Sweeping Circular Arcs

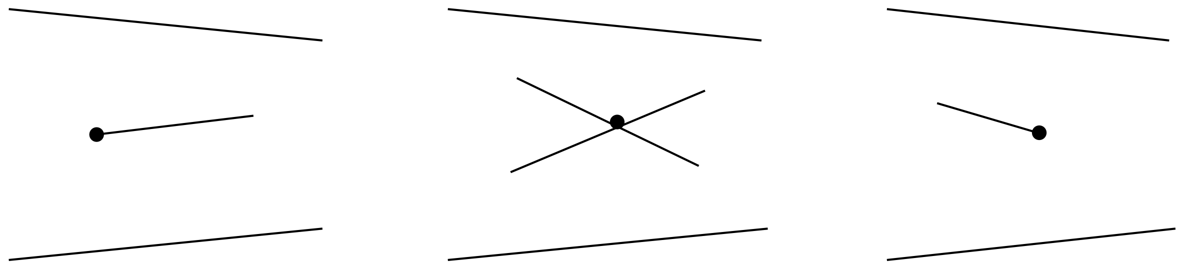
Susan Hert and Kurt Mehlhorn

Max-Planck-Institute for Computer Science

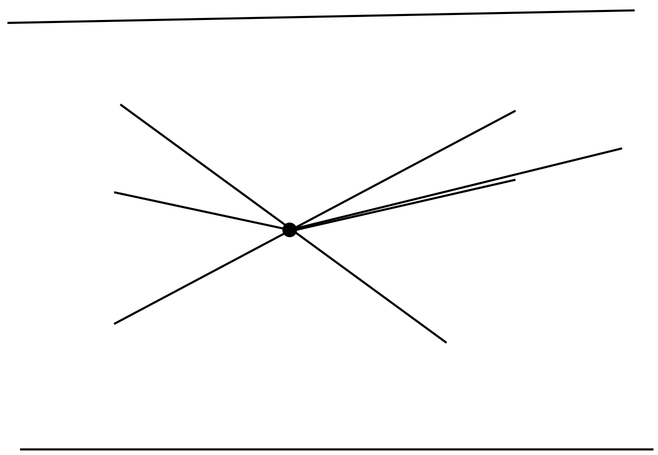
- Bentley-Ottman Sweep for Line Segments
- Implementation
- Extension to Circular Arcs
- A First Implementation
- Demo
- Optimizations

Bentley-Ottman Sweep

- with general position assumption



- for arbitrary inputs (cf. de Berg et al and LEDAbook)



remark: embedding

Geometric Primitives

- lex-order of endpoints and intersection points
- orientation of three points
- point above (on, below) segment
- overlap of segments
- slope compare
- compute intersection

- second and third predicate reduce to sign of determinants

Implementations

- LEDA, see LEDAbook, Section 10.7
- CGAL, see Michael Seel, PhD thesis
  - improved embedding
  - CGALization

Applications

- Gen-Polygons of LEDA, see LEDAbook, Section 10.8

  halfspaces and regular boolean operations
- Nef-Polygons of CGAL, see Michael Seel, PhD thesis

  halfspaces, boolean operations, topological operations (int, cl, bd)

Our Goal

extend the above to polygons with straight-line and circular arcs

Circle Points and Circular Arcs

- Rational Circle = circle with rational center and rational square radius
- Rational Line = line with rational coefficients
- Circle Point = intersection of rational circle with rational line
- Circular Arc = arc on a rational circle delineated by circle points

Lemma 1 *Intersection points of circular arcs are circle points*

Proof: subtraction of the equations of two rational circles yields equation of rational line. ∎

Extension of Sweep Algorithm

- circular arcs may intersect twice
- treatment of starting and passing segment slightly more involved

A First Implementation

- Seel's generic sweep
- CGAL kernel
- additional point and segment classes: circle points and circle arcs
- primitives for arcs and circle points:
  - intersection of two circular arcs, intersection circular and straight arc,
  - slope order, point above segment, . . . , always several cases
  - Question: can one (should one) treat straight segments as circular arcs??
- number type: LEDA real

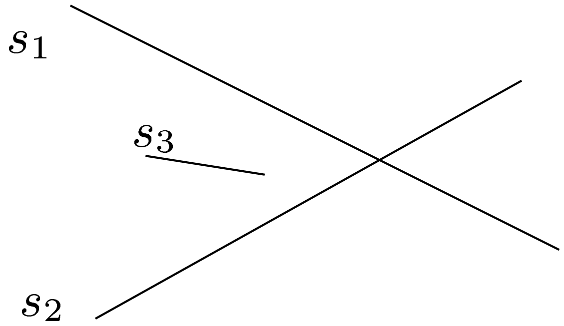
A First Implementation

- Seel's generic sweep
- CGAL kernel
- additional point and segment classes: circle points and circle arcs
- primitives for arcs and circle points:
  - intersection of two circular arcs, intersection circular and straight arc,
  - slope order, point above segment, . . . , always several cases
  - Question: can one (should one) treat straight segments as circular arcs??
- number type: LEDA real

program swept two arcs in 10 minutes

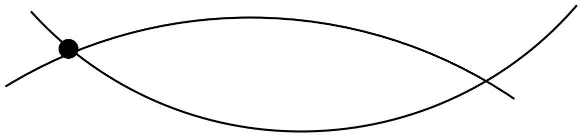
Optimization I: Prefer Combinatorics over Numerics

- dictionary for intersections



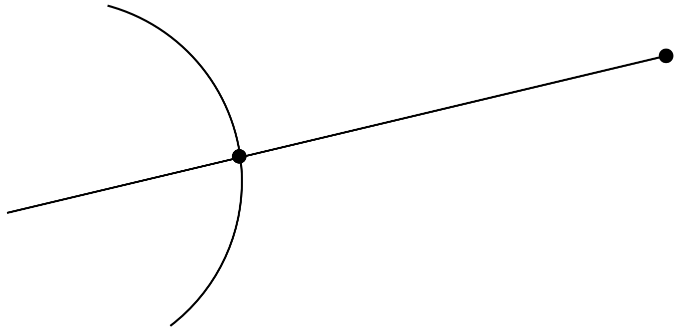
  - $s_1 \cap s_2$ is recomputed and reinserted into X-struct after s_3 is swept
  - it is better to store it in a dictionary (under key (s_1, s_2))
  - optimization is used in LEDAsweep and generic sweep, but now it really pays

- fast equality test for identically constructed points



  - assume sweep point is at left intersection of circular arcs
  - at this point: we compare the sweep point to itself
  - this is extremely costly when done numerically

Optimization II: Retain Geometric Information



- default implementation of segments: store endpoints
- when endpoints are circle points it is very costly to recover the underlying line
- better implementation: store underlying line and the endpoints
- generalizes to arbitrary curves: store underlying curve and two curve points

Conclusion:
- reconsider kernel design
- coordinate-centered versus geometry-centered

Optimization III: Use Heavy Machinery Sparingly

- use reals only if rationals do not suffice
- first implementation used *real*s as the default number type
- improved implementation uses *rational*s whenever possible.
- switches to *real*s only when square roots are involved
- Question: can *real*s do this automatically?

Running Times

10 curves (5 segments, 5 circle arcs), 46 vertices, 112 edges

version	time in second
without fast equal, without dictionary	53.0000
without fast equal, with dictionary	42.0000
with fast equal, all real	6.5000
with fast equal, thrifty arithmetic	0.8000
with fast equal, only line segments	0.0300
LEDA sweep, only segments, filtered predicates	0.0035

What Next

- O. Devillers and A. Fronville and B. Mourrain and M. Teillaud: Algebraic methods and arithmetic filtering for exact predicates on circle arcs
  does it help?
- relate to work in Tel Aviv
- Mulmuley's randomized incremental algorithm instead of sweep:
  for segments, the sweep is slightly faster if the number of intersections is small
- further optimization
- circle points and circle arcs as CGAL/LEDA extension package.
- polgons and boolean operations on them:
  - generalized polygons of LEDA: implementation should generalize without much difficulty
  - Nef polygons of CGAL: that's more complicated and we dare no prediction at this moment