Computational Geometry

Efi Fogel

Tel Aviv University

CGAL 2D Arrangements
Dec. 19th, 2016

Outline

1. CGAL 2D Arrangements
   - Definitions & Complexity
   - Representation
   - Literature

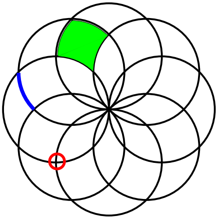
Outline

1. CGAL 2D Arrangements
   - Definitions & Complexity
   - Representation
   - Literature

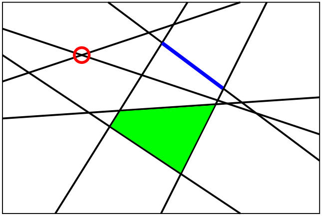
Two Dimensional Arrangements

Definition (Arrangement)

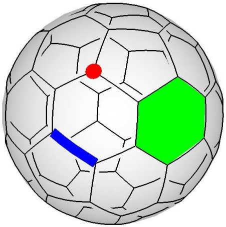
Given a collection $\mathcal{C}$ of curves on a surface, the arrangement $\mathcal{A}(\mathcal{C})$ is the partition of the surface into vertices, edges and faces induced by the curves of $\mathcal{C}$.



An arrangement of circles in the plane.



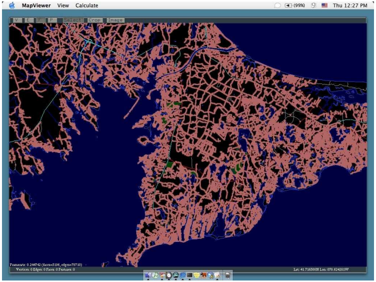
An arrangement of lines in the plane.

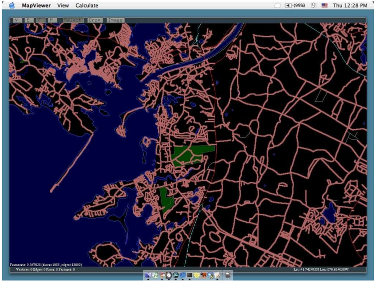


An arrangement of great-circle arcs on a sphere.

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 2D Complexity

Definition (Well Behaved Curves)

Curves in a set $\mathcal{C}$ are well behaved, if each pair of curves in $\mathcal{C}$ intersect at most some constant number of times.

Theorem (Arrangement in $\mathbb{R}^2$)

The maximum combinatorial complexity of an arrangement of n well-behaved curves in the plane is $\Theta(n^2)$.

The complexity of arrangements induced by n non-parallel lines is $\Omega(n^2)$.

Planar Maps

Definition (Planar Graph)

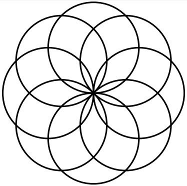
A planar graph is a graph that can be embedded in the plane.

Definition (Planar Map)

A planar map is the embedding of a planar graph in the plane. It is a subdivision of the plane into vertices, (bounded) edges, and faces.

Theorem (Euler Formula)

Let v, e, and f be the number of vertices, edges, and faces (including the unbounded face) of a planar map, then $v - e + f = 2$.



vertices — 25

edges — 56

faces — 33

Surface Maps

Planar maps generalize to surfaces!

Definition (genus)

A topologically invariant property of a surface defined as the largest number of nonintersecting simple closed curves that can be drawn on the surface without separating it.

Theorem (Euler Formula)

Let v, e, and f be the number of vertices, edges, and faces of a map embedded on a surface with genus g, then $v - e + f = 2 - 2g$.

If each face is incident to at least 3 edges $\implies 3f \leq 2e$

$$3v - 3e + 3f = 6 - 6g \leq 3v - 3e + 2e$$

$$e \leq 3v - 6 + 6g$$

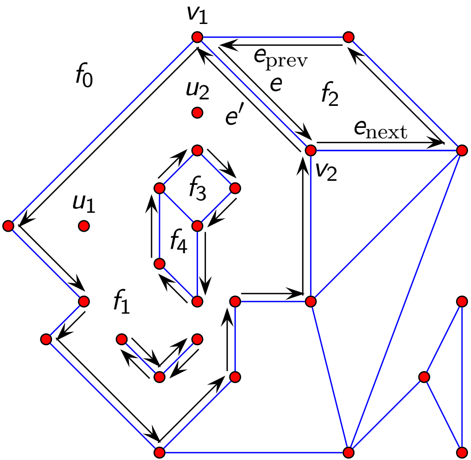
In a planar triangulation $e = 3v - 6,\ f = 2v - 4$

Outline

1. CGAL 2D Arrangements
   - Definitions & Complexity
   - Representation
   - Literature

The Doubly-Connected Edge List

- One of a family of combinatorial data-structures called the *halfedge data-structures*.
- Represents each edge using a pair of directed *halfedges*.
- Maintains incidence relations among cells of 0 (vertex), 1 (edge), and 2 (face) dimensions.



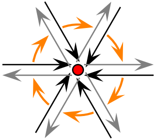
- The target vertex of a halfedge and the halefedge are incident to each other.
- The source and target vertices of a halfedge are adjacent.

The Doubly-Connected Edge List Components

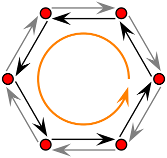
- Vertex
  - An incident halfedge pointing at the vertex.
- Halfedge
  - The opposite halfedge.
  - The previous halfedge in the component boundary.
  - The next halfedge in the component boundary.
  - The target vertex of the halfedge.
  - The incident face.
- Face
  - An incident halfedge on the outer CCB.
  - An incident halfedge on each inner CCB.
- Connected component of the boundary (CCB)
  - The circular chains of halfedges around faces.

Arrangement Representation

- The halfedges incident to a vertex form a circular list.
- The halfedges are sorted in clockwise order around the vertex.

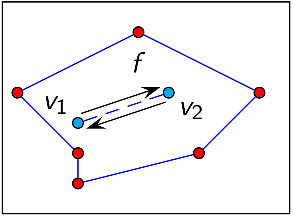


- The halfedges around faces form circular chains.
- All halfedges of a chain are incident to the same face.
- The halfedges are sorted in counterclockwise order along the boundary.

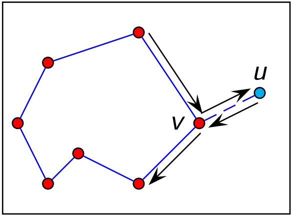


- Geometric interpretation is added by classes built on top of the halfedge data-structure.

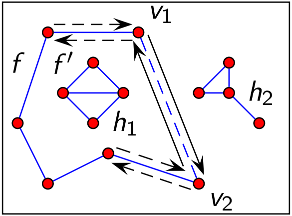
Modifying the Arrangement



Inserting a curve that induces a new hole inside the face f, `arr.insert_in_face_interior(c,f)`.



Inserting a curve from an existing vertex u that corresponds to one of its endpoints, `insert_from_left_vertex(c,v)`, `insert_from_right_vertex(c,v)`.



Inserting an x-monotone curve, the endpoints of which correspond to existing vertices v_1 and v_2, `insert_at_vertices(c,v1,v2)`.

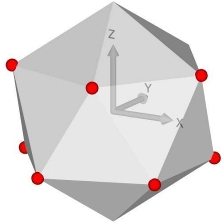
- The new pair of halfedges close a new face f'.
- The hole h_1, which belonged to f before the insertion, becomes a hole in this new face.

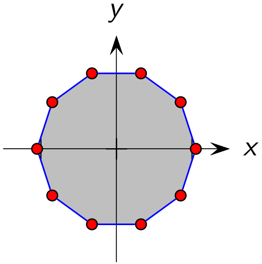
Application: Obtaining Silhouettes of Polytopes

Application

Given a convex polytope P obtain the outline of the shadow of P cast on the xy-plane, where the scene is illuminated by a light source at infinity directed along the negative z-axis.

- The silhouette is represented as an arrangement with two faces:
  - an unbounded face and
  - a single hole inside the unbounded face.





An icosahedron and its silhouette.

Application: Obtaining Silhouettes of Polytopes: Insertion

- Insert an edge into the arrangement only once to avoid overlaps.
  - Maintain a set of handles to polytope edges the projection of which have already been inserted into the arrangement.
  - Implemented with the std :: set data-structure.
    - ★ Requires the provision of a model of the *StrictWeakOrdering*.
    - ★ A functor that compares handles:

```
struct Less_than_handle {
  template <typename Type>
  bool operator()(Type s1, Type s2) const { return (&(*s1) < &(*s2)); }
};
```

```
std::set<Polyhedron_halfedge_const_handle, Less_than_handle>}
```

- Determine the appropriate insertion routines.
  - Maintain a map that maps polyhedron vertices to corresponding arrangement vertices.
  - Implemented with the std :: map data-structure.

```
std::map<typename Polyhedron::Vertex_const_handle,
         typename Arrangement::Vertex_handle, Less_than_handle>
```

Application: Obtaining Silhouettes of Polytopes: Construction

Obtain the arrangement $\mathcal{A}$ that represents the silhouette of a Convex Polytope P
1. Construct the input convex polytope P.
2. Compute the normals to all facets of P.
3. **for each** facet f of P
4.   **if** f is facing upwards (has a positive z component)
5.   **for each** edge e on the boundary of f
6.   **if** the projection of e hasn't been inserted yet into $\mathcal{A}$
7.   Insert the projection of e into $\mathcal{A}$.

Computes the normal to a facet.

```
struct Normal_equation {
  template <typename Facet> typename Facet::Plane_3 operator()(Facet & f) {
    typename Facet::Halfedge_handle h = f.halfedge();
    return CGAL::cross_product(h->next()->vertex()->point() -
                               h->vertex()->point(),
                               h->next()->next()->vertex()->point() -
                               h->next()->vertex()->point());
  }
};
```

Outline

1. CGAL 2D Arrangements
   - Definitions & Complexity
   - Representation
   - Literature

Arrangement Bibliography I

Boris Aronov and Dmitriy Drusvyatskiy
Complexity of a Single Face in an Arrangement of s-Intersecting Curves
arXiv:1108.4336, 2011

Jon Louis Bentley and Thomas Ottmann.
Algorithms for Reporting and Counting Geometric Intersections.
IEEE Transactions on Computers, 28(9): 643–647, 1979.

Eric Berberich, Efi Fogel, Dan Halperin, Michael Kerber, and Ophir Setter.
Arrangements on parametric surfaces ii: Concretizations and applications, 2009.
Mathematics in Computer Science, 4(1):67–91,2010.

Ulrich Finke and Klaus H. Hinrichs.
Overlaying simply connected planar subdivisions in linear time.
In *Proceedings of* 11^{th} *Annual ACM Symposium on Computational Geometry (SoCG)*, pages 119–126. Association for Computing Machinery (ACM) Press, 1995.

Ron Wein, Efi Fogel, Baruch Zukerman, Dan Halperin, and Eric Berberich.
2D Arrangements.
In Cgal Editorial Board, editor, *Cgal User and Reference Manual*. 4.4 edition, 2014. `http://www.cgal.org/Manual/latest/doc_html/cgal_manual/packages.html#Pkg:Arrangement2`.

David G. Kirkpatrick.
Optimal search in planar subdivisions.
SIAM Journal on Computing. 12(1):28–35,1983.

N. Sarnak and Robert E. Tarjan.
Planar point location using persistent search trees.
Communications of the ACM. 29(7):669–679, 1986.

Arrangement Bibliography II

- Kentan Mulmuley.
  A fast planar partition algorithm, I.
  Journal of Symbolic Computation. 10(3-4):253–280,1990.
- Raimund Seidel
  A Simple and Fast Incremental Randomized Algorithm for Computing Trapezoidal Decompositions and for Triangulating Polygons.
  Computational Geometry: Theory and Applications. 1(1):51–64, 1991.
- Olivier Devillers, Sylvain Pion, and Monique Teillaud.
  Walking in a triangulation.
  International Journal of Foundations of Computer Science. 13:181–199,2002.
- Luc Devroye Christophe, Christophe Lemaire, and Jean-Michel Moreau.
  Fast Delaunay Point-Location with Search Structures.
  In *Proceedings of* 11^{th} *Canadian Conference on Computational Geometry*. Pages 136–141, 1999.
- Luc Devroye, Ernst Peter Mücke, and Binhai Zhu.
  A Note on Point Location in Delaunay Triangulations of Random Points.
  Algorithmica. 22:477–482, 1998.
- Olivier Devillers.
  The Delaunay hierarchy.
  International Journal of Foundations of Computer Science. 13:163-180, 2002.
- Sunil Arya
  A Simple Entropy-Based Algorithm for Planar Point Location.
  ACM Transactions on Graphics. 3(2), 2007
- Masato Edahiro, Iwao Kokubo, And Takao Asano
  A new Point-Location Algorithm and its Practical Efficiency: comparison with existing algorithms
  ACM Transactions on Graphics. 3(2):86–109, 1984.

Arrangement Bibliography III

- Micha Sharir and Pankaj Kumar Agarwal
  Davenport-Schinzel Sequences and Their Geometric Applications.
  Cambridge University Press, New York, NY, 1995.

- Bernard Chazelle, Leonidas J. Guibas, and Der-Tsai Le.
  The Power of Geometric Duality.
  BIT, 25:76–90, 1985.

- Herbert Edelsbrunner,
  Algorithms in Combinatorial Geometry,
  Springer, Heidelberg, 1987.

- Mark de Berg, Mark van Kreveld, Mark H. Overmars, and Otfried Cheong.
  Computational Geometry: Algorithms and Applications.
  Springer, 3^{rd} edition, 2008.

- Herbert Edelsbrunner, Raimund Seidel, and Micha Sharir.
  On the Zone Theorem for Hyperplane Arrangements.
  SIAM Journal on Computing. 22(2):418–429,1993.

- Silvio Micali and Vijay V. Vazirani.
  An $O(\sqrt{|V||E|})$ Algorithm for Finding Maximum Matching in General Graphs.
  Proceedings of 21^{st} *Annual IEEE Symposium on the Foundations of Computer Science*, pages 17–27, 1980.

- Jack Edmonds.
  Paths, Trees, and Flowers.
  Canadian Journal of Mathematics, 17:449–467,1965.

- Robert Endre Tarjan.
  Data structures and network algorithms, Society for Industrial and Applied Mathematics (SIAM), 1983.

Arrangement Bibliography IV

Marcin Mucha and Piotr Sankowski.
Maximum Matchings via Gaussian Elimination
Proceedings of 45^{th} *Annual IEEE Symposium on the Foundations of Computer Science*, pages 248–255, 2004.

Jeremy G. Siek, Lie-Quan Lee, and Andrew Lumsdaine.
The BOOST *Graph Library*.
Addison-Wesley, 2002

Efi Fogel, Ron Wein, and Dan Halperin.
CGAL *Arrangements and Their Applications, A Step-by-Step Guide*.
Springer, 2012.