Computing Motorcycle Graphs Based on Kinetic Triangulations

Willi Mann[1] Martin Held[1] Stefan Huber[2]

[1]Dept. of Computer Science
University of Salzburg
Austria

[2]Dept. of Mathematics
University of Salzburg
Austria

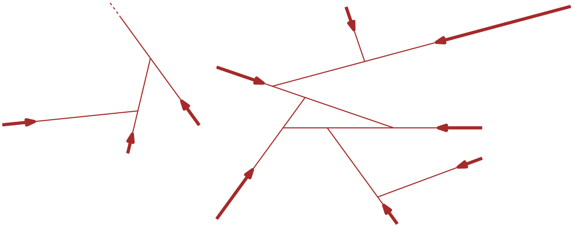
CCCG 2012 in Charlottetown, Canada
August 8–10

Introduction to motorcycle graphs

A **motorcycle** is a point that moves with constant velocity.

- It has a start point.
- It leaves a **trace** behind it.
- It stops moving (**crash**) when reaching another's trace.

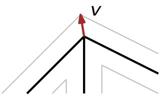
The **motorcycle graph** $\mathcal{M}(m_1, \ldots, m_n)$ of the motorcycles $m_1, \ldots, m_n$ is the arrangement of their traces.



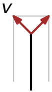
Motorcycle graph induced by a PSLG I

We are given a planar straight-line graph G.

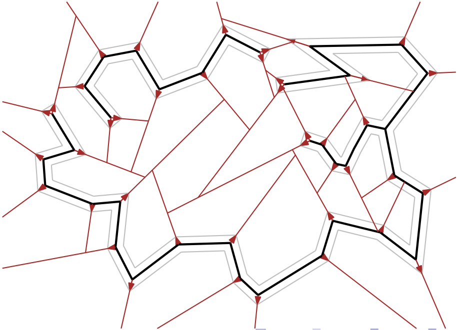
- All edges of G are considered to be solid walls: motorcycles crash against them.
- Motorcycles induced by G:
  1. We consider the straight-skeleton wavefront $\mathcal{W}_G(\epsilon)$ of G — i.e., mitered offset — for a small time ϵ.
     - For each reflex vertex v in $\mathcal{W}_G(\epsilon)$ we define a motorcycle m.
     - m has the same start point and velocity as v.



(a)

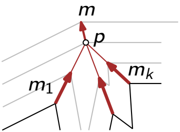


(b)

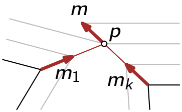


Motorcycle graph induced by a PSLG II

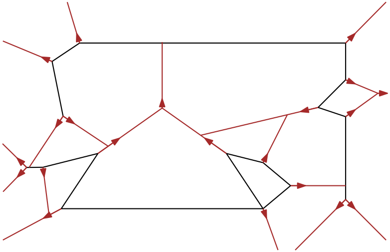
2. Assume that $m_1, \ldots, m_k$ simultaneously crash into each other at the location $p \in \mathbb{R}^2$ and time t.
   - If, in a local neighborhood of p, all traces until time t lie in a half plane then we **launch a new motorcycle** m at p.



(a)



(b)



- The **motorcycle graph** $\mathcal{M}(G)$ **induced by** G is defined as the arrangement of all traces.

Applications

Strong algorithmic and geometric relationship to straight skeletons:

- Non-procedural characterization of straight skeletons.
- Straight-skeleton algorithms based on motorcycle graphs: [Huber and Held, 2011b], [Cheng and Vigneron, 2007]
- $\mathcal{P}$-completeness of straight skeletons.

Further related applications:

- Repetitive ray shooting-and-insertion algorithm: [Ishaque et al., 2009]
- Art-gallery algorithm related to motorcycle graphs: [Czyzowicz et al., 1989]
- Motorcycle graphs on quadrilateral meshes: [Eppstein et al., 2008]

Prior Work

- **Problem introduced by [Eppstein and Erickson, 1999].**
  - $O(n^{17/11+\epsilon})$ time and space algorithm.
  - Not suitable for implementation.
- **Best worst-case time complexity by [Cheng and Vigneron, 2007].**
  - $O(n\sqrt{n}\log n)$ time complexity.
  - Uses $1/\sqrt{n}$-cuttings.
  - Needs to know all motorcycles a-priori. Cannot compute generalized motorcycle graph. Not suitable for implementation.
- **Practical approach by [Huber and Held, 2011a].**
  - Implementation MOCA uses $\sqrt{n} \times \sqrt{n}$ geometric hash.
  - Stochastic motivation: motorcycles cross $O(1)$ grid cells on average provided that they are distributed uniformly enough.
    - $O(n \log n)$ runtime in practice.
    - Used by our straight-skeleton code BONE [Huber and Held, 2011b].
  - However, for contrived input configurations — e.g., densely sampled convex bodies — it requires up to $O(n^2\sqrt{n}\log n)$ time.

Our algorithm

Our algorithm is split into two steps:

- The first step computes $\mathcal{M}(G)$ inside the convex hull of G.
  - Based on kinetic triangulations.
- The second step computes $\mathcal{M}(G)$ outside the convex hull of G.
  - A plane-sweep algorithm.

Computing $\mathcal{M}(G)$ inside conv G

Basic idea: Consider a kinetic triangulation T such that

- each motorcycle is a moving vertex, and
- each crash is indicated by a topological change, i.e., a collapse of a triangle.

Therefore,

- each wall shall be an edge of T, and
- for each motorcycle m its trace shall be an (growing) edge of T.

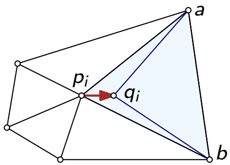
In contrast to inflexible and static geometric hashing,

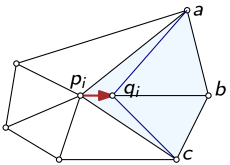
- kinetic triangulations use more triangles at regions of higher density, and
- they adapt over time.

Initial triangulation

The initial triangulation is obtained as follows:

1. Compute a constrained triangulation T of G within conv G.
2. Each initially present motorcycle m_i starts from a vertex p_i of G.
   - Make a duplicate q_i of p_i.
   - q_i models the moving motorcycle m_i with velocity $v_i \in \mathbb{R}^2$ and start point p_i.
3. Merge q_i to T:





Simulation of the kinetic triangulation

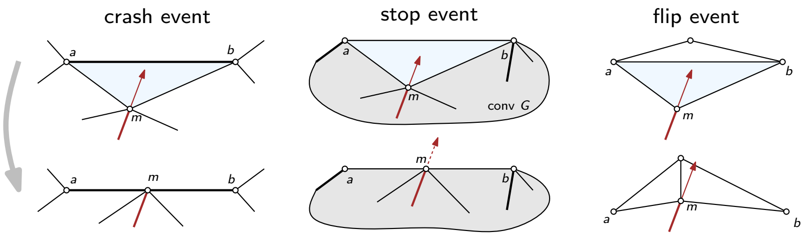
Basic algorithm:

- Obtain initial triangulation
- For every triangle
  - compute the collapse time (root of a quadratic polynomial) and
  - add a collapse event into a chronological priority queue Q.
- Until Q is empty:
  - Fetch the next event,
  - adapt the triangulation and possibly add new events.

The algorithm is finished when no moving vertex remains.

Event types

- **Crash event:** a motorcycle m reached a wall or a trace (or another motorcycle).
  - Remove collapsed triangle, reschedule triangles incident to m.
  - If necessary launch a new motorcycle.
- **Stop event:** a motorcycle reached an edge of conv G.
  - Similar to a crash event, but motorcycle will resume in the second phase of the algorithm.
- **Flip event:** a motorcycle reached an edge that is not a wall, a trace or an edge of conv G.
  - Reschedule the two involved triangles.

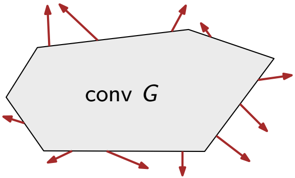


Computing $\mathcal{M}(G)$ outside conv G

We are given

- a set of motorcycles that started within conv G and were stopped at bd conv G and
- a set of motorcycles that start at bd conv G.

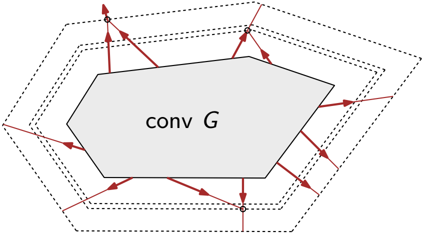
In either case, all motorcycles reside on bd conv G and head for $\mathbb{R}^2 \setminus$ conv G.



Sweep-plane algorithm

Outline of the basic algorithm:

- We expand the convex polygon $P = \text{conv}\, G$ by moving P's edges outwards, in parallel and at unit speed.
- Maintain the intersection of the expanding P with $\mathcal{M}(G)$.
  - Motorcycles are held in a doubly-linked circular list L.
- If two neighboring motorcycles switch positions on P then one crashed into the other.



Complexity Analysis

Inside the convex hull:

- Setup of initial triangulation and filling the priority queue: $O(n \log n)$ time.
- Handling the flip events:
  - A single flip event requires to reschedule two triangles: $O(\log n)$ time.
- Handling a crash/stop event:
  - Remove the collapsed triangle.
  - Reschedule all triangles incident to the moving vertex.
    - f flip events increase the sum of degrees of all moving vertices by at most $2f$.
    - All crash/stop events require in total $O(n + 2f)$ reschedules.

Outside the convex hull:

- Setup of the priority queue: $O(n \log n)$ time.
- Each crash requires amortized $O(\log n)$ time.
  - Only local modifications of the circular list.

Lemma

The overall complexity is $O((n + f) \log n)$, where f is the number of flip events.

Number of flip events

Best known upper bound for f is $O(n^3)$:

- No example known that exceeds $O(n^2)$ flip events.
- Runtime tests show that $f \in O(n)$ in practice.

Can we employ Steiner vertices to reduce f?

Steiner triangulations

Lemma

We can place Steiner vertices such that all flip events vanish.

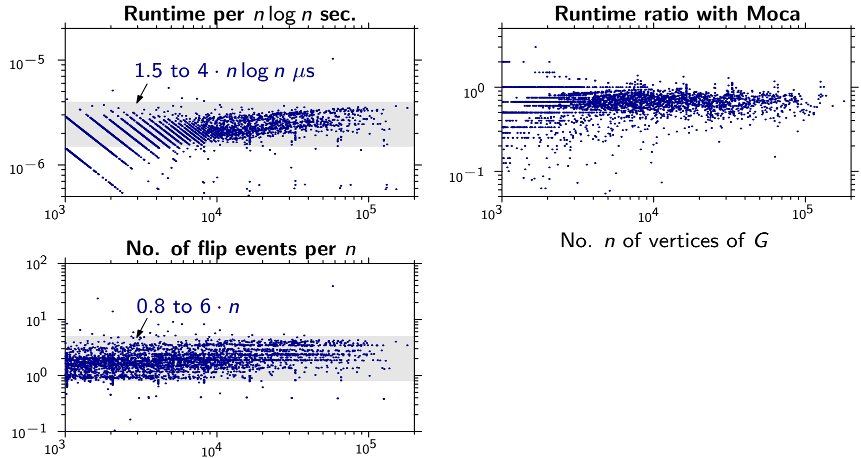
Unfortunately, proof uses $\mathcal{M}(G)$ to obtain such a triangulation.

- However, it is worth to look for suitable Steiner triangulations!

Skipped: Heuristics to insert Steiner edges on the motorcycle's traces. Number of flip events reduces by 20 %. However, no gain in runtime performance.

Experimental results

- Our implementation is written in C++, using Triangle by [Shewchuk, 1996].
  - Double-precision floating-point arithmetics.
  - MPFR support built-in. A slow-down by a factor of 25 with a prec. of 212 bits.



Summary

- Currently fastest implementation, simple-to-implement algorithm.
- Plane-sweep algorithm time-optimal outside conv G.
- Steiner triangulations are a promising approach to reduce number of flip events. (Future work)
- **Skipped in the talk:** Robust handling of concurrent events, i.e., to avoid infinite loops of concurrent events.

Bibliography I

Cheng, S.-W. and Vigneron, A. (2007).
Motorcycle graphs and straight skeletons.
Algorithmica, 47(2):159–182.

Czyzowicz, J., Rival, I., and Urrutia, J. (1989).
Galleries, light matchings and visibility graphs.
In *Proc. 1st Workshop Alg. Data Struct. (WADS '89)*, pages 316–324, Ottawa, Canada. Springer.

Eppstein, D. and Erickson, J. (1999).
Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions.
Discrete Comp. Geom., 22(4):569–592.

Eppstein, D., Goodrich, M. T., Kim, E., and Tamstorf, R. (2008).
Motorcycle graphs: Canonical quad mesh partitioning.
Computer Graph. Forum, 27(5):1477–1486.

Huber, S. and Held, M. (2011a).
Motorcycle graphs: Stochastic properties motivate an efficient yet simple implementation.
ACM J. on Exp. Alg., 16:1.3:1.1–1.3:1.17.

Bibliography II

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

Ishaque, M., Speckmann, B., and Tóth, C. D. (2009).
Shooting permanent rays among disjoint polygons in the plane.
In *Proc. 25th ACM Symp. on Comp. Geom. (SoCG '09)*, pages 51–60, New York, NY, USA. ACM.

Shewchuk, J. R. (1996).
Triangle: Engineering a 2D quality mesh generator and delaunay triangulator.
In Lin, M. C. and Manocha, D., editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lect. Notes in Comp. Sci.*, pages 203–222. Springer Berlin.

Steiner triangulations

Lemma

We can place Steiner vertices such that all flip events vanish.

Proof sketch:

- The overlay $G + \mathcal{M}(G)$ tessellates $\mathbb{R}^2$ into convex cells.
  - Add $\mathcal{M}(G)$ as Steiner vertices and constrained edges.
  - Triangulate each convex cell arbitrarily.
- All motorcycles drive along Steiner edges.
  - Hence, no flip events occur.

Unfortunately, we do not know $\mathcal{M}(G)$ in advance.

- However, it is worth to look for suitable Steiner triangulations!

Heuristics for Steiner triangulations

We applied two heuristics:

1. We exploit the average trace length, cf. [Huber and Held, 2011a].
   - Insert Steiner edges with length $c/\sqrt{n}$ for some constant $c > 0$.
2. For $c\sqrt{n}$ randomly chosen motorcycles insert their infinite track as Steiner edges.

In both heuristics, Steiner edges are restricted in length

- if they reach a wall, or
- if they reach the convex hull of G.

Intersections among Steiner edges are resolved by splitting them by Steiner vertices.

- One expects $O(n)$ intersection points at most.

Experiments: Number of flip events reduced by 20 %. However, the costs of the flip events saved do not outweigh the preprocessing costs.