line segments in the plane are fundamental structures in computational geometry, and often more complex objects are modelled by their boundary segments or polygons. One particularly well-studied example is a crossing-free spanning graph: the *encompassing graph* for disjoint line segments in the plane is a connected planar straight line graph (PSLG) whose vertices are the segment endpoints and that contains every input segment as an edge.

A simple construction shows that not every set of n disjoint segments in the plane admits an *encompassing path*. But there is always a path that encompasses $\Theta(\log n)$ segments and does not cross any other input segment [6]. The question, whether encompassing trees of bounded degree exist, was answered in the affirmative by Bose and Toussaint [4]. Later Bose, Houle and Toussaint [3] constructed an encompassing tree of maximal degree *three* in $O(n \log n)$ time and proved that the runtime is optimal.

Recently, Hoffmann, Speckmann, and Tóth [5] have shown that for every set of disjoint segments a *pointed* binary encompassing tree can be constructed in $O(n^{4/3} \log n)$ time. A PSLG is *pointed* iff at every vertex p all incident edges lie on one side of a line through p.

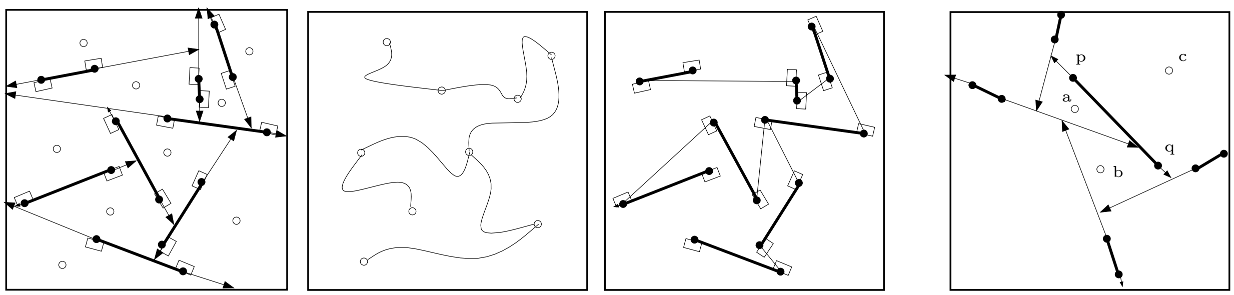
Pointed PSLGs are tightly connected to minimum pseudo-triangulations, which have numerous applications in motion planning [10], kinetic data structures [8], collision detection [1], and guarding [9]. Streinu [10] showed that a minimum pseudo-triangulation of V is a pointed PSLG on the vertex set V with a maximal number of edges. As opposed to triangulations, there is always a bounded degree pseudo-triangulation of a set of points in the plane [7]. A bounded degree pointed encompassing tree for disjoint segments leads to a bounded degree pointed encompassing pseudo-triangulation, due to a result of Aichholzer et al. [2].

In this paper, we improve all previous results on encompassing trees of n segments and give a simple algorithm to construct a pointed binary encompassing tree in optimal $O(n \log n)$ time. Moreover, for every vertex of the tree all incident edges lie on one side of the line through the incident input segment.

Theorem 1 *For a set S of n disjoint line segments in the plane, we can build in $O(n \log n)$ time a binary encompassing tree such that for every segment endpoint p of every input segment pq the edges incident to p lie in a halfplane bounded by the line through pq.*

Tunnel Graphs. The free space around the segments can be partitioned into $n+1$ convex cells[1] by the following well known partitioning algorithm: For every segment endpoint p of every input segment pq, extend pq beyond p until it hits another input segment, a previously drawn extension, or to infinity.

[1]For simplicity, we assume that no three segment endpoints are collinear.



(a) Partition. (b) Tunnel Graph. (c) Encompassing Tree. (d) Disconnected.

Fig. 1: An example for a partition with an assignment (a), the corresponding tunnel graph (b), and the resulting tree (c). A partition for which no assignment gives a connected tunnel graph (d).

Consider a set of segments S and a convex partition $P(S)$ obtained by the above algorithm. Let us assign every p to an incident cell $\tau(p)$ of the partition. A key tool in our proof is the *tunnel* graph $T(S, P(S), \tau)$ of $P(S)$ and the assignment τ, defined as follows: The nodes of T correspond to the convex cells of $P(S)$, two nodes a and b are connected by an edge iff there is a segment $pq \in S$ such that $\tau(p) = a$ and $\tau(q) = b$. It is easy to see that the tunnel graph is planar. As T has $n + 1$ nodes and n edges, it is connected iff it is a tree.

Theorem 2 *For any set S of n disjoint line segments, we can construct in $O(n \log n)$ time a convex partition $P(S)$ and an assignment τ such that the tunnel graph $T(S, P(S), \tau)$ is a tree.*

We note that Theorem 2 does not hold for every partition: Fig. 1(d) shows 7 disjoint line segments and a convex partition such that there is no assignment for which the tunnel graph is connected. The proof of Theorem 2 can be found in the full version of this paper. Our main result follows from Theorem 2.

Proof of Theorem 1. Consider a partition $P(S)$ and an assignment τ provided by Theorem 2. In each cell connect the segment endpoints assigned to it by a simple path.

The resulting graph is clearly a PSLG that encompasses the input segments. The maximal degree is three because we add at most two new edges at every segment endpoint. It remains to prove connectivity. Let p and r be two segment endpoints. We know that the tunnel graph is connected, so there is an alternating sequence of cells and segments $(a_1 = \tau(p), p_1q_1, a_2, \ldots, p_{k-1}q_{k-1}, a_k = \tau(r))$ such that $\tau(p_i) = a_i$ and $\tau(q_i) = a_{i+1}$, for every i. As all segment endpoints assigned to the same cell are connected, this path corresponds to a path in the encompassing graph. □

References

[1] P. K. Agarwal, J. Basch, L. J. Guibas, J. Hershberger, and L. Zhang, Deformable free space tilings for kinetic collision detection, in *Proc. 4th WAFR*, 2001, 83–96.

[2] O. Aichholzer, M. Hoffmann, B. Speckmann, and Cs. D. Tóth, Degree bounds for constrained pseudo-triangulations, in: *Proc. 15th CCCG*, 2003, pp. 155–158.

[3] P. Bose, M. E. Houle, and G.T. Toussaint, Every set of disjoint line segments admits a binary tree, *Discrete Comput Geom.* **26** (2001), 387–410.

[4] P. Bose and G. T. Toussaint, Growing a tree from its branches, *J. Algorithms* **19** (1995), 86–103.

[5] M. Hoffmann, B. Speckmann, and Cs. D. Tóth, Pointed binary encompassing trees, in *Proc. 9th SWAT*, 2004.

[6] M. Hoffmann and Cs. D. Tóth, Alternating paths through disjoint line segments, *Inf. Proc. Letts.* **87** (2003), 287–294.

[7] L. Kettner, D. Kirkpatrick, A. Mantler, J. Snoeyink, B. Speckmann, and F. Takeuchi, Tight degree bounds for pseudo-triangulations of points, *Comput. Geom. Theory Appl.* **25** (2003), 1–12.

[8] D. Kirkpatrick and B. Speckmann, Kinetic maintenance of context-sensitive hierarchical representations for disjoint simple polygons, in *Proc. 18th SoCG*, 2002, pp. 179–188.

[9] B. Speckmann and Cs. D. Tóth, Allocating vertex π-guards in simple polygons, *14th SODA*, 2003, pp. 109–118.

[10] I. Streinu, A combinatorial approach to planar non-colliding robot arm motion planning, *41st FOCS*, 2000, pp. 443–453.