A Wavefront-Like Strategy for Computing
Multiplicatively Weighted Voronoi Diagrams

Martin Held\textsuperscript{1}  Stefan de Lorenzo\textsuperscript{1}
\textsuperscript{1}University of Salzburg, Department of Computer Science

March 19, 2019
Problem Specification

Given:

A set $S$ of $n$ input points in the plane, where every $s \in S$ is associated with a real-valued weight $w(s) > 0$. 

Computational Geometry and Applications Lab
UNIVERSITÄT SALZBURG

Computational Geometry and Applications Lab
UNIVERSITÄT SALZBURG
Problem Specification

Given:

A set $S$ of $n$ input points in the plane, where every $s \in S$ is associated with a real-valued weight $w(s) > 0$. 

Computational Geometry and Applications Lab

UNIVERSITÄT SALZBURG
Problem Specification

Given:
A set $S$ of $n$ input points in the plane, where every $s \in S$ is associated with a real-valued weight $w(s) > 0$.

Compute:
The multiplicatively weighted Voronoi diagram (MWVD) $\mathcal{VD}_w(S)$ of $S$. 
Multiplicatively Weighted Voronoi Diagrams

- The Voronoi edges are formed by straight-line segments and circular arcs.
Multiplicatively Weighted Voronoi Diagrams

- The Voronoi edges are formed by straight-line segments and circular arcs.
- The Voronoi regions are (possibly) multiply connected.
- The MWVD has a quadratic combinatorial complexity in the worst case.
We present a wavefront-based approach for computing MWVDs. The wavefront covers an increasing portion of the plane over time. It consists of wavefront arcs and wavefront vertices. Whenever a wavefront arc vanishes or spawns, a new Voronoi node is discovered.
• Every site is associated with an offset circle.
• Every site is associated with an offset circle.
• Every site is associated with an offset circle.
• Two moving intersection points trace out the bisector as time progresses.
• Every site is associated with an offset circle.
• Two moving intersection points trace out the bisector as time progresses.
• Inactive arcs along the offset circles are eliminated.
• The active arcs are stored in sorted angular order.
• Every site is associated with an offset circle.
• Two moving intersection points trace out the bisector as time progresses.
• Inactive arcs along the offset circles are eliminated.
• The active arcs are stored in sorted angular order.
• Every site is associated with an **offset circle**.
• Two *moving intersection points* trace out the bisector as time progresses.
• *Inactive arcs* along the offset circles are eliminated.
• The *active arcs* are stored in sorted angular order.
• Every site is associated with an offset circle.
• Two moving intersection points trace out the bisector as time progresses.
• Inactive arcs along the offset circles are eliminated.
• The active arcs are stored in sorted angular order.
• The wavefront is formed by parts of offset circles.
• A wavefront vertex is a specific intersection point of two offset circles.
• Active arcs do not necessarily coincide with wavefront arcs.
Event Handling

- **Collision** and **domination events** mark the initial and last contact (of a pair of offset circles), respectively.

- **Edge** and **break-through events** happen whenever active arcs vanish or spawn.

- These events are stored in a priority queue $Q$. 
• **Collision** and **domination events** mark the initial and last contact (of a pair of offset circles), respectively.

• **Edge** and **break-through events** happen whenever active arcs vanish or spawn.

• These events are stored in a priority queue $Q$. 

![Diagram showing event handling in Computational Geometry and Applications Lab, Universit"at Salzburg.](image-url)
Event Handling

- **Collision** and *domination events* mark the initial and last contact (of a pair of offset circles), respectively.
- **Edge** and *break-through events* happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $Q$. 
• **Collision** and **domination events** mark the initial and last contact (of a pair of offset circles), respectively.

• **Edge** and **break-through events** happen whenever active arcs vanish or spawn.

• These events are stored in a priority queue $Q$. 
Event Handling

- **Collision** and **domination events** mark the initial and last contact (of a pair of offset circles), respectively.
- **Edge** and **break-through events** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $Q$. 
Event Handling

- **Collision** and **domination events** mark the initial and last contact (of a pair of offset circles), respectively.
- **Edge** and **break-through events** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $Q$. 

![Diagram showing event handling in Computational Geometry and Applications Lab at Universität Salzburg]
• *Collision* and *domination events* mark the initial and last contact (of a pair of offset circles), respectively.

• *Edge* and *break-through events* happen whenever active arcs vanish or spawn.

• These events are stored in a priority queue $Q$. 
Event Handling

- **Collision** and **domination events** mark the initial and last contact (of a pair of offset circles), respectively.
- **Edge** and **break-through events** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $Q$. 
• **Collision** and **domination events** mark the initial and last contact (of a pair of offset circles), respectively.

• **Edge** and **break-through events** happen whenever active arcs vanish or spawn.

• These events are stored in a priority queue $Q$. 
Event Handling

- **Collision** and **domination events** mark the initial and last contact (of a pair of offset circles), respectively.
- **Edge** and **break-through events** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $Q$. 
Event Handling

- Initially, the collisions of every pair of offset circles are computed.
- Afterwards, the events are successively popped from $Q$.
- A collision event is **valid** if the collision point is situated within active arcs.
Event Handling

- Initially, the collisions of every pair of offset circles are computed.
- Afterwards, the events are successively popped from $Q$.
- A collision event is **valid** if the collision point is situated within active arcs.
- On each edge and break-through event, three offset circles need to be updated.
- The angular order of active arcs only changes at events.
Wavefront Propagation
Wavefront Propagation
Wavefront Propagation
Wavefront Propagation
Wavefront Propagation
Wavefront Propagation
Wavefront Propagation
Wavefront Propagation
• All topological changes of the wavefront are properly detected.
• A quadratic number of collision events are computed in any case.
• In the worst case $O(n^2)$ break-through events take place.
• At most $O(n^2)$ active arcs are generated during the wavefront propagation.
• Therefore, the algorithms runtime is $O(n^2 \log n)$ in the worst case.
• Our algorithm is also able to handle additive weights.
Discussion

- Our algorithm is also able to handle additive weights.
- Improve the average case behavior of our strategy.
- Reduce the number of collisions that are computed.
Thank you for your attention!