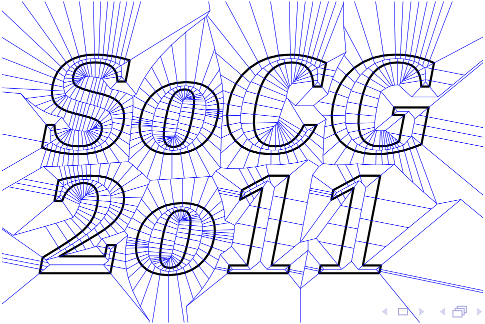
Theoretical and Practical Results on Straight Skeletons of Planar Straight-Line Graphs

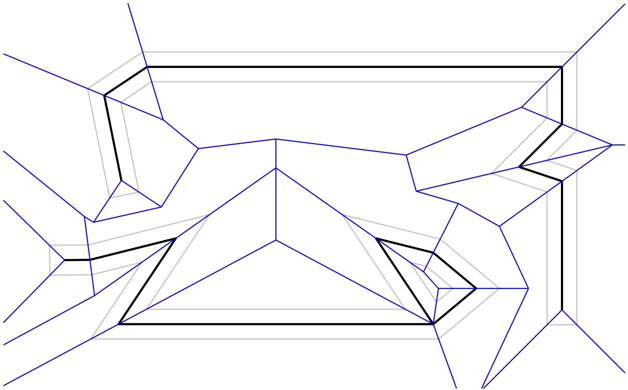
Stefan Huber Martin Held

University of Salzburg
Computer Science Department
Salzburg, Austria



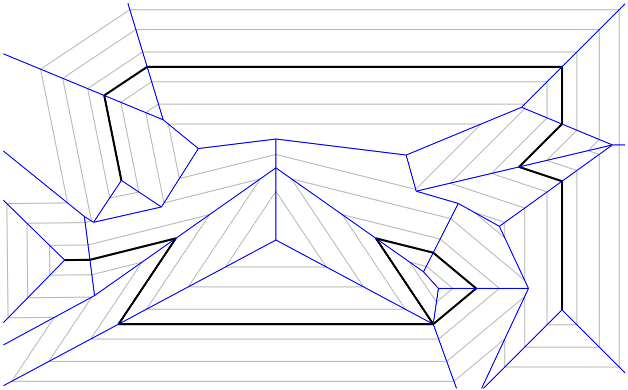
Straight skeleton of a PSLG G: Definition

- [Aichholzer and Aurenhammer, 1998]: self-parallel wavefront propagation.
  - Topological events:



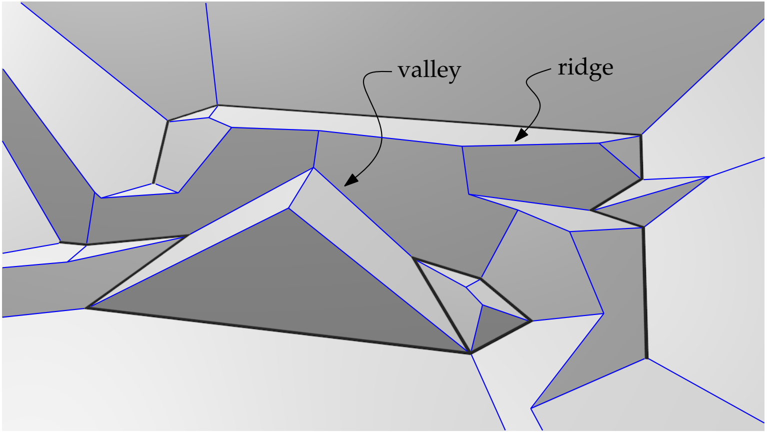
Straight skeleton of a PSLG G: Definition

- [Aichholzer and Aurenhammer, 1998]: self-parallel wavefront propagation.
  - Topological events:
    - edge events
    - split events
- **Notation:** wavefront $\mathcal{W}(G, t)$, straight skeleton $\mathcal{S}(G)$, arcs and faces



Terrain model

- $\mathcal{T}(G) := \bigcup_{t \geq 0} \mathcal{W}(G, t) \times \{t\}$
- $\mathcal{S}(G)$ is the projection of valleys and ridges onto the ground plane.
- If one knows $\mathcal{T}(G)$ then one knows $\mathcal{S}(G)$, and vice versa.



Prior work

Algorithms with sub-quadratic runtime:

- [Eppstein and Erickson, 1999]
  $O(n^{17/11+\epsilon})$ runtime, PSLGs as input, very complex, no implementation.
- [Cheng and Vigneron, 2007]
  $O(n\sqrt{n}\log^2 n)$ expected runtime, "non-degenerated" polygons with holes as input, no implementation.

Prior work

Algorithms with sub-quadratic runtime:

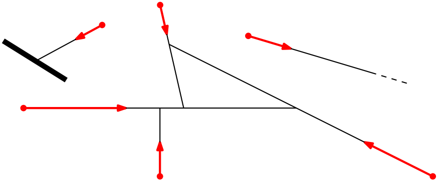
- [Eppstein and Erickson, 1999]
  $O(n^{17/11+\epsilon})$ runtime, PSLGs as input, very complex, no implementation.
- [Cheng and Vigneron, 2007]
  $O(n\sqrt{n}\log^2 n)$ expected runtime, "non-degenerated" polygons with holes as input, no implementation.

Implementations:

- By F. Cacciola, shipped with CGAL, only polygons with holes, **quadratic runtime and memory footprint in practice.**

Motorcycle graph

- **Motorcycle**: moving point, constant velocity.
- **Trace**: left behind each motorcycle.
- **Crash**: motorcycle reaches another's trace.



- Introduced by [Eppstein and Erickson, 1999].
- Used by [Cheng and Vigneron, 2007] for their straight-skeleton algorithm.
  - Motorcycle graph **induced by** a simple non-degenerate polygon.
- Additionally: **wall**: solid straight-line segment.

Roadmap for a practical straight-skeleton algorithm

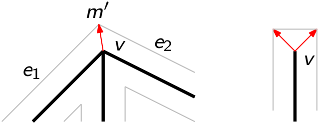
1. Generalized motorcycle graph $\mathcal{M}(G)$ induced by an arbitrary PSLG G.
   - [Cheng and Vigneron, 2007] excluded so-called vertex events.
   - **Basic requirement:** $\mathcal{M}(G)$ should cover all reflex arcs of $\mathcal{S}(G)$.
2. Exploit the geometric relation between $\mathcal{M}(G)$ and $\mathcal{S}(G)$ in order to come up with a (practical) straight-skeleton algorithm.

Definition

A **reflex arc** of $\mathcal{S}(G)$ is traced out by a reflex wavefront vertex. Likewise for **convex arcs**.

Ingredients of the motorcycle graph:

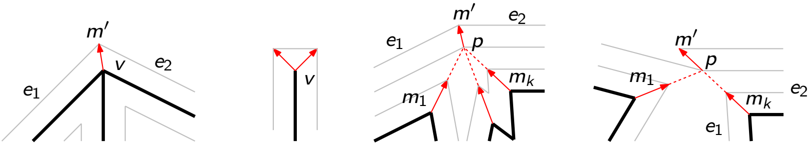
- **Walls:** each edge of G is a wall.
- **Motorcycles:**
  - **(a), (b):** We launch a motorcycle at every reflex vertex v of $\mathcal{W}(G, 0)$.



(a) (b)

Ingredients of the motorcycle graph:

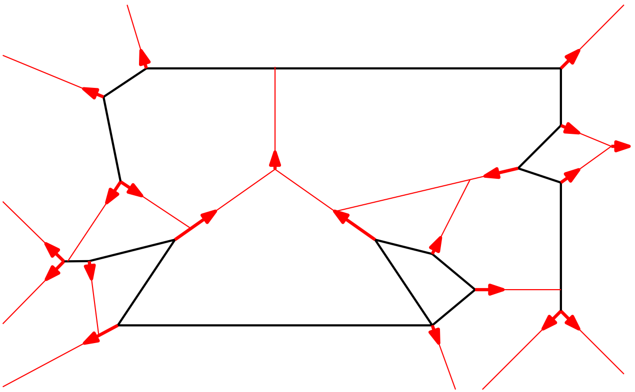
- **Walls:** each edge of G is a wall.
- **Motorcycles:**
  - **(a), (b):** We launch a motorcycle at every reflex vertex v of $\mathcal{W}(G, 0)$.
  - **(c), (d)**: If $m_1, \ldots, m_k$ crash simultaneously at p such that a disk around p is partitioned into a reflex and convex slices then we **launch a new motorcycle** m' starting at p.



(a) (b) (c) (d)

Motorcycle graph induced by a PSLG

We denote the resulting motorcycle graph by $\mathcal{M}(G)$.



Geometric relation between $\mathcal{M}(G)$ and $\mathcal{S}(G)$

Lemma

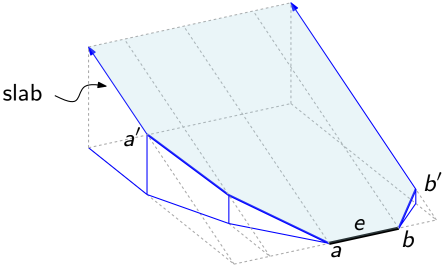
Consider a point p of $\mathcal{M}(G)$ which does not coincide with G. Then a local disk around p is tessellated into convex slices by $\mathcal{M}(G)$.

Theorem

The reflex arcs of $\mathcal{S}(G)$ are covered by $\mathcal{M}(G)$.

Alternative characterization of $\mathcal{S}(G)$

- Define for every wavefront edge a 3D slab based on $\mathcal{M}(G)$.

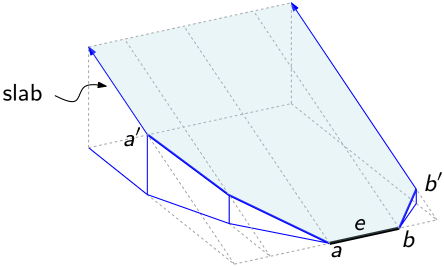


Theorem

The lower envelope $L(G)$ of these slabs is equal to $\mathcal{T}(G)$.

Alternative characterization of $\mathcal{S}(G)$

- Define for every wavefront edge a 3D slab based on $\mathcal{M}(G)$.



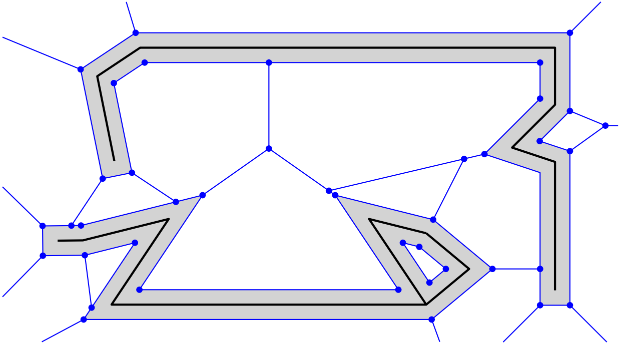
Theorem

The lower envelope $L(G)$ of these slabs is equal to $\mathcal{T}(G)$.

- Extends a result of [Eppstein and Erickson, 1999]. Their slabs are bounded below by (tilted) reflex straight-skeleton arcs.
- Extends a result of [Cheng and Vigneron, 2007]. They considered simple non-degenerated polygons as input.

A wavefront-type algorithm

- $\mathcal{M}(G, t)$: those parts of $\mathcal{M}(G)$ which have not been swept by the wavefront until time t.
- Extended wavefront $\mathcal{W}^*(G, t)$: the overlay of $\mathcal{W}(G, t)$ and $\mathcal{M}(G, t)$.

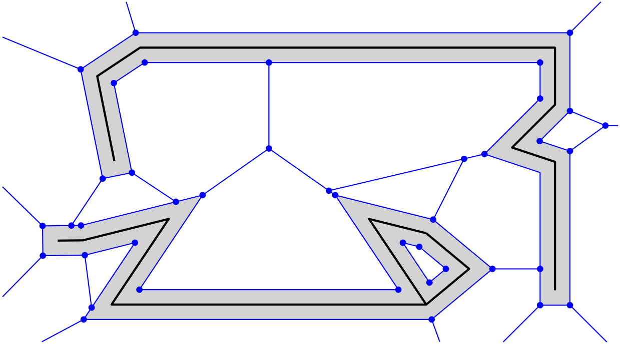


- We simulate the propagation of $\mathcal{W}^*(G, t)$.

Key properties of $\mathcal{W}^*(G, t)$

Corollary

Split events happen within the corresponding motorcycle traces and consequently within the extended wavefront $\mathcal{W}^*(G, t)$.



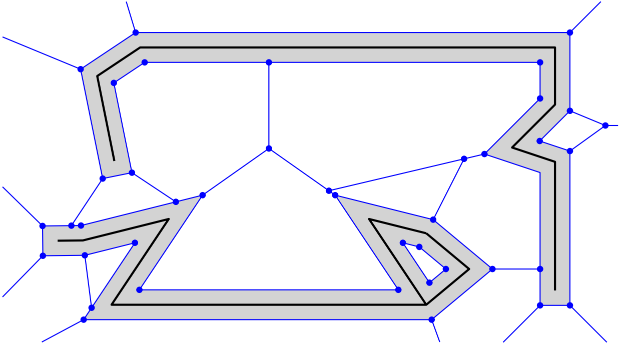
Key properties of $\mathcal{W}^*(G, t)$

Lemma

For any $t \geq 0$ the set $\mathbb{R}^2 \setminus \bigcup_{t' \in [0,t]} \mathcal{W}^*(G, t')$ consists of open convex faces.

Corollary

Only neighboring vertices can meet during the propagation of $\mathcal{W}^*(G, t)$.



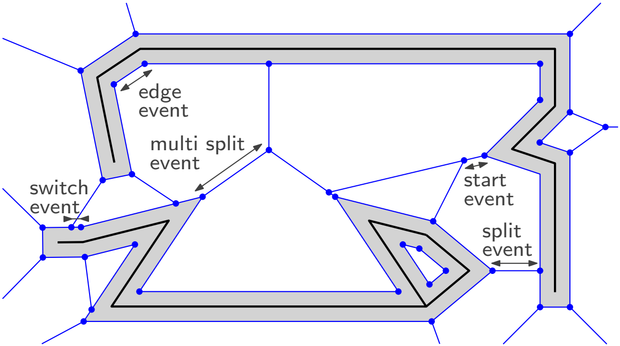
Basic algorithm

Event: a topological change of $\mathcal{W}^*(G,t)$, i.e. an edge of $\mathcal{W}^*(G,t)$ collapsed to zero length.

Algorithm

1. Compute the initial extended wavefront $\mathcal{W}^*(G,0)$.
2. Keep events in **priority queue** and process them in chronological order.

Algorithmic details: Types of events



Runtime complexity

- Switch events:
  - A convex vertex does not meet a moving Steiner point twice.
  - Hence, the number k of switch events is in $O(nr)$, where r denotes the number of reflex wavefront vertices.
- All other events can be processed in total $O(n \log n)$ time.

Theorem

If $\mathcal{M}(G)$ is given then our algorithm takes $O((n+k)\log n)$ time, where k is the number of switch events, with $k \in O(nr)$.

Runtime complexity

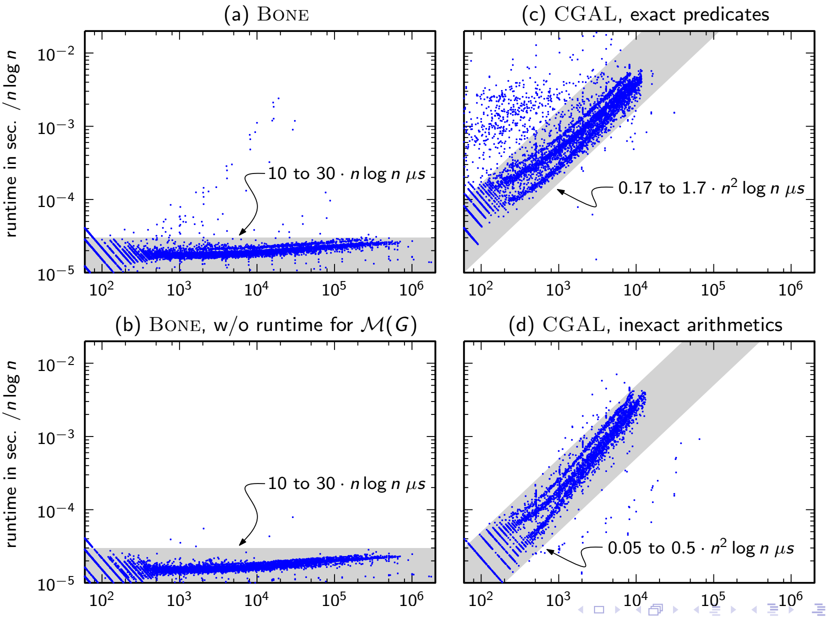
- Switch events:
  - A convex vertex does not meet a moving Steiner point twice.
  - Hence, the number k of switch events is in $O(nr)$, where r denotes the number of reflex wavefront vertices.
- All other events can be processed in total $O(n \log n)$ time.

Theorem

If $\mathcal{M}(G)$ is given then our algorithm takes $O((n+k)\log n)$ time, where k is the number of switch events, with $k \in O(nr)$.

- $k \in O(n)$ for real word data, as confirmed by experiments.
- $\mathcal{M}(G)$ is computed by MOCA [Huber and Held, 2011].
  - $O(n \log n)$ runtime for practical input.

Experimental results: Implementation Bone



Experimental results: Implementation Bone

Random polygons generated by RPG.

Size n	Bone		CGAL	
	MB	factor	MB	factor
256	1.44		3.77	
512	2.65	1.8x	13.4	3.5x
1 024	5.06	1.9x	51.1	3.8x
2 048	9.86	1.9x	201	3.9x
4 096	19.5	2.0x	792	3.9x
8 192	38.7	2.0x	3 197	4.0x
16 384	77.1	2.0x	12 600	3.9x

Table: Memory usage of Bone and CGAL

Summary

- Theory:
  - Generalized motorcycle graph to PSLGs.
  - Extended important results of [Eppstein and Erickson, 1999] and [Cheng and Vigneron, 2007].
  - An application: straight skeleton algorithm using graphics hardware.
- Implementation BONE:
  - Handles arbitrary PSLG as input.
  - Promising experimental results show an $O(n \log n)$ runtime for practical input.
  - By a linear factor faster and more space-efficient than CGAL.

Future work:

- Boost BONE to industrial strength.
- Employing MPFR (almost done).
- Employing CORE (in progress).

Finish — any questions?

Figure: Terrain based on the straight skeleton of "SoCG 2o11". Generated by BONE and rendered with the open-source modeling software BLENDER.

Aichholzer, O. and Aurenhammer, F. (1998).
Straight Skeletons for General Polygonal Figures in the Plane.
In Samoilenko, A., editor, *Voronoi's Impact on Modern Science, Book 2*, pages 7–21. Institute of Mathematics of the National Academy of Sciences of Ukraine, Kiev, Ukraine.

Cheng, S.-W. and Vigneron, A. (2007).
Motorcycle Graphs and Straight Skeletons.
Algorithmica, 47:159–182.

Eppstein, D. and Erickson, J. (1999).
Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions.
Discrete Comput. Geom., 22(4):569–592.

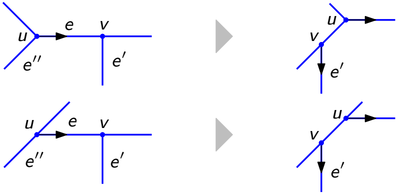
Huber, S. and Held, M. (2011).
Motorcycle Graphs: Stochastic Properties Motivate an Efficient Yet Simple Implementation.
ACM J. Experimental Algorithmics.

Appendix

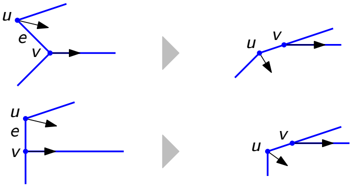
Algorithmic details: types of events

(a) edge event

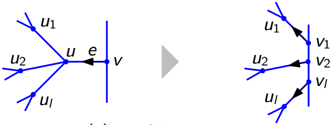
(b) split event



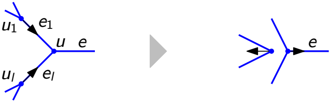
(c) start event



(d) switch event



(e) multi start event



(f) multi split event

Worst-case runtime complexity