CGAL's Arrangement Package and Derivatives

Efi Fogel, Dan Halperin, Idit Haran, Michal Meyerovitch, Ron Wein, and Baruch Zukerman[1]

[1]Tel Aviv University, Israel

with contributions from the following sites:

[2]Max-Planck-Institut für Informatik, Saarbrücken, Germany

[3]INRIA Sophia Antipolis, France

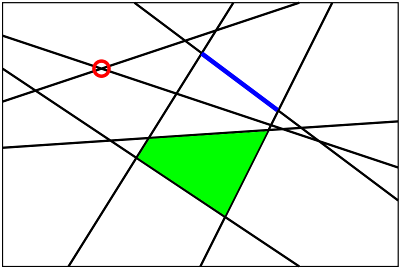
LIAMA 10th Anniversary, 2007

Outline

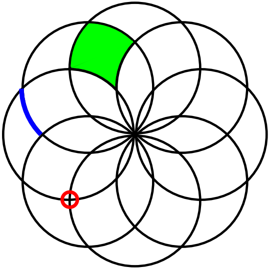
1. Arrangements
2. Boolean Set Operations
3. Envelopes
4. Minkowski Sums

Arrangement Definition

Given a collection Γ of planar curves, the arrangement $\mathcal{A}(\Gamma)$ is the partition of the plane into vertices, edges and faces induced by the curves of Γ



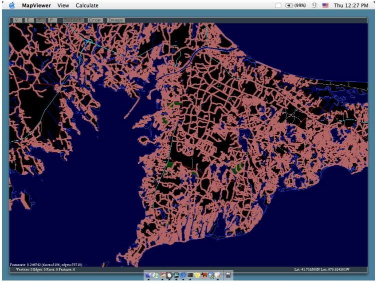
An arrangement of lines

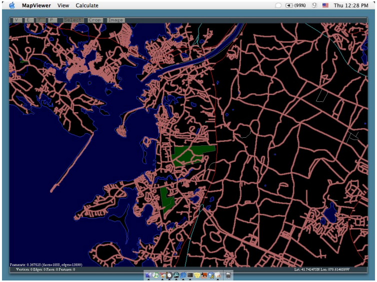


An arrangement of circles

Arrangement Background

- Arrangements have numerous applications
  - robot motion planning, computer vision, GIS, optimization, computational molecular biology





A planar map of the Boston area showing the top of the arm of cape cod.

Raw data comes from the US Census 2000 TIGER/line data files.

The `Arrangement_2` Package

- Constructs, maintains, modifies, traverses, queries, and presents subdivisions of the plane
- Robust and exact
  - All inputs are handled correctly (including degenerate)
  - Exact number types are used to achieve exact results
- Efficient
- Generic, easy to interface, extend, and adapt
- Part of the CGAL basic library

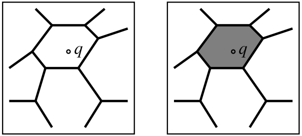
Arrangement Traits Classes

- Define the family of curves
- Aggregate geometric types and operations over the types

Curve Family	Arithmetic	Boundness	Class Name
linear segments	rational	bounded	Arr_non_caching_segment_traits_2 Arr_segment_traits_2
linear segments, rays, lines	rational	bounded	Arr_linear_traits_2
piecewise linear curves	rational	bounded	Arr_polyline_traits_2
circular arcs, linear segments	rational	bounded	Arr_circle_segment_traits_2 Arr_circular_line_arc_traits_2
algebraic curves degree ≤ 2	algebraic	bounded unbounded	Arr_conic_traits_2 Arr_conix_traits_2
algebraic curves degree ≤ 3	algebraic	unbounded	Arr_cubix_traits_2
algebraic curves degree ≤ 4	algebraic	unbounded	Arr_quadrix_traits_2
planar Bézier curves	algebraic	unbounded	Arr_Bezier_curve_traits_2
rational function arcs	algebraic	unbounded	Arr_rational_arc_traits_2

Arrangement Point Location

Given a subdivision A of the space into cells and a query point q, find the cell of A containing q



- Fast query processing
- Reasonably fast preprocessing
- Small space data structure

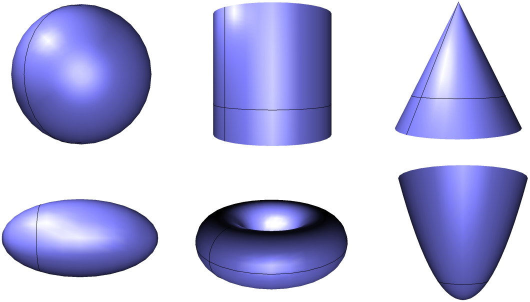
	Naive	Walk	RIC	Landmarks	Triangulat	PST
Preprocess time	none	none	$O(n \log n)$	$O(k \log k)$	$O(n \log n)$	$O(n \log n)$
Memory space	none	none	$O(n)$	$O(k)$	$O(n)$	$O(n \log n)^{(*)}$
Query time	bad	reasonable	good	good	quite good	good
Code	simple	quite simple	complicated	quite simple	modular	complicated

Walk — Walk along a line **RIC** — Random Incremental Construction based on trapezoidal decomposition
Triangulat — Triangulation **PST** — Persistent Search Tree
k — number of landmarks
(*) Can be reduced to $O(n)$

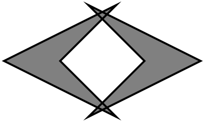
Arrangements on Surfaces — A Glimpse to the Future

A direction for future development is to extend the package to support arrangements on continuous two-dimensional parametric surfaces, which may

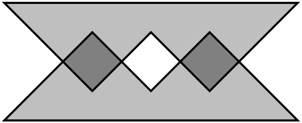
- be unbounded
- contain curves of discontinuity
- contain singularity points



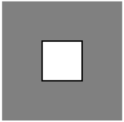
Boolean Set Operations



Union



Intersection



Complement

For two point sets P and Q and a point r:

Complement	$R = \overline{P}$	
Union	$R = P \cup Q$	
Intersection	$R = P \cap Q$	$= \overline{\overline{P} \cup \overline{Q}}$
Difference	$R = P \setminus Q$	$= P \cap \overline{Q}$
Symmetric Difference	$R = (P \setminus Q) \cup (Q \setminus P)$	
Intersection predicate	$P \cap Q \neq \emptyset$	Overlapping cell(s) are not explicitly computed
Containment predicate	$r \in P$	
Interior, Boundary, Closure		
Regularization	$R = \text{closure}(\text{interior}(P))$	

The `Nef_polyhedron_{2,S2,3}` Packages

A Nef polyhedron is obtained from a finite set of open halfspaces by set complement and set intersection operations.

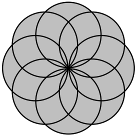
- Supports the construction of Nef polyhedron

  `Nef_polyhedron_2` — in $\mathbb{R}^2$
  `Nef_polyhedron_S2` — embedded on $\mathbb{S}^2$
  `Nef_polyhedron_3` — in $\mathbb{R}^3$
- Supports
  - Boolean set operations
  - interior, boundary, and closure operations
    - composed operations, e.g., regularization
- Robust and efficient

The `Boolean_set_operation_2` Package

- Supports
  - regularized Boolean set-operations
  - intersection predicates
  - point containment predicates
- Operands and results are regularized point sets bounded by x-monotone curves referred to as general polygons
  - General polygons may have holes
- Based on the `Arrangement_2` and `Polygon_2` packages
- Robust and efficient
  - Extremely efficient aggregated operations

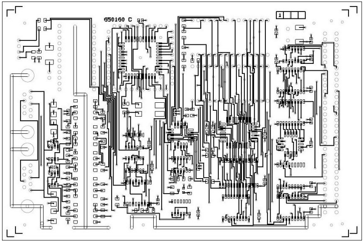
The union of eight unit discs

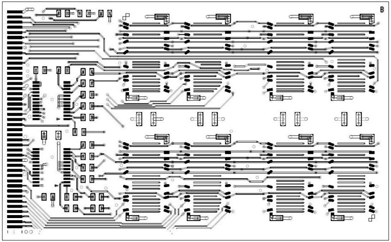


VLSI: Union of PCB Components

Input File	No. of Polygons	No. of Circles	Union Size			Time) (sec.)
			V	E	F	
VLSI_1	2593	645	13130	13130	614	3.50
VLSI_2	22406	294	14698	14698	357	24.70

V — no. of vertices **E** — no. of edges **F** — no. of faces



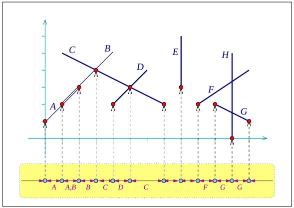


Courtesy of Maniabarco Inc.

Envelopes in $\mathbb{R}^2$

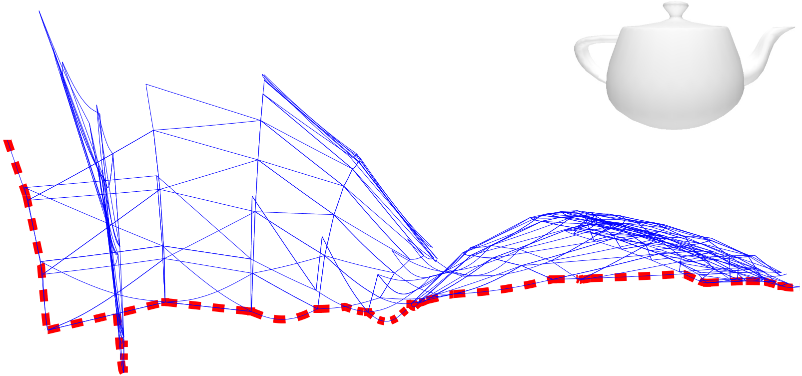
Given a set of x-monotone curves $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$, their lower envelope (resp. upper envelope) is the point-wise minimum (resp. maximum) of all curves.

The minimization diagram (resp. maximization diagram) of $\mathcal{C}$ is the subdivision of the x-axis into cells, such that the identity of the curves that induce the lower (resp. upper) envelope over a specific cell of the subdivision (an edge or a vertex) is the same.



- The minimization diagram of 8 line segments
- Each diagram vertex points to the point associated with it
- Each diagram edge is marked with segments that induce it

Computer Aided Manifacturing

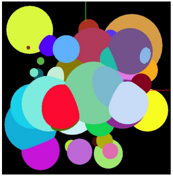
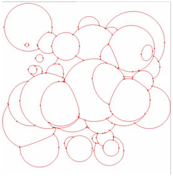
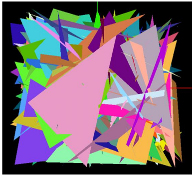
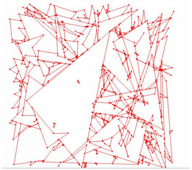


The lower envelope of a set of line segments and hyperbolic arcs obtained by the radial projection of a triangulated Utah teapot.

Envelopes in $\mathbb{R}^3$

Given a set of *xy*-monotone surfaces $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, their lower envelope (resp. upper envelope) is the point-wise minimum (resp. maximum) of all surfaces.

The minimization diagram (resp. maximization diagram) of $\mathcal{S}$ is the subdivision of the *xy*-plane into cells, such that the identity of the surfaces that induce the lower (resp. upper) diagram over a specific cell of the subdivision (a face, an edge, or a vertex) is the same.

The `Envelope_3` Package

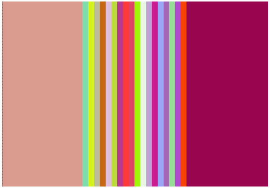
- Constructs lower and upper envelopes of surfaces

Surface Family	Class Name	
triangles	`Env_triangle_traits_3`	
spheres	`Env_sphere_traits_3`	
planes and half planes	`Env_plane_traits_3`	
quadrics	`Env_quadric_traits_3`	mpii

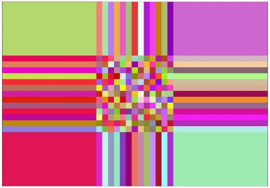
- Based on the `Arrangement_2` package
- Exploits
  - Overlay computation (using the sweep line framework)
  - Isolated points
  - Zone traversal
- Robust and efficient

Voronoi Diagrams of Points in the Plane

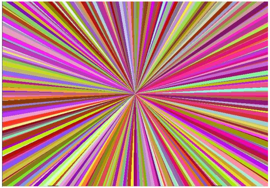
- Computed as upper envelopes of planes
- Represented as planar arrangements of unbounded curves



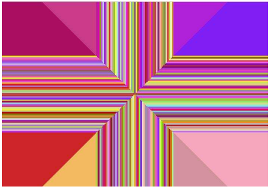
points along a line segment



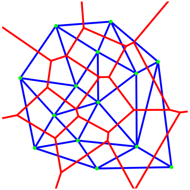
points on a grid inside a square



points on a circle



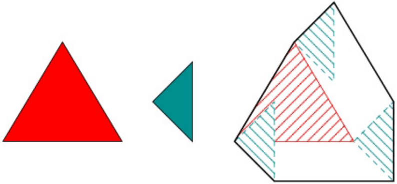
points on a square boundary



Minkowski Sum in $\mathbb{R}^2$ Background

- Given two sets P and Q in the plane, their Minkowski sum, denoted $A \oplus B$ is:

$$P \oplus Q = \{p + q \,|\, p \in P, q \in Q\}$$



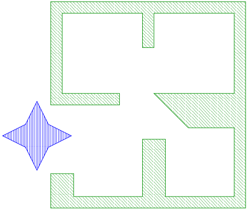
- Minkowski sums are used in many applications
  - motion planning, computer-aided design, manufacturing

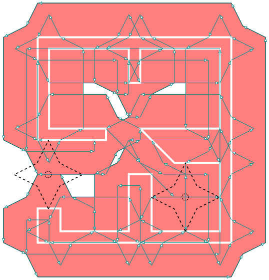
The Minkowski_sum_2 Package

- Based on the `Arrangement_2`, `Polygon_2`, and `Partition_2` packages
- Works well with the `Boolean_set_operations_2` package
  - e.g., It is possible to compute the union of offset polygons
- Robust and efficient
- Supports Minkowski sums of two simple polygons
  - Implemented using either decomposition or convolution
  - Exact
- Supports Minkowski sums of a simple polygon and a disc (polygon offseting)
  - Offers either an exact computation or a conservative approximation scheme

Motion Planning: Computing the free space

- The input robot and the obstacle are non-convex
- Exploits the convolution method
- The output sum contains four holes





For Further Reading I

D. Halperin at al.
Effective Computational Geometry for Curves and Surfaces.
Chapter 1: *Arrangements*
Springer, 2006.

R. Wein, E. Fogel, B. Zukerman, and D. Halperin.
Advanced Programming Techniques Applied to CGAL's Arrangement Package.
To be published in *Computational Geometry - Theory and Applications (CGTA), Special issue on CGAL*, 2006.

I. Haran and D. Halperin.
An Experimental Study of Point Location in General Planar Arrangements.
Proc. 8th Workshop on Algorithm Engineering and Experimentation (Alenex'06), 16–25, 2006.

For Further Reading II

- M. Meyerovitch and D. Halperin.
  Robust, Generic, and Efficient Construction of Envelopes of Surfaces in Three-Dimensional Space.
  Proc. 14th Annual European Symposium on Algorithms (ESA), 4168:792–803, 2006.

- R. Wein
  Exact and Efficient Construction of Planar Minkowski Sums using the Convolution Method.
  Proc. 14th Annual European Symposium on Algorithms (ESA), 4168:829-840, 2006.

- E. Berberich, E. Fogel, D. Halperin, and R. Wein
  Sweeping over Curves and Maintaining Two-Dimensional Arrangements on Surfaces.
  Submitted to the *23rd Annual Symposium on Computational Geometry (SoCG)*, 2007

For Further Reading III

- E. Berberich, A. Eigenwillig, M. Hemmer, S. Hert, and L. Kettner, K. Mehlhorn, J. Reichel, S. Schmitt, E. Schömer, and N. Wolpert
  EXACUS: Efficient and exact algorithms for curves and surfaces.
  Proc. 13th Annual European Symposium on Algorithms (ESA), 3669:155–166, 2005

- E. Berberich, E. Eigenwillig, M. Hemmer, S. Hert, K. Mehlhorn, and E. Schömer
  A Computational Basis for Conic Arcs and Boolean Operations on Conic Polygons.
  Proc. 10th Annual European Symposium on Algorithms (ESA), 2461:174–186, 2002

- A. Eigenwillig, L. Kettner, E. Schömer, and N. Wolpert
  Exact, Efficient and Complete Arrangement Computation for Cubic Curves.
  Computational Geometry, 35(1-2):36–73, Elsevier, 2006

For Further Reading IV

E. Berberich, M. Hemmer, L. Kettner, E. Schömer, and N. Wolpert
An Exact, Complete, and Efficient Implementation for Computing Planar Maps of Quadric Intersection Curves.
Proc. 21st Annual Symposium on Computational Geometry (SoCG), 99–106, 2005