The CGAL Arrangement Package and Its Applications

Efi Fogel, Dan Halperin, Idit Haran, Michal Meyerovitch,
Oren Nechushtan, Ron Wein, and Baruch Zukerman
Tel Aviv Univ.

February 22, 2006

Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

CGAL Goals

- *"make the large body of geometric algorithms developed in the field of CG available for industrial and academic applications"*
- Promote the research in Computational Geometry (CG)

CGAL Goals

- *"make the large body of geometric algorithms developed in the field of CG available for industrial and academic applications"*
- Promote the research in Computational Geometry (CG)

➥ Develop robust programs

CGAL History

Written in C++

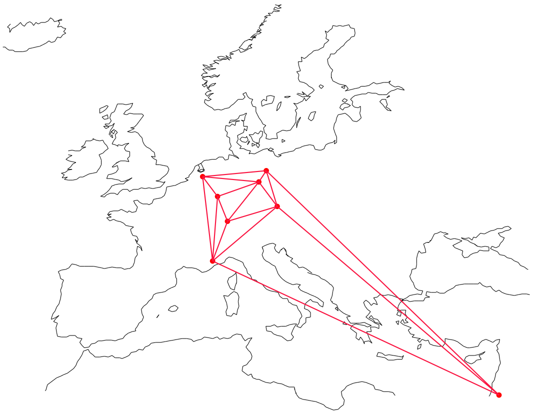
Follows the *generic programming* paradigm

Development started in 1995

GEOMETRY FACTORY was created in 2003

Consortium of 6 active European sites:

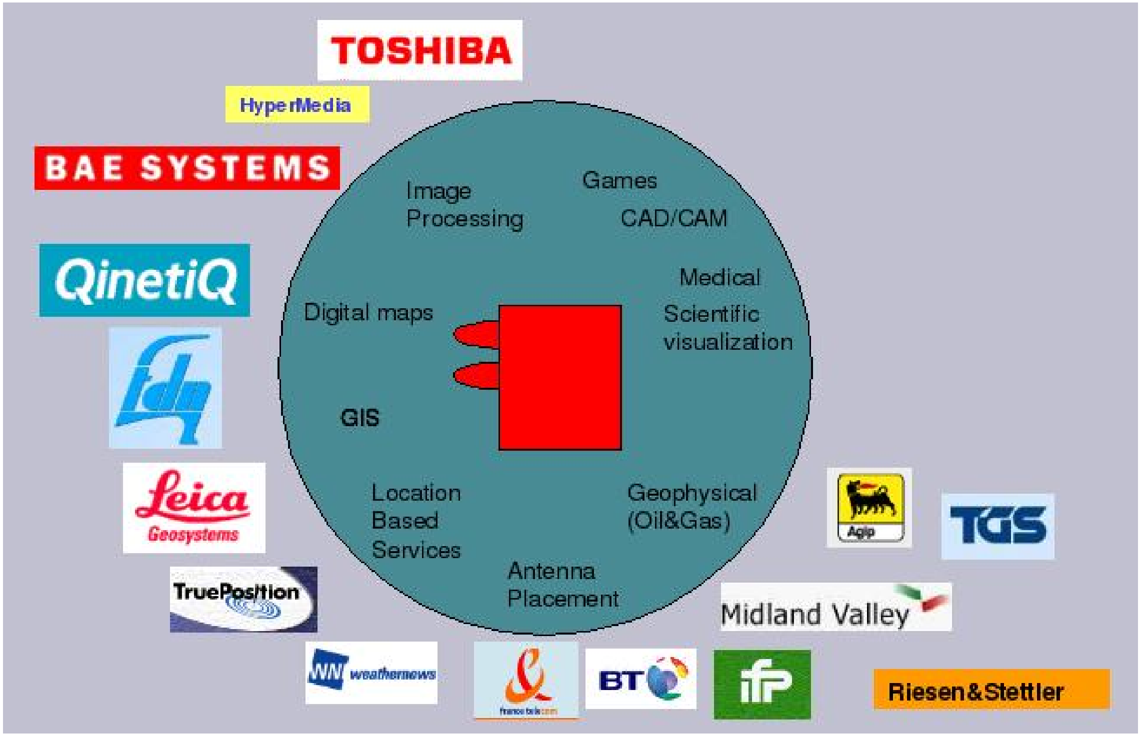
1. Utrecht University
2. INRIA Sophia Antipolis
3. ETH Zürich
4. MPII Saarbrücken
5. Tel Aviv University
6. Freie Universität Berlin
7. RISC Linz
8. Martin-Luther-Universität Halle



CGAL in numbers

- 350.000 lines of C++ code
- ~2000 pages manual
- Release cycle of ~12 months
- CGAL 2.4: 9300 downloads (18 months)
- CGAL 3.0.1: 6200 downloads (8 months)
- 4000 subscribers to the announcement list (7000 for gcc)
- 800 users registered on discussion list (600 in gcc-help)
- 50 developers registered on developer list

Commercial Customers of Geometry Factory



CGAL Structure

Basic Library

Algorithms and Data Structures

Kernel

Geometric Objects of constant size

Geometric Operations on object of constant size

Support Library

configurations, assertions,...

Visualization

Files

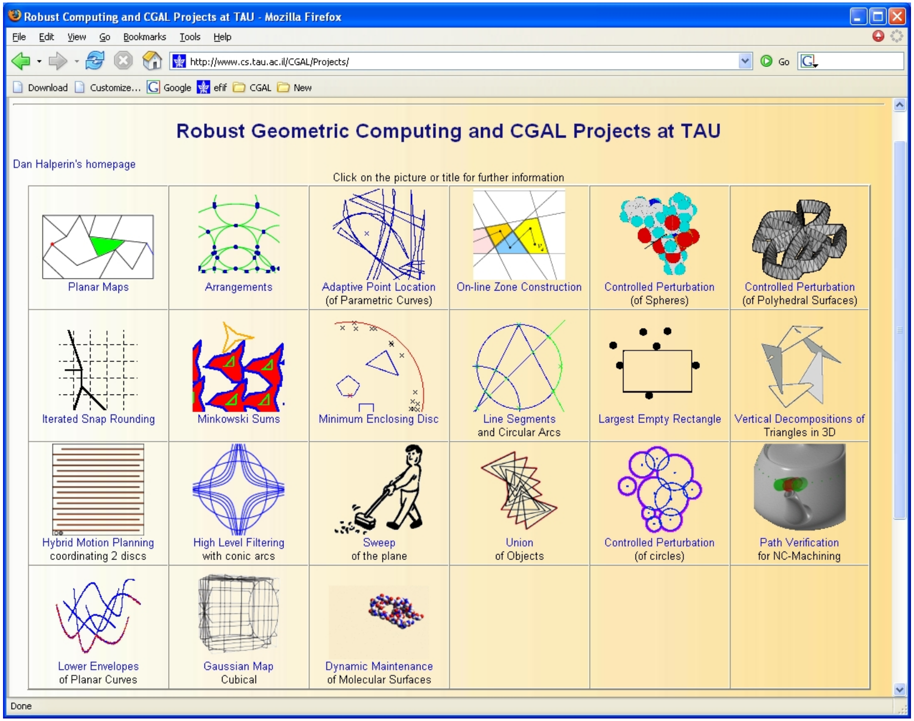
I/O

Number Types

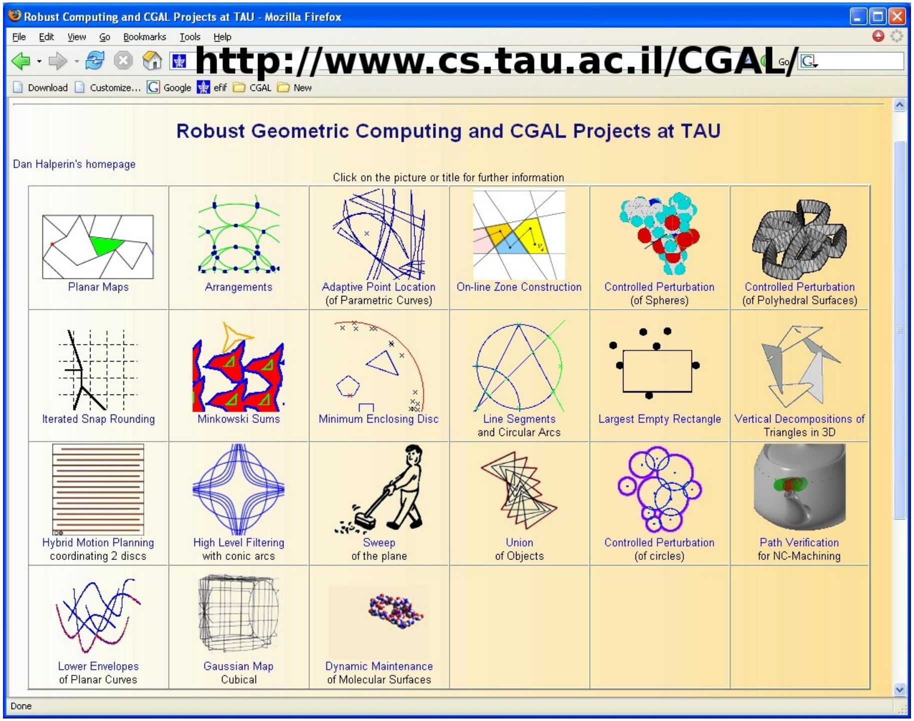
Generators

...

CGAL Projects in TAU



CGAL Projects in TAU

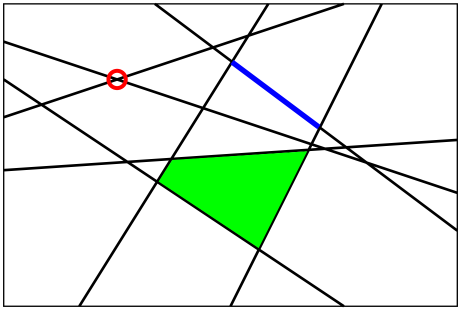


Overview

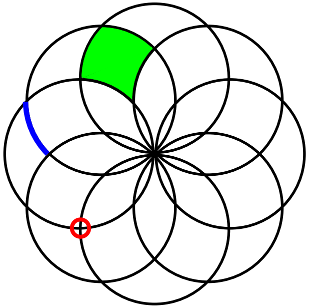
- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

Arrangements

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



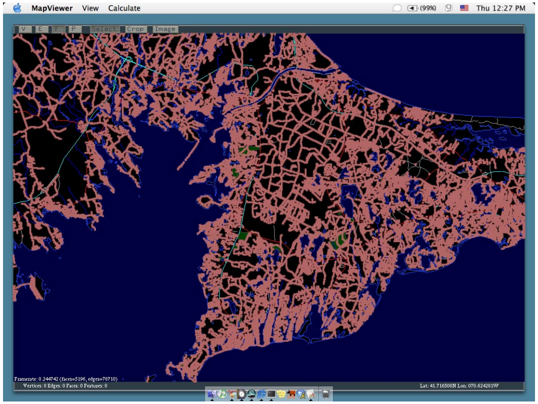
An arrangement of lines

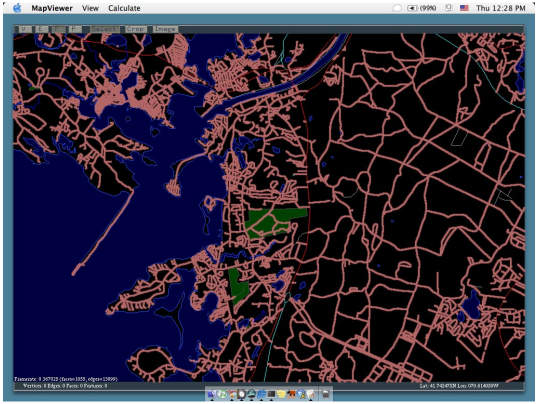


An arrangement of circles

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.

Arrangement Goals

- Construct, maintain, modify, traverse, query and present 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

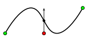
Arrangement Traits

- Separates geometric aspects from topological aspects
  - `Arrangement_2<Traits,Dcel>` — main component[1]
- Parameter of package
  - Defines the family of curves in interest
  - Package can be used with any family of curves for which a traits class is supplied
- Aggregates
  - Geometric types (point, curve)
  - Operations over types (accessors, predicates, constructors)

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

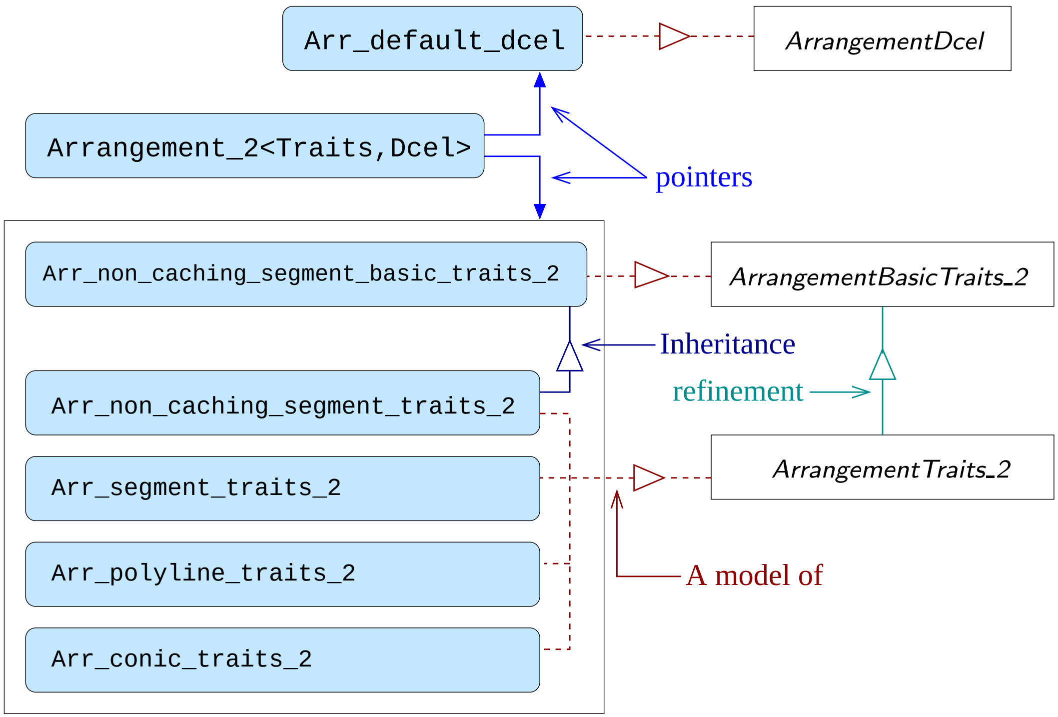
Arrangement Traits Requirements

- Types: `Curve_2`, `X_monotone_curve_2`, `Point_2`
- Methods:



Compare two points

Arrangement Architecture

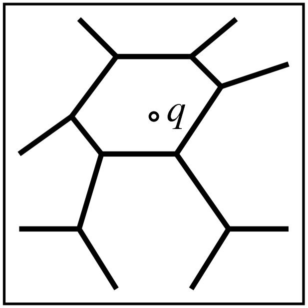


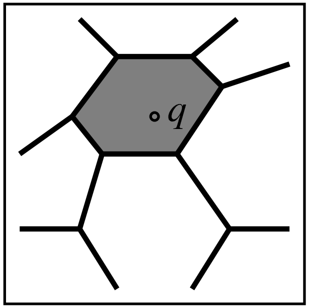
Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

Point Location

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





Goals

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

Point-Location Strategies

	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

(*) Can be reduced to $O(n)$

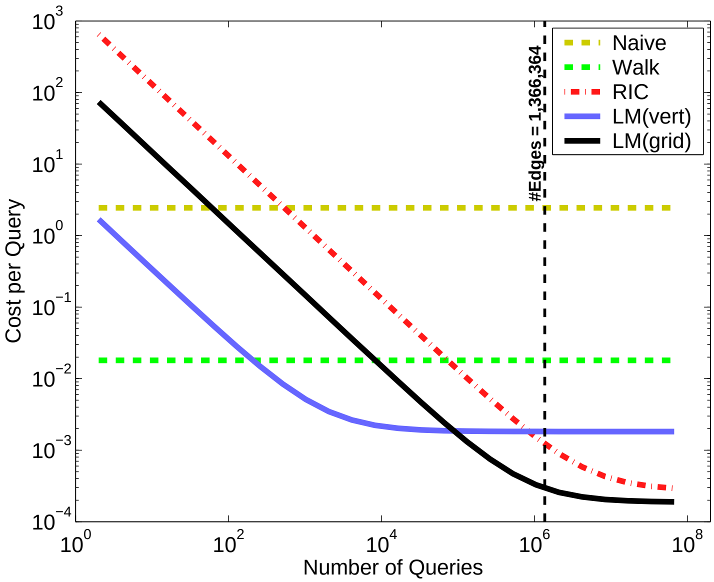
k – number of landmarks

Landmark Point-Location Strategy (jump & walk)

- Select landmarks: vertices, random points, grid, Halton, etc.[2]
- Store landmarks in nearest neighbor search structure, locate landmarks in arrg using batched point location (sweepline)
- For query q: find nearest landmark ℓ, walk from ℓ to q
- Much faster than CGAL's naive or walk strategies
- Much faster preprocessing than CGAL's RIC strategy
- The best combined performance — see next slide
- The ultimate effective strategy? if programming effort and maintainability are considered as well

[2] Idit Haran and Dan Halperin

Landmark Point-Location: performance example



- ≈ 1.5 Million line segments
- Logarithmic scale
- Seconds per query

$$cost(m) = \frac{\text{preprocessing time}}{m} + \text{average query time}$$

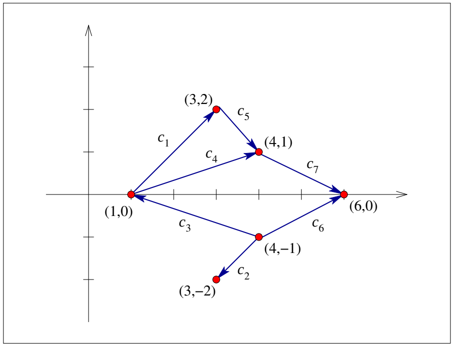
Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

The Complexity of the Outer Face

Outer Face — the unique unbounded face in $\mathcal{A}(\Gamma)$

Outer Face Complexity — the number of edges in the Outer Face (without repetitions)[3]

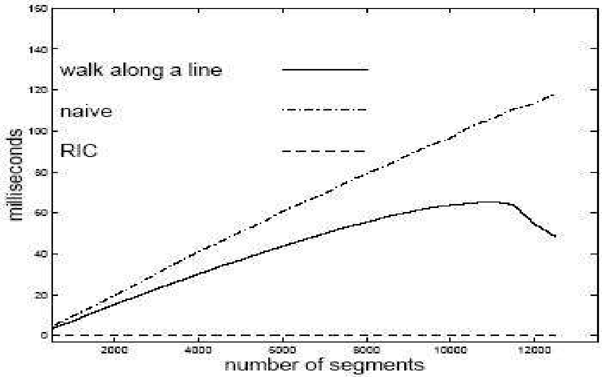


The outer-face complexity is 6

[3] Oren Nechushtan, Noga Alon, Dan Halperin, and Micha Sharir

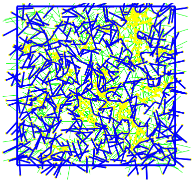
The Complexity of the Outer Face

- Concentrating at random-segment arrangements
- Striving to compute the precise complexity
- Gives an improved analysis of the walk point location

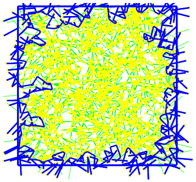


Random-Segment Arrangements

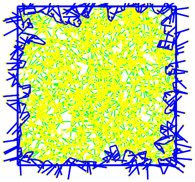
- One endpoint is chosen randomly in the unit square
- The direction is chosen randomly
- Up to $20,000,000$ segments



600 segments
579 boundary
before phase transition



900 segments
372 boundary
after phase transition

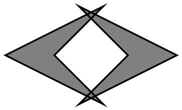


1200 segments
393 boundary
beyond ...

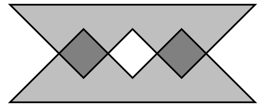
Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

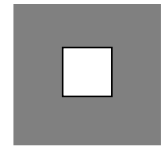
Boolean Set Operations



Union



Intersection



Complement

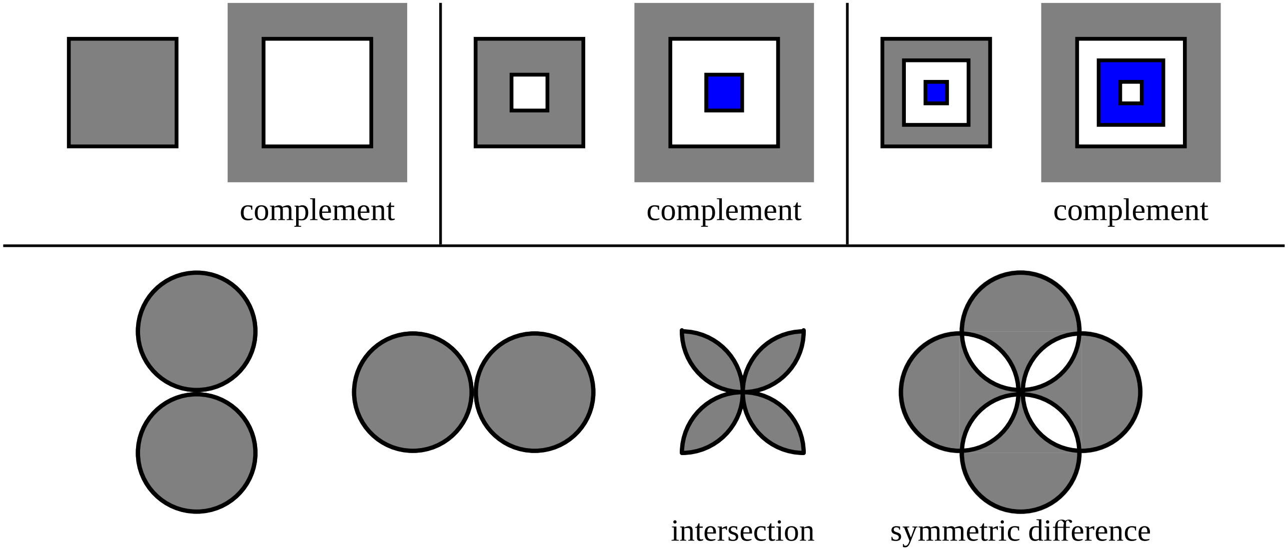
For two point sets P and Q and a point r:

Intersection predicate	$P \cap Q \neq \emptyset$, overlapping cell(s) are not explicitly computed.
Intersection	$R = P \cap Q$
Union	$R = P \cup Q$
Difference	$R = P \setminus Q$
Symmetric Difference	$R = (P \setminus Q) \cup (Q \setminus P)$
Complement	$R = \overline{P}$
Containment predicate	$r \in P$

Boolean set Operations: Terms and Definitions

- Regularized Boolean-set operations (op*)
  - P op* Q = closure(interior(P op Q))
  - Appear in Constructive Solid Geometry (CSG)
    - Used to avoid isolated elements and open boundaries
- *general polygon* — a connected point set whose boundary edges are weakly x-monotone
- *general polygon with holes* — a general polygon that contains holes, which are general polygons
- *general polygon set* — central component
  - Represented by an arrangement. Robust & exact
  - Can output a set of general polygons with holes
  - Parameterized by a traits class

Boolean Set Operations

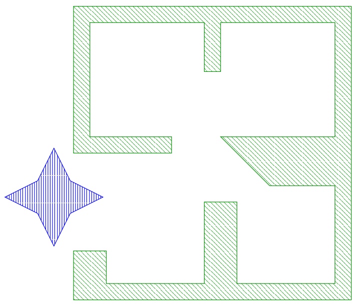


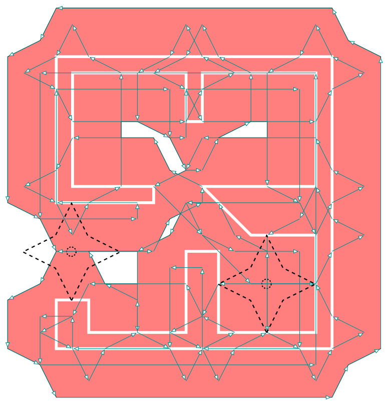
Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

Minkowski Sums in $\mathbb{R}^2$

- Implemented using either decomposition or convolution[4]
- Based on CGAL's arrangements. Robust and exact
- Efficient
  - Running times are two orders of magnitude faster than the ones reported with CGAL 2.0





[4] Ron Wein and Dan Halperin

Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

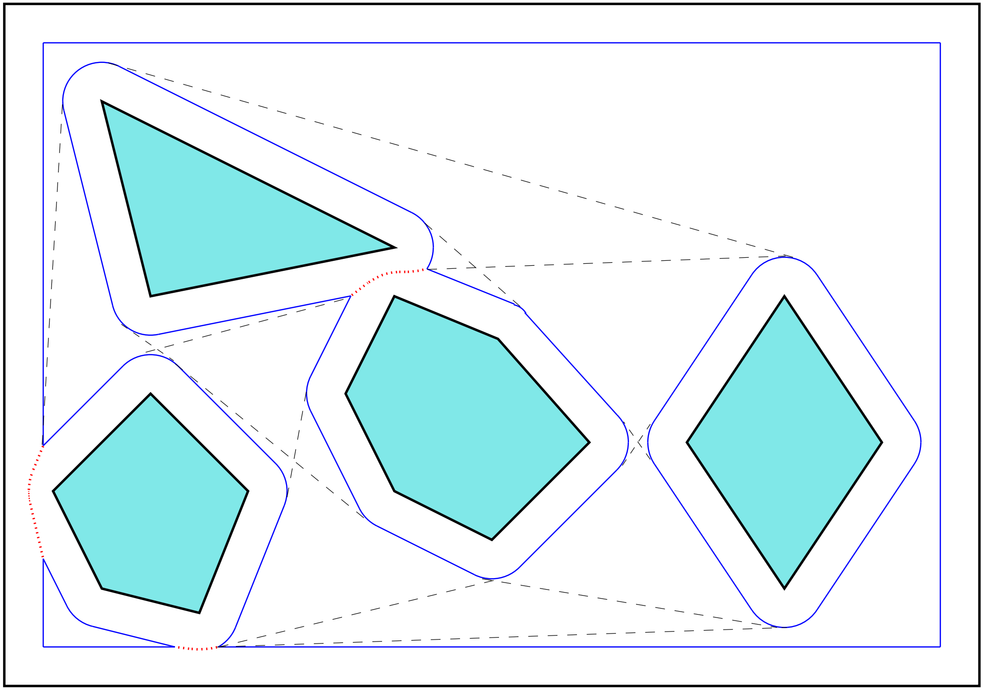
The Visibility-Voronoi Complex

- $VV^{(c)}$ is a hybrid between the visibility graph and the Voronoi diagram[5]
- Parameterized by a clearance value c
- Evolves from the visibility graph to the Voronoi diagram, as c grows $0 \longrightarrow \infty$
- Can be used for planning natural-looking paths for a robot translating amidst obstacles

  Natural-Looking path — smooth, keeps where possible an amount of clearance c from the obstacles

[5] Ron Wein, Jur P. van den Berg, and Dan Halperin

The Visibility-Voronoi Complex



Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

Envelopes in $\mathbb{R}^3$

- Let $F = \{f_1, f_2, \ldots, f_n\}$ be a collection of partially defined bivariate functions
- The lower envelope E_F of F is the pointwise minimum of these functions:

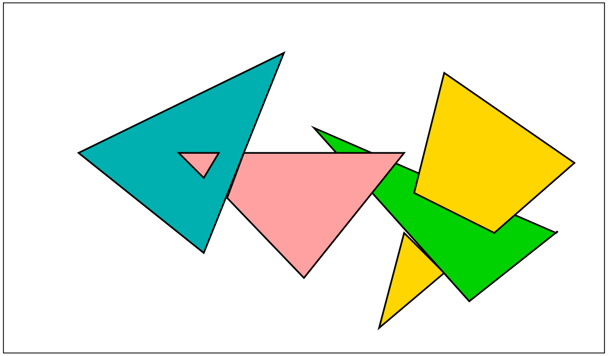
$$E_F(x, y) = \min f_i(x, y) \qquad x, y \in R$$

- If a function f_i is not defined over (x, y), $f_i(x, y) = \infty$
- The minimization diagram is the partition of the plane into maximal connected regions such that E_F is attained by the same subset of functions over each region[6]

[6] Michal Meyerovitch and Dan Halperin

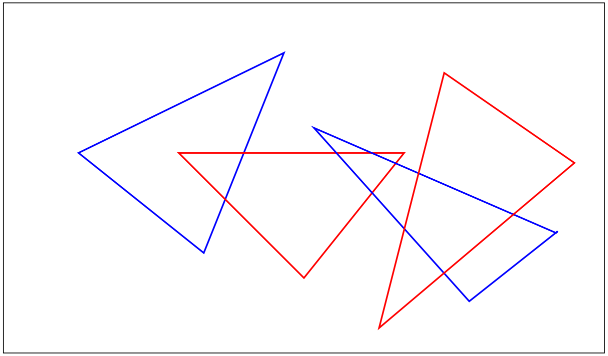
Divide & Conquer Algorithm

- Partition F into 2 sub-collections F_1, F_2
- Recursively construct the *minimization diagrams* M_1, M_2
- Merge M_1, M_2 to obtain the minimization diagram of F



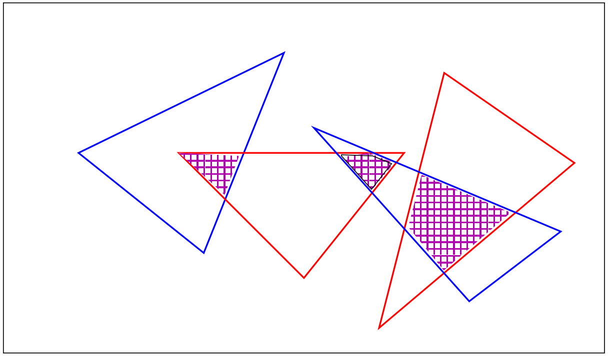
Divide & Conquer Algorithm

- Partition F into 2 sub-collections F_1, F_2
- Recursively construct the *minimization diagrams* M_1, M_2
- Merge M_1, M_2 to obtain the minimization diagram of F



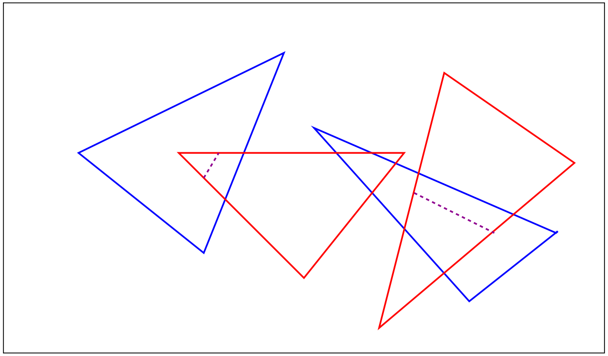
Divide & Conquer Algorithm

- Partition F into 2 sub-collections F_1, F_2
- Recursively construct the *minimization diagrams* M_1, M_2
- Merge M_1, M_2 to obtain the minimization diagram of F



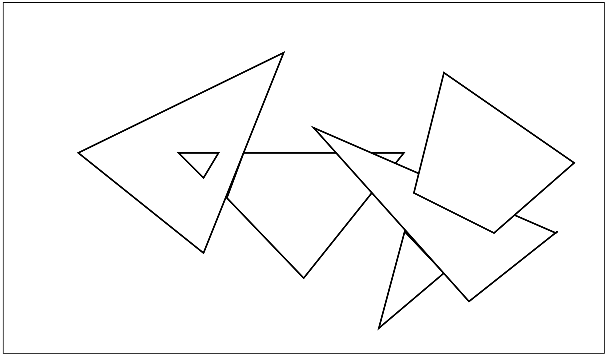
Divide & Conquer Algorithm

- Partition F into 2 sub-collections F_1, F_2
- Recursively construct the *minimization diagrams* M_1, M_2
- Merge M_1, M_2 to obtain the minimization diagram of F



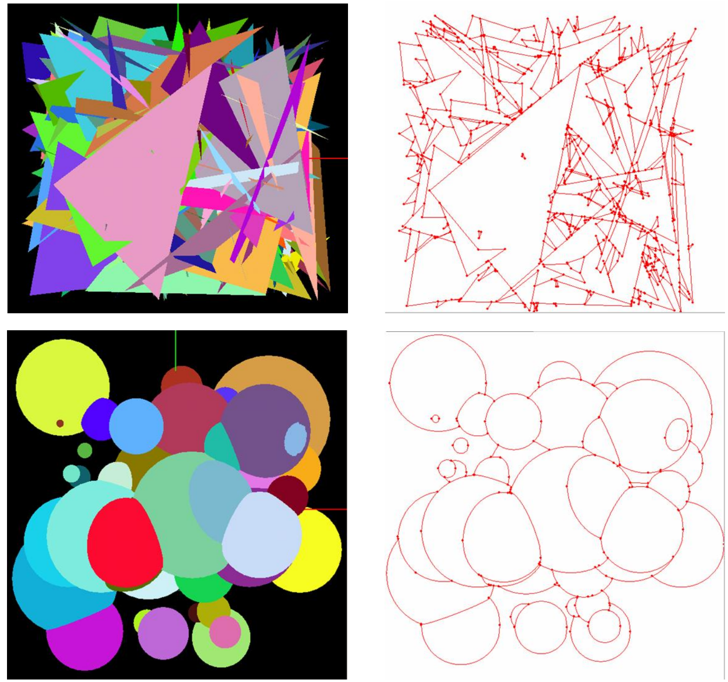
Divide & Conquer Algorithm

- Partition F into 2 sub-collections F_1, F_2
- Recursively construct the *minimization diagrams* M_1, M_2
- Merge M_1, M_2 to obtain the minimization diagram of F



Lower Envelope

- Based on arrangements
  - Robust & exact
- Exploits:
  - Overlay
  - Isolated points
  - Zone traversal
- Currently works for:
  - Triangles
  - Spheres
- Near-future goal:
  - Quadrics (with MPII)



Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

Minkowski Sums in $\mathbb{R}^3$

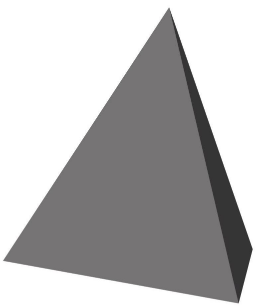
- P and Q are two convex polyhedra (polytopes) in $\mathbb{R}^d$
  - P translated by a vector t is denoted by P^t

$M = P \oplus Q = \{p + q \mid p \in P, q \in Q\}$	Minkowski sum
$\delta(P, Q) = \min\{\lVert t \rVert \mid P^t \cap Q \neq \emptyset, t \in \mathbb{R}^d\}$	minimum distance
$\pi(P, Q) = \min\{\lVert t \rVert \mid P^t \cap Q = \emptyset, t \in \mathbb{R}^d\}$	penetration depth
$\pi_d(P, Q) = \min\{a \mid P^{\vec{d}a} \cap Q = \emptyset\}$	directional penetration depth

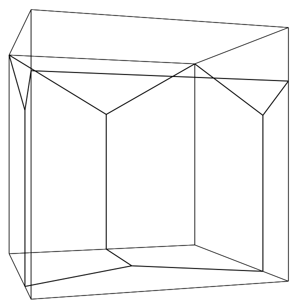
The Cubical Gaussian Map

The *Cubical Gaussian Map* (CGM) C of a polytope P in $\mathbb{R}^3$ is a set-valued function from P to the 6 faces of the unit cube whose edges are parallel to the major axes and are of length two[7]

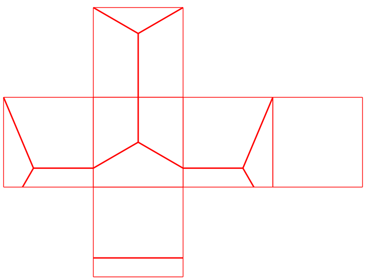
A Tetrahedron



The primal



The CGM



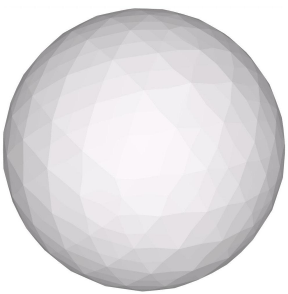
The CGM unfolded

[7] Efi Fogel and Dan Halperin

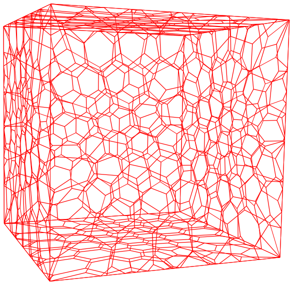
The Cubical Gaussian Map of 3D Minkowski Sums

Amounts to computing the overlay of six arrangement pairs

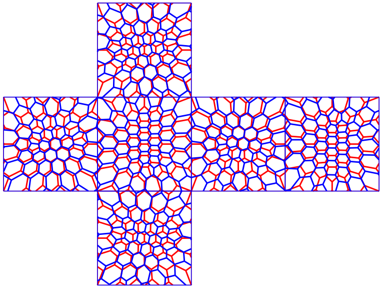
The Minkowski sum of two geodesic spheres level 4 slightly rotated with respect to each other



The primal



The CGM



The CGM unfolded

Polytope complexities

Object Name	**Primal**			**Dual**		
	V	**E**	**F**	**V**	**HE**	**F**
Icosahedron (Icos)	12	30	20	72	192	36
Dioctagonal Pyramid (DP)	17	32	17	105	304	59
Pentagonal Hexecontahedron (PH)	92	150	60	196	684	158
Truncated Icosidodecahedron (TI)	120	180	62	230	840	202
Geodesic Sphere level 4 (GS4)	252	750	500	708	2124	366

Icos — Icosahedron

DP — Dioctagonal Pyramid   ODP — Orthogonal Dioctagonal Pyramid

PH — Pentagonal Hexecontahedron

TI — Truncated Icosidodecahedron

GS4 — Geodesic Sphere level 4   RGS4 — Rotated Geodesic Sphere level 4

Minkowski sum of polytopes in $\mathbb{R}^3$

Obj. 1	Obj. 2	Minkowski Sum						CGM	NGM	LP	CH	$\frac{F_1 F_2}{F}$
		Primal			Dual							
		V	E	F	V	HE	F					
Icos	Icos	12	30	20	72	192	36	**0.01**	0.36	0.04	0.10	20
DP	ODP	131	261	132	242	832	186	**0.02**	1.08	0.35	0.31	2.2
PH	TI	248	586	340	514	1670	333	**0.05**	2.94	1.55	3.85	10.9
GS4	RGS4	1048	2568	1531	1906	6288	1250	**0.31**	14.33	5.80	107.35	163.3

Time consumption (in seconds) of various Minkowski-sum computations

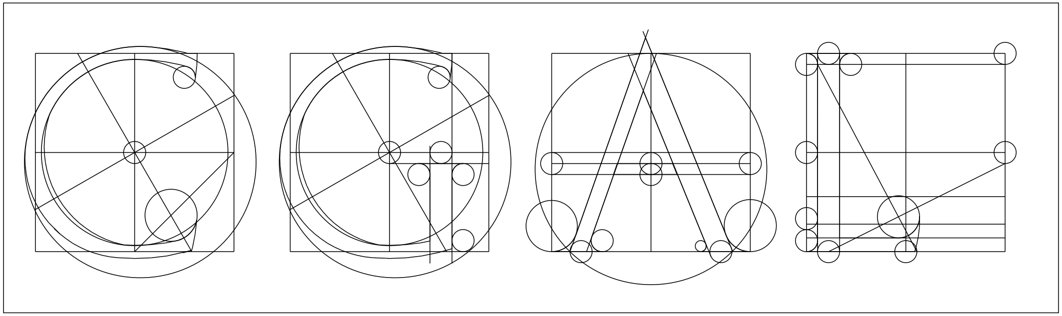
CGM — the Cubical Gaussian Map based method
NGM — the Nef polyhedra based method [Hachenberger, Kettner]
LP — Linear Programming based algorithm [Fukuda, Weibel]
CH — the Convex Hull method
$\frac{F_1 F_2}{F}$ — the ratio between the product of the number of input facets and the number of output facets

Overview

- CGAL
- Arrangements
- Point-Location Strategies
- Outer-Face Complexity
- Boolean Set Operations
- Minkowski Sums in $\mathbb{R}^2$
- The Visibility Voronoi Complex
- Envelopes in $\mathbb{R}^3$
- Minkowski Sums in $\mathbb{R}^3$
- Demo and/or Video

Demo and/or Video

The arrangement of the CGAL logo



- Input: 425 line segments and 34 circles
- Output: 944 vertices, 2941 halfedges, and 534 faces

TOC

Overview ❖

Overview ❖

CGAL Goals ❖

CGAL History ❖

CGAL in numbers ❖

Commercial Customers of GEOMETRY FACTORY ❖

CGAL Structure ❖

CGAL Projects in TAU ❖

CGAL Projects in TAU ❖

Overview ❖

Arrangements ❖

Arrangement Background ❖

Arrangement Goals ❖

Arrangement Traits ❖

Arrangement Traits Requirements ❖

Arrangement Architecture ❖

Overview ❖

Point Location ❖

Point-Location Strategies ❖

Landmark Point-Location Strategy (jump & walk) ❖

Landmark Point-Location: performance example ❖

Overview ❖

The Complexity of the Outer Face ❖

The Complexity of the Outer Face ❖

Random-Segment Arrangements ❖

Overview ❖

Boolean Set Operations ❖

Boolean set Operations: Terms and Definitions ❖

Boolean Set Operations ❖

Overview ❖

Minkowski Sums in $\mathbb{R}^2$ ❖

Overview ❖

The Visibility-Voronoi Complex ❖

The Visibility-Voronoi Complex ❖

Overview ❖

Envelopes in $\mathbb{R}^3$ ❖

Divide & Conquer Algorithm ❖

Divide & Conquer Algorithm ❖

Divide & Conquer Algorithm ❖

Divide & Conquer Algorithm ❖

Divide & Conquer Algorithm ❖

Lower Envelope ❖

Overview ❖

Minkowski Sums in $\mathbb{R}^3$ ❖

The Cubical Gaussian Map ❖

The Cubical Gaussian Map of 3D Minkowski Sums ❖

Polytope complexities ❖

Minkowski sum of polytopes in $\mathbb{R}^3$ ❖

Overview ❖

Demo and/or Video ❖