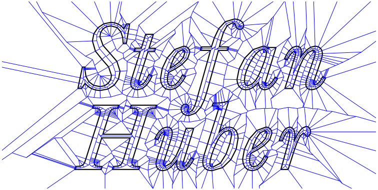
Computing Straight Skeletons and Motorcycle Graphs: Theory and Practice



Universität Salzburg, Austria

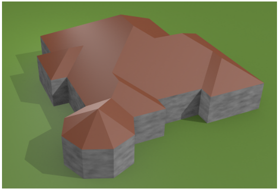
August 4, 2011

Motivation

Consider the following problems in *computational geometry*:

Motivation

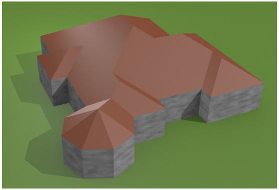
Consider the following problems in *computational geometry*:



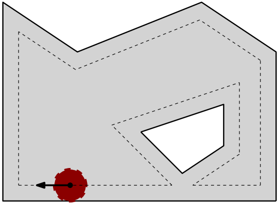
Roof construction

Motivation

Consider the following problems in *computational geometry*:

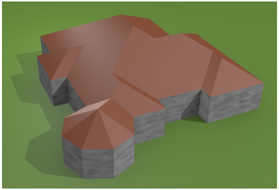


Roof construction

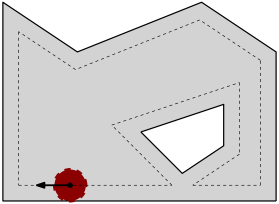


Tool path generation

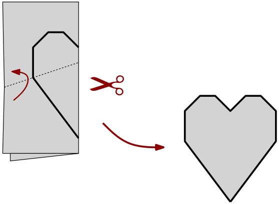
Consider the following problems in *computational geometry*:



Roof construction



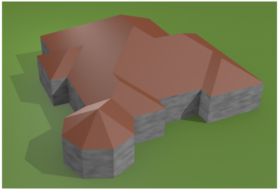
Tool path generation



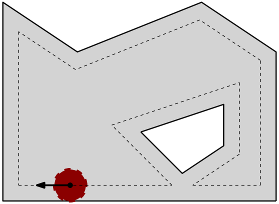
Fold-and-cut problem

Motivation

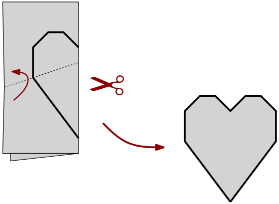
Consider the following problems in *computational geometry*:



Roof construction



Tool path generation



Fold-and-cut problem

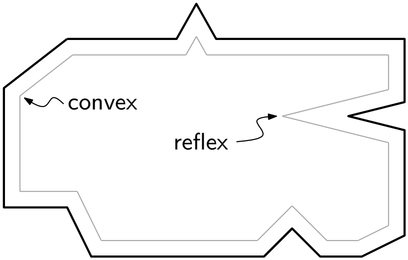
Using the straight skeleton, we obtain

- simple,
- efficient,
- numerically stable

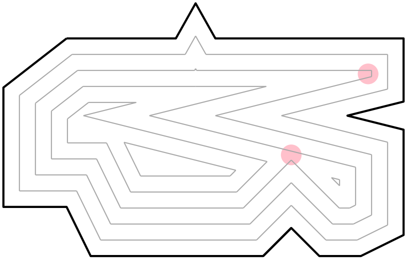
algorithms for many problems in computational geometry.

What is the straight skeleton?

- Introduced by [Aichholzer et al., 1995] for simple polygons P.
- Definition based on **wavefront propagation process**:

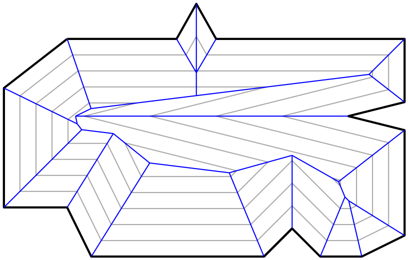


- Introduced by [Aichholzer et al., 1995] for simple polygons P.
- Definition based on **wavefront propagation process**:
  - edge events,
  - split events.



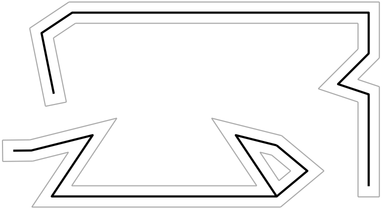
What is the straight skeleton?

- Introduced by [Aichholzer et al., 1995] for simple polygons P.
- Definition based on **wavefront propagation process**:
  - edge events,
  - split events.
- Straight skeleton $\mathcal{S}(P)$: set of loci traced out by wavefront vertices.



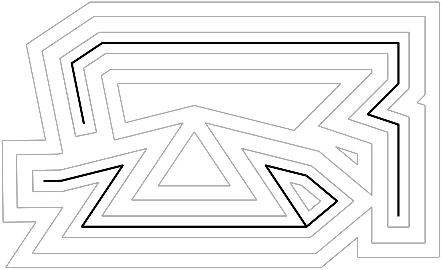
What is the straight skeleton?

- Extended to PSLGs[1] G by [Aichholzer and Aurenhammer, 1996].
  - Defined on the entire plane.
  - Rectangular caps at terminal vertices.



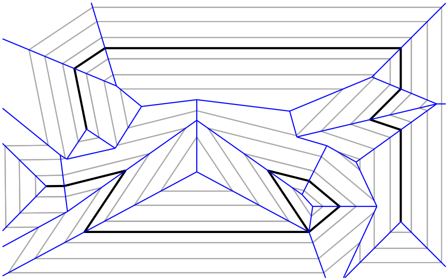
[1] Planar Straight-Line Graph

- Extended to PSLGs[1] G by [Aichholzer and Aurenhammer, 1996].
  - Defined on the entire plane.
  - Rectangular caps at terminal vertices.
- Straight skeleton $\mathcal{S}(G)$: set of loci traced out by wavefront vertices.



[1] Planar Straight-Line Graph

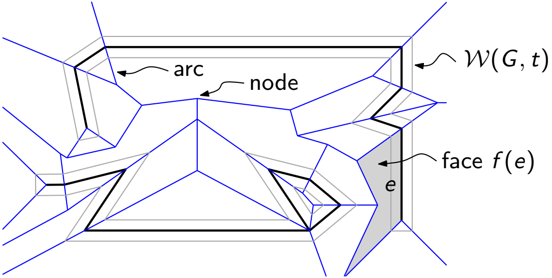
- Extended to PSLGs[1] G by [Aichholzer and Aurenhammer, 1996].
  - Defined on the entire plane.
  - Rectangular caps at terminal vertices.
- Straight skeleton $\mathcal{S}(G)$: set of loci traced out by wavefront vertices.



[1] Planar Straight-Line Graph

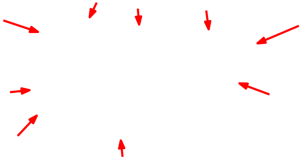
What is the straight skeleton?

- Extended to PSLGs[1] G by [Aichholzer and Aurenhammer, 1996].
  - Defined on the entire plane.
  - Rectangular caps at terminal vertices.
- Straight skeleton $\mathcal{S}(G)$: set of loci traced out by wavefront vertices.



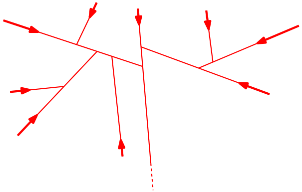
[1] Planar Straight-Line Graph

- Introduced by [Eppstein and Erickson, 1999].
- A **motorcycle** is a moving point with
  - a start point and
  - a constant velocity.



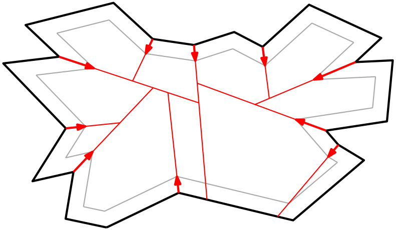
Motorcycle graph

- Introduced by [Eppstein and Erickson, 1999].
- A **motorcycle** is a moving point with
  - a start point and
  - a constant velocity.
- Consider n motorcycles $m_1, \ldots, m_n$.
  - Each motorcycle leaves a **trace** behind.
  - A motorcycle **crashes** when reaching another motorcycle's trace.
  - The **motorcycle graph** $\mathcal{M}(m_1, \ldots, m_n)$ is the arrangement of all traces.

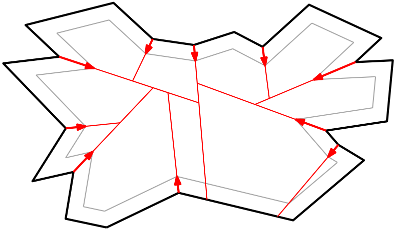


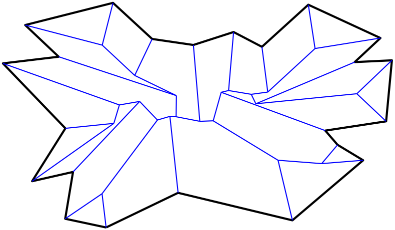
Motorcycle graph

- Motorcycle graph $\mathcal{M}(P)$ induced by a polygon P:
  - A motorcycle for each reflex wavefront vertex, with the same velocity.
  - The edges of P are considered to be walls.



- Motorcycle graph $\mathcal{M}(P)$ induced by a polygon P:
  - A motorcycle for each reflex wavefront vertex, with the same velocity.
  - The edges of P are considered to be walls.





- P is **non-degenerate** if two or more motorcycles do not crash simultaneously into each other.

Theorem ([Cheng and Vigneron, 2002])

$\mathcal{M}(P)$ covers the reflex arcs of $\mathcal{S}(P)$.

Current situation

- **Known algorithms:**
  - $r \in O(n)$: number of reflex wavefront vertices.
  - Impl.: algorithm is suitable for implementation.

Alg.	Time	PSLG	Impl.
[Aichholzer et al., 1995]	$O(nr \log n)$	No	Yes
[Aichholzer and Aurenhammer, 1996]	$O(n^3 \log n)$	Yes	Yes
[Eppstein and Erickson, 1999]	$O(n^{1+\epsilon} + n^{8/11+\epsilon} r^{9/11+\epsilon})$	Yes	No
[Cheng and Vigneron, 2002]	exp. $O(n \log^2 n + r\sqrt{r} \log r)$	No	No

Current situation

- **Known algorithms:**
  - $r \in O(n)$: number of reflex wavefront vertices.
  - Impl.: algorithm is suitable for implementation.

Alg.	Time	PSLG	Impl.
[Aichholzer et al., 1995]	$O(nr \log n)$	No	Yes
[Aichholzer and Aurenhammer, 1996]	$O(n^3 \log n)$	Yes	Yes
[Eppstein and Erickson, 1999]	$O(n^{1+\epsilon} + n^{8/11+\epsilon} r^{9/11+\epsilon})$	Yes	No
[Cheng and Vigneron, 2002]	exp. $O(n \log^2 n + r\sqrt{r} \log r)$	No	No

- **Known implementations:**
  CGAL: for polygons with holes, quadratic runtime and memory usage.

Road map

Objective: Find an algorithm which is

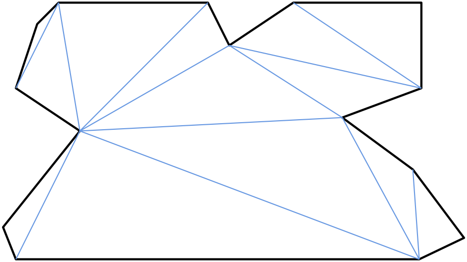
- fast in practice and
- simple enough to be implemented.

Approach:

- Algorithm by Aichholzer and Aurenhammer sounds promising.
- However, worst-case time complexity: $O(n^3 \log n)$.
- Open question: is the bound tight?

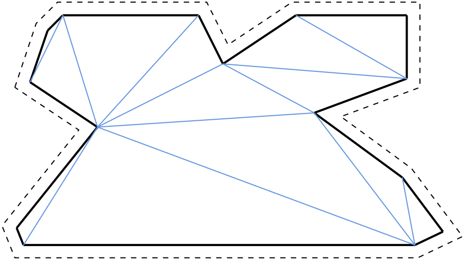
Triangulation-based approach

- Presented by [Aichholzer and Aurenhammer, 1996], handles PSLGs as input.



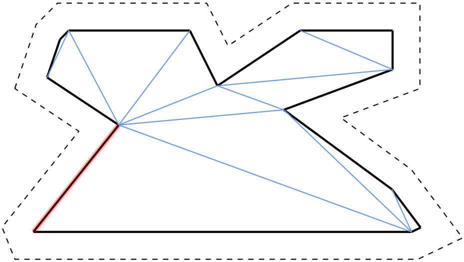
Triangulation-based approach

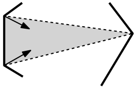
- Presented by [Aichholzer and Aurenhammer, 1996], handles PSLGs as input.



Triangulation-based approach

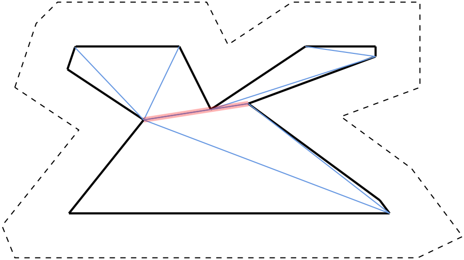
- Presented by [Aichholzer and Aurenhammer, 1996], handles PSLGs as input.
- Kinetic triangulation:
  - Edge events

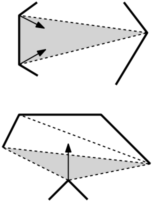




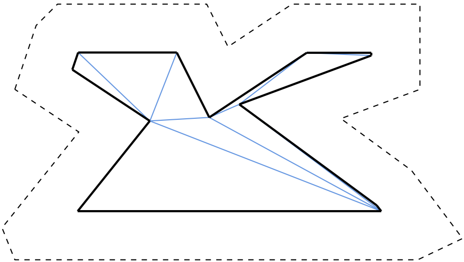
Triangulation-based approach

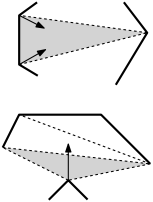
- Presented by [Aichholzer and Aurenhammer, 1996], handles PSLGs as input.
- Kinetic triangulation:
  - Edge events
  - Flip events





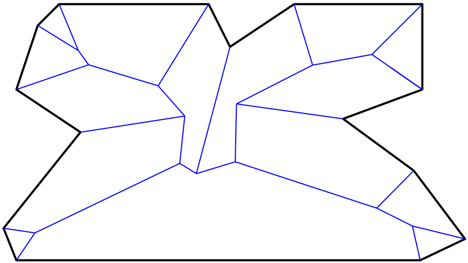
- Presented by [Aichholzer and Aurenhammer, 1996], handles PSLGs as input.
- Kinetic triangulation:
  - Edge events
  - Flip events

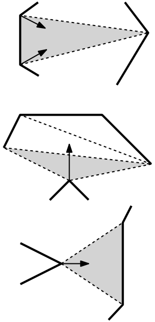




Triangulation-based approach

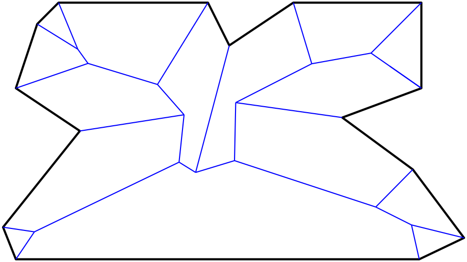
- Presented by [Aichholzer and Aurenhammer, 1996], handles PSLGs as input.
- Kinetic triangulation:
  - Edge events
  - Flip events
  - Split events

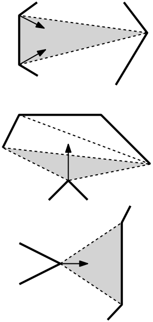




Triangulation-based approach

- Presented by [Aichholzer and Aurenhammer, 1996], handles PSLGs as input.
- Kinetic triangulation:
  - Edge events: $O(n)$
  - Flip events: $O(n^3)$
  - Split events: $O(n)$
  - Results in an $O(n^3 \log n)$ runtime.





Number of flip events

Number of flip events

Lemma ([Huber and Held, 2010b])

There exists polygons P with n vertices and triangulations T of P such that $\Omega(n)$ diagonals each reappear $\Omega(n)$ times.

Number of flip events

Lemma ([Huber and Held, 2010b])

There exists polygons P with n vertices and triangulations T of P such that $\Omega(n)$ diagonals each reappear $\Omega(n)$ times.

Lemma ([Huber and Held, 2010b])

There exist polygons P with n vertices, for which any triangulation leads to $\Omega(n^2)$ flip events.

Number of flip events

Lemma ([Huber and Held, 2010b])

There exists polygons P with n vertices and triangulations T of P such that $\Omega(n)$ diagonals each reappear $\Omega(n)$ times.

Lemma ([Huber and Held, 2010b])

There exist polygons P with n vertices, for which any triangulation leads to $\Omega(n^2)$ flip events.

Lemma ([Huber and Held, 2010b])

Every PSLG G with n vertices admits a triangulation that employs $O(n)$ Steiner points and is free of flip events.

Proof idea:

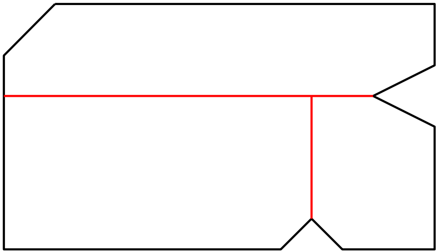
- Add the nodes and arcs of $\mathcal{S}(G)$ as Steiner points and triangulation diagonals.
- Show that the faces of $\mathcal{S}(G)$ can be triangulated such that flip events are avoided.
- Why does that work? Reflex wavefront vertices move on diagonals!

A novel wavefront-type algorithm

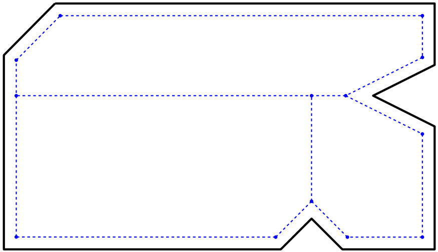
- Previous Steiner triangulations are based on $\mathcal{S}(G)$.
- Can we get rid of this requirement?

A novel wavefront-type algorithm

- Previous Steiner triangulations are based on $\mathcal{S}(G)$.
- Can we get rid of this requirement?
- Consider a non-degenerate polygon P as input:
  - Replacing $\mathcal{S}(P)$ by $\mathcal{M}(P)$.
  - Reflex arcs of $\mathcal{S}(P)$ are covered by $\mathcal{M}(P)$.

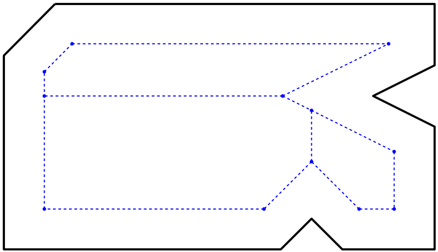


- Previous Steiner triangulations are based on $\mathcal{S}(G)$.
- Can we get rid of this requirement?
- Consider a non-degenerate polygon P as input:
  - Replacing $\mathcal{S}(P)$ by $\mathcal{M}(P)$.
  - Reflex arcs of $\mathcal{S}(P)$ are covered by $\mathcal{M}(P)$.
- Simulate the **extended wavefront**. [Huber and Held, 2010a]

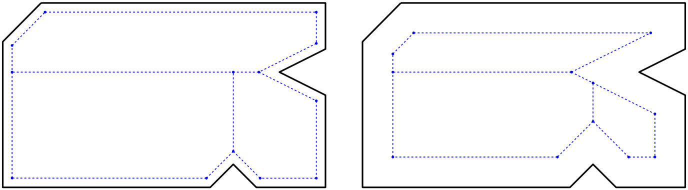


A novel wavefront-type algorithm

- Previous Steiner triangulations are based on $\mathcal{S}(G)$.
- Can we get rid of this requirement?
- Consider a non-degenerate polygon P as input:
  - Replacing $\mathcal{S}(P)$ by $\mathcal{M}(P)$.
  - Reflex arcs of $\mathcal{S}(P)$ are covered by $\mathcal{M}(P)$.
- Simulate the **extended wavefront**. [Huber and Held, 2010a]



A novel wavefront-type algorithm



Lemma ([Huber and Held, 2010a])

The faces of the extended wavefront are convex at any time.

- Hence, in the simulation, we only need to consider neighboring vertices.

Theorem ([Cheng and Vigneron, 2002])

$\mathcal{M}(P)$ *covers the reflex arcs of* $\mathcal{S}(P)$.

- Hence, split events happen within the motorcycle traces.

A novel wavefront-type algorithm

Works only for a restricted class of input data:

- No arbitrary polygons.
- No PSLGs.

Idea

Generalize $\mathcal{M}(P)$ of non-degenerate polygons P to $\mathcal{M}(G)$ of arbitrary PSLGs G!

A novel wavefront-type algorithm

Works only for a restricted class of input data:

- No arbitrary polygons.
- No PSLGs.

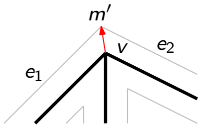
Idea

Generalize $\mathcal{M}(P)$ of non-degenerate polygons P to $\mathcal{M}(G)$ of arbitrary PSLGs G!

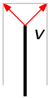
However:

- $\mathcal{M}(G)$ needs to cover all reflex arcs of $\mathcal{S}(G)$!
- The extended wavefront still needs to induce a convex tessellation!

- (a–b) Place a motorcycle for each reflex wavefront vertex in $\mathcal{W}(G, 0)$.

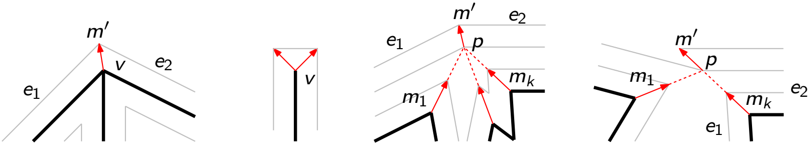


(a)

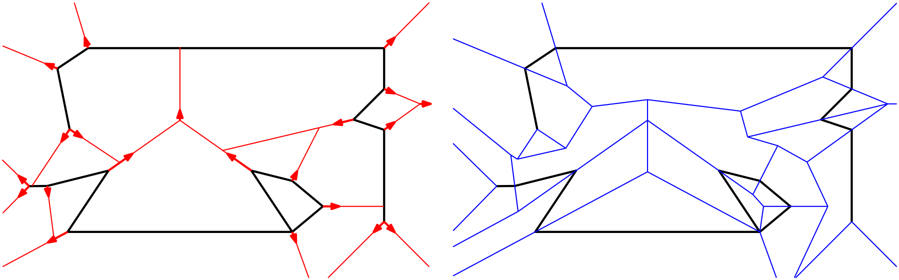


(b)

- (a–b) Place a motorcycle for each reflex wavefront vertex in $\mathcal{W}(G, 0)$.
- Launch a new motorcycle when multiple motorcycles meet:
  - (c) To cover reflex straight skeleton arcs.
  - (d) To respect the convex tessellation property.



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



Lemma ([Huber and Held, 2011c])

Consider a point p in the relative interior of $\mathcal{M}(G)$. Then a local disc is tessellated into convex slices by $\mathcal{M}(G)$.

Theorem ([Huber and Held, 2011c])

$\mathcal{M}(G)$ covers the reflex arcs of $\mathcal{S}(G)$.

- **A non-procedural alternative characterization of $\mathcal{S}(G)$:**
  - Generalizes a result of [Cheng and Vigneron, 2002].

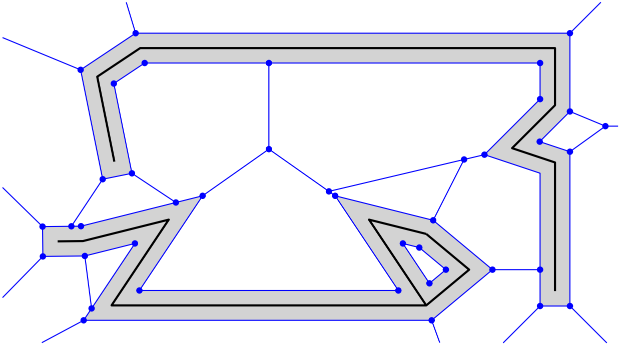
- **A non-procedural alternative characterization of $\mathcal{S}(G)$:**
  - Generalizes a result of [Cheng and Vigneron, 2002].
- **Computing $\mathcal{S}(G)$ using graphics hardware:**
  - Apply techniques used by [Hoff et al., 1999] for Voronoi diagrams.

Applications of the theorem

- **A non-procedural alternative characterization of $\mathcal{S}(G)$:**
  - Generalizes a result of [Cheng and Vigneron, 2002].
- **Computing $\mathcal{S}(G)$ using graphics hardware:**
  - Apply techniques used by [Hoff et al., 1999] for Voronoi diagrams.
- **A novel wavefront-type algorithm to compute $\mathcal{S}(G)$.**

A wavefront-type algorithm

We simulate the propagation of the extended wavefront:

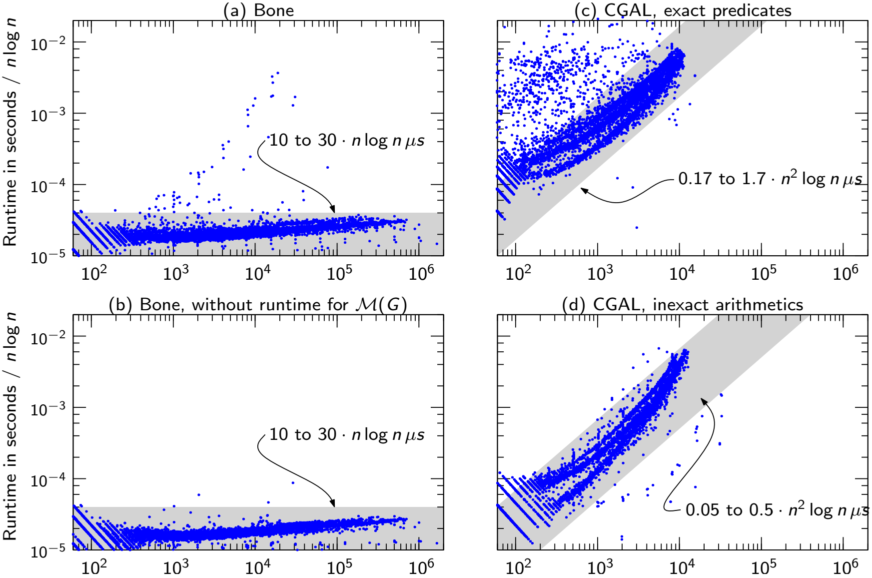


- Split events happen within the extended wavefront.
- Only neighboring vertices can meet.

BONE: a C++ implementation:

- Double-precision floating-point arithmetics.
- $O(n)$ space complexity.
- $O(n \log n)$ runtime in practice.
- $O(nr \log n)$ runtime in the worst case.
  - r is the number of reflex wavefront vertices.
  - Unlikely to occur in the real-world.

Runtime


(a) Bone
Runtime in seconds / $n \log n$
10^{-2}
10^{-3}
10^{-4}
10^{-5}
$10 \text{ to } 30 \cdot n \log n\ \mu s$
10^2 10^3 10^4 10^5 10^6
(c) CGAL, exact predicates
$0.17 \text{ to } 1.7 \cdot n^2 \log n\ \mu s$
(b) Bone, without runtime for $\mathcal{M}(G)$
$10 \text{ to } 30 \cdot n \log n\ \mu s$
(d) CGAL, inexact arithmetics
$0.05 \text{ to } 0.5 \cdot n^2 \log n\ \mu s$

Memory consumption

Random polygons generated by RPG.

Size n	**Bone** MB	**Bone** factor	**CGAL** MB	**CGAL** 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

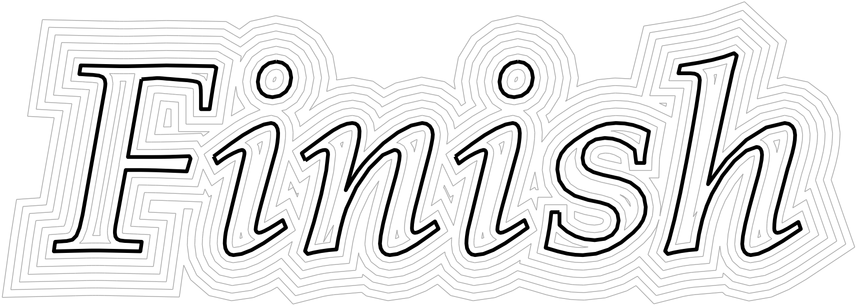
Table: Memory usage of Bone and CGAL

Summary

- Investigations of the triangulation-based algorithm:
  - Huber, S. and Held, M. (2010b). Straight skeletons and their relation to triangulations. In *Proc. 26th Europ. Workshop on Comp. Geom. (EuroCG '10)*, pages 189–192, Dortmund, Germany.
- Generalization of the motorcycle graph concept:
  - Alternative characterization of $\mathcal{S}(G)$.
  - Motivates an algorithm based on graphics hardware.
  - Huber, S. and Held, M. (2011c). Theoretical and practical results on straight skeletons of planar straight-line graphs. In *Proc. 27th Annual Symp. on Comp. Geom. (SoCG '11)*, pages 171–178, Paris, France.
- A novel wavefront-type algorithm and the implementation Bone:
  - Accepts arbitrary PSLGs as input.
  - The fastest and most memory-efficient implementation at the moment.
  - Huber, S. and Held, M. (2010a). Computing straight skeletons of planar straight-line graphs based on motorcycle graphs. In *Proc. 22nd Canad. Conf. on Comp. Geom. (CCCG '10)*, pages 187–190, Winnipeg, Canada.
  - Huber, S. and Held, M. (2011c). Theoretical and practical results on straight skeletons of planar straight-line graphs. In *Proc. 27th Annual Symp. on Comp. Geom. (SoCG '11)*, pages 171–178, Paris, France.

Summary: topics not covered in the talk

- MOCA: a motorcycle graph implementation:
  - Exploits geometric hashing, which is motivated by a stochastic analysis of the mean trace length.
  - MOCA: currently the only implementation which exhibits a subquadratic runtime in practice.
  - Huber, S. and Held, M. (2009). A practice-minded approach to computing motorcycle graphs. In *Proc. 25th Europ. Workshop on Comp. Geom. (EuroCG '09)*, pages 305–308, Brussels, Belgium.
  - Huber, S. and Held, M. (2011b). Motorcycle graphs: Stochastic properties motivate an efficient yet simple implementation. *ACM Journal on Exp. Alg.*, 17. in press.
- Straight skeletons approximating motorcycle graphs:
  - We are given n motorcycles. We can determine a G such that $\mathcal{S}(G)$ approximates $\mathcal{M}(m_1, \ldots, m_n)$.
  - We have an algorithm that computes $\mathcal{M}(m_1, \ldots, m_n)$ using $\mathcal{S}(G)$.
  - We obtained a proof for the P-completeness of straight skeletons of polygons with holes.
  - Huber, S. and Held, M. (2011a). Approximating a motorcycle graph by a straight skeleton. In *Proc. 23rd Canad. Conf. on Comp. Geom. (CCCG '11)*, Toronto, Canada. to appear.


Finish

Aichholzer, O. and Aurenhammer, F. (1996).
Straight Skeletons for General Polygonal Figures.
In *Proc. 2nd Annu. Internat. Conf. Comput. Combinatorics*, volume 1090 of *Lecture Notes Comput. Sci.*, pages 117–126. Springer-Verlag.

Aichholzer, O., Aurenhammer, F., Alberts, D., and Gärtner, B. (1995).
A novel type of skeleton for polygons.
Journal Universal Comp. Science, 1(12):752–761.

Cheng, S.-W. and Vigneron, A. (2002).
Motorcycle Graphs and Straight Skeletons.
In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 156–165, San Francisco, CA, USA.

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.

Hoff, K. et al. (1999).
Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware.
In *Comput. Graphics (SIGGRAPH '99 Proc.)*, pages 277–286, Los Angeles, CA.

Huber, S. and Held, M. (2009).
A practice-minded approach to computing motorcycle graphs.
In *Proc. 25th Europ. Workshop on Comp. Geom. (EuroCG '09)*, pages 305–308, Brussels, Belgium.

Huber, S. and Held, M. (2010a).
Computing straight skeletons of planar straight-line graphs based on motorcycle graphs.
In *Proc. 22nd Canad. Conf. on Comp. Geom. (CCCG '10)*, pages 187–190, Winnipeg, Canada.

Huber, S. and Held, M. (2010b).
Straight skeletons and their relation to triangulations.
In *Proc. 26th Europ. Workshop on Comp. Geom. (EuroCG '10)*, pages 189–192, Dortmund, Germany.

Bibliography III

Huber, S. and Held, M. (2011a).
Approximating a motorcycle graph by a straight skeleton.
In *Proc. 23rd Canad. Conf. on Comp. Geom. (CCCG '11)*, Toronto, Canada.
to appear.

Huber, S. and Held, M. (2011b).
Motorcycle graphs: Stochastic properties motivate an efficient yet simple implementation.
ACM Journal on Exp. Alg., 17.
in press.

Huber, S. and Held, M. (2011c).
Theoretical and practical results on straight skeletons of planar straight-line graphs.
In *Proc. 27th Annual Symp. on Comp. Geom. (SoCG '11)*, pages 171–178, Paris, France.