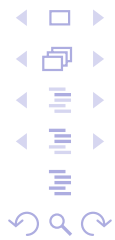
Computational Geometry (Master Course)

Mohammad Farshi

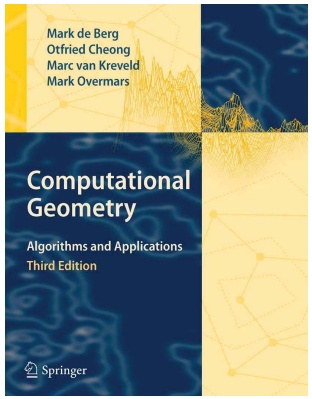
Department of Computer Science,
Yazd University

1389-2

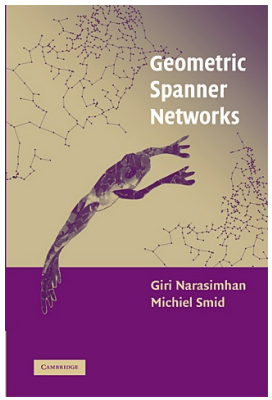


Textbooks:

1) Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars, **Computational Geometry Algorithms and Applications**, 3rd Edition, Springer-Verlag Berlin Heidelberg, 2008.



2) Giri Narasimhan, Michiel Smid, **Geometric Spanner Networks**, CAMBRIDGE UNIVERSITY PRESS, 2007.



Grading and webpage:

Yazd Univ.

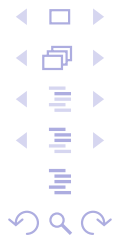
Grading:

- Midterm exam: 7
- Final exam: 8
- Presentation: 3
- Homework: 2

Course Webpage:

http://cs.yazduni.ac.ir/farshi/Teaching/CG3892/CG.html

Introduction



What is CG?

Computational Geometry

Computational geometry is a branch of computer science devoted to the study of algorithms which can be stated in terms of geometry.

Applications

- Computer graphics,
- Computer-aided design and manufacturing (CAD/CAM),
- Robotics (motion planning and visibility problems),
- Geographic Information Systems (GIS) (geometrical location and search, route planning),
- Integrated Circuit design (IC geometry design and verification),
- and so on.

What is CG?

Computational Geometry

Computational geometry is a branch of computer science devoted to the study of algorithms which can be stated in terms of geometry.

Applications

- Computer graphics,
- Computer-aided design and manufacturing (CAD/CAM),
- Robotics (motion planning and visibility problems),
- Geographic Information Systems (GIS) (geometrical location and search, route planning),
- Integrated Circuit design (IC geometry design and verification),
- and so on.

What is CG?

The main branches of computational geometry are:

- **Combinatorial computational geometry**, also called **algorithmic geometry**, which deals with geometric objects as discrete entities. A groundlaying book in the subject by Preparata and Shamos dates the first use of the term "computational geometry" in this sense by 1975.
- **Numerical computational geometry**, also called **machine geometry**, computer-aided geometric design (CAGD), or geometric modeling, which deals primarily with representing real-world objects in forms suitable for computer computations in CAD/CAM systems. This branch may be seen as a further development of descriptive geometry and is often considered a branch of computer graphics or CAD.

What is CG?

Combinatorial computational geometry

- The primary goal is to develop *efficient algorithms* and *data structures* for *solving problems* stated in terms of basic geometrical objects: points, line segments, polygons, polyhedra, etc.
- Example: The closest pair problem: Given n points in the plane, find the two with the smallest distance from each other. The brute-force algorithm takes $\mathcal{O}(n^2)$ time. A classic result: an algorithm that takes $\mathcal{O}(n \log n)$ time. Also randomized algorithms that take $\mathcal{O}(n)$ expected time, as well as a deterministic algorithm that takes $\mathcal{O}(n \log \log n)$ time.
- Computational geometry focuses heavily on *computational complexity* since the algorithms are meant to be used on very large data sets containing tens or hundreds of millions of points.

What is CG?

Combinatorial computational geometry

- The primary goal is to develop *efficient algorithms* and *data structures* for *solving problems* stated in terms of basic geometrical objects: points, line segments, polygons, polyhedra, etc.
- Example: The closest pair problem: Given n points in the plane, find the two with the smallest distance from each other. The brute-force algorithm takes $\mathcal{O}(n^2)$ time. A classic result: an algorithm that takes $\mathcal{O}(n \log n)$ time. Also randomized algorithms that take $\mathcal{O}(n)$ expected time, as well as a deterministic algorithm that takes $\mathcal{O}(n \log \log n)$ time.
- Computational geometry focuses heavily on *computational complexity* since the algorithms are meant to be used on very large data sets containing tens or hundreds of millions of points.

What is CG?

Combinatorial computational geometry

- The primary goal is to develop *efficient algorithms* and *data structures* for *solving problems* stated in terms of basic geometrical objects: points, line segments, polygons, polyhedra, etc.
- Example: The closest pair problem: Given n points in the plane, find the two with the smallest distance from each other. The brute-force algorithm takes $\mathcal{O}(n^2)$ time. A classic result: an algorithm that takes $\mathcal{O}(n \log n)$ time. Also randomized algorithms that take $\mathcal{O}(n)$ expected time, as well as a deterministic algorithm that takes $\mathcal{O}(n \log \log n)$ time.
- Computational geometry focuses heavily on *computational complexity* since the algorithms are meant to be used on very large data sets containing tens or hundreds of millions of points.

What is CG?

Problem classes

- Static problems
  1. Convex hull: Given a set of points, find the smallest convex polyhedron/polygon containing all the points.
  2. Line segment intersection: Find the intersections between a given set of line segments.
  3. Delaunay triangulation
  4. Voronoi diagram: Given a set of points, partition the space according to which point is closest.
  5. Linear programming
  6. Closest pair of points: Given a set of points, find the two with the smallest distance from each other.
  7. Euclidean shortest path: Connect two points in a Euclidean space (with polyhedral obstacles) by a shortest path.
  8. Polygon triangulation: Given a polygon, partition its interior into triangles

What is CG?

Problem classes

- Geometric query problems
  1. Range searching: Preprocess a set of points, in order to efficiently count the number of points inside a query region.
  2. Point location: Given a partitioning of the space into cells, produce a data structure that efficiently tells in which cell a query point is located.
  3. Nearest neighbor: Preprocess a set of points, in order to efficiently find which point is closest to a query point.
  4. Ray tracing: Given a set of objects in space, produce a data structure that efficiently tells which object a query ray intersects first.

What is CG?

Problem classes

- Dynamic problems
- Variations
  - Point in polygon: Decide whether a point is inside or outside a given polygon.

What is CG?

Problem classes

- Dynamic problems
- Variations
  - Point in polygon: Decide whether a point is inside or outside a given polygon.

Journals:

Journals

1. Computational Geometry: Theory and Applications (CGTA)
2. Discrete & Computational Geometry (DCG)
3. International Journal of Computational Geometry and Applications (IJCGA)
4. Journal of Computational Geometry (NEW)
5. Other algorithmic journals

Conferences:

Conferences

1. ACM Symposium on Computational Geometry (SOCG)
2. Canadian Conference on Computational Geometry (CCCG)
3. European Workshop on Computational Geometry (EWCG)
4. International Conference on Computational Geometry and Computer Vision
5. Others, like SODA, STOC, ESA.