WIKIPEDIA

Computational geometry

Computational geometry is a branch of computer science devoted to the study of algorithms which can be stated in terms of geometry. Some purely geometrical problems arise out of the study of computational geometric algorithms, and such problems are also considered to be part of computational geometry. While modern computational geometry is a recent development, it is one of the oldest fields of computing with a history stretching back to antiquity.

Computational complexity is central to computational geometry, with great practical significance if algorithms are used on very large datasets containing tens or hundreds of millions of points. For such sets, the difference between $O(n^2)$ and $O(n \log n)$ may be the difference between days and seconds of computation.

The main impetus for the development of computational geometry as a discipline was progress in computer graphics and computer-aided design and manufacturing (CAD/CAM), but many problems in computational geometry are classical in nature, and may come from mathematical visualization.

Other important applications of computational geometry include robotics (motion planning and visibility problems), geographic information systems (GIS) (geometrical location and search, route planning), integrated circuit design (IC geometry design and verification), computer-aided engineering (CAE) (mesh generation), and computer vision (3D reconstruction).

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.[1]
- *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. The term "computational geometry" in this meaning has been in use since 1971.[2]

Although most algorithms of computational geometry have been developed (and are being developed) for electronic computers, some algorithms were developed for unconventional computers (e.g. optical computers [3])

Contents

Combinatorial computational geometry
- Problem classes
  - Static problems
  - Geometric query problems
  - Dynamic problems
  - Variations

Numerical computational geometry
See also
References
Further reading
Journals
Combinatorial/algorithmic computational geometry
External links

Combinatorial computational geometry

The primary goal of research in combinatorial computational geometry is to develop efficient algorithms and data structures for solving problems stated in terms of basic geometrical objects: points, line segments, polygons, polyhedra, etc.

Some of these problems seem so simple that they were not regarded as problems at all until the advent of computers. Consider, for example, the *Closest pair problem*:

- Given n points in the plane, find the two with the smallest distance from each other.

One could compute the distances between all the pairs of points, of which there are $n(n-1)/2$, then pick the pair with the smallest distance. This brute-force algorithm takes $O(n^2)$ time; i.e. its execution time is proportional to the square of the number of points. A classic result in computational geometry was the formulation of an algorithm that takes $O(n \log n)$. Randomized algorithms that take $O(n)$ expected time,[4] as well as a deterministic algorithm that takes $O(n \log \log n)$ time,[5] have also been discovered.

Problem classes

The core problems in computational geometry may be classified in different ways, according to various criteria. The following general classes may be distinguished.

Static problems

In the problems of this category, some input is given and the corresponding output needs to be constructed or found. Some fundamental problems of this type are:

- Convex hull: Given a set of points, find the smallest convex polyhedron/polygon containing all the points.
- Line segment intersection: Find the intersections between a given set of line segments.
- Delaunay triangulation
- Voronoi diagram: Given a set of points, partition the space according to which points are closest to the given points.
- Linear programming
- Closest pair of points: Given a set of points, find the two with the smallest distance from each other.
- Farthest pair of points

- Largest empty circle: Given a set of points, find a largest circle with its center inside of their convex hull and enclosing none of them.
- Euclidean shortest path: Connect two points in a Euclidean space (with polyhedral obstacles) by a shortest path.
- Polygon triangulation: Given a polygon, partition its interior into triangles
- Mesh generation
- Boolean operations on polygons

The computational complexity for this class of problems is estimated by the time and space (computer memory) required to solve a given problem instance.

Geometric query problems

In geometric query problems, commonly known as geometric search problems, the input consists of two parts: the search space part and the query part, which varies over the problem instances. The search space typically needs to be preprocessed, in a way that multiple queries can be answered efficiently.

Some fundamental geometric query problems are:

- Range searching: Preprocess a set of points, in order to efficiently count the number of points inside a query region.
- 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.
- Nearest neighbor: Preprocess a set of points, in order to efficiently find which point is closest to a query point.
- Ray tracing: Given a set of objects in space, produce a data structure that efficiently tells which object a query ray intersects first.

If the search space is fixed, the computational complexity for this class of problems is usually estimated by:

- the time and space required to construct the data structure to be searched in
- the time (and sometimes an extra space) to answer queries.

For the case when the search space is allowed to vary, see "Dynamic problems".

Dynamic problems

Yet another major class is the dynamic problems, in which the goal is to find an efficient algorithm for finding a solution repeatedly after each incremental modification of the input data (addition or deletion input geometric elements). Algorithms for problems of this type typically involve dynamic data structures. Any of the computational geometric problems may be converted into a dynamic one, at the cost of increased processing time. For example, the range searching problem may be converted into the dynamic range searching problem by providing for addition and/or deletion of the points. The dynamic convex hull problem is to keep track of the convex hull, e.g., for the dynamically changing set of points, i.e., while the input points are inserted or deleted.

The computational complexity for this class of problems is estimated by:

- the time and space required to construct the data structure to be searched in
- the time and space to modify the searched data structure after an incremental change in the search space
- the time (and sometimes an extra space) to answer a query.

Variations

Some problems may be treated as belonging to either of the categories, depending on the context. For example, consider the following problem.

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

In many applications this problem is treated as a single-shot one, i.e., belonging to the first class. For example, in many applications of computer graphics a common problem is to find which area on the screen is clicked by a pointer. However, in some applications, the polygon in question is invariant, while the point represents a query. For example, the input polygon may represent a border of a country and a point is a position of an aircraft, and the problem is to determine whether the aircraft violated the border. Finally, in the previously mentioned example of computer graphics, in CAD applications the changing input data are often stored in dynamic data structures, which may be exploited to speed-up the point-in-polygon queries.

In some contexts of query problems there are reasonable expectations on the sequence of the queries, which may be exploited either for efficient data structures or for tighter computational complexity estimates. For example, in some cases it is important to know the worst case for the total time for the whole sequence of N queries, rather than for a single query. See also "amortized analysis".

Numerical computational geometry

This branch is also known as **geometric modelling** and **computer-aided geometric design** (CAGD).

Core problems are curve and surface modelling and representation.

The most important instruments here are parametric curves and parametric surfaces, such as Bézier curves, spline curves and surfaces. An important non-parametric approach is the level-set method.

Application areas of computational geometry include shipbuilding, aircraft, and automotive industries.

See also

- List of combinatorial computational geometry topics
- List of numerical computational geometry topics
- CAD/CAM/CAE
- List of geometric algorithms
- Solid modeling
- Computational topology
- Computer representation of surfaces
- Digital geometry

- Discrete geometry (combinatorial geometry)
- Space partitioning
- Tricomplex number
- Robust geometric computation
- Wikiversity:Topic:Computational geometry
- Wikiversity:Computer-aided geometric design

References

1. Franco P. Preparata and Michael Ian Shamos (1985). *Computational Geometry - An Introduction*. Springer-Verlag. ISBN 0-387-96131-3. 1st edition; 2nd printing, corrected and expanded, 1988.
2. A.R. Forrest, "Computational geometry", *Proc. Royal Society London*, 321, series 4, 187-195 (1971)
3. Yevgeny B. Karasik (2019). *Optical Computational Geometry*. ISBN 979-8511243344.
4. S. Khuller and Y. Matias. A simple randomized sieve algorithm for the closest-pair problem (https://www.sciencedirect.com/science/article/pii/S0890540185710498/pdf?md5=aef3739b6ce3d0a18f11ef1915827ffe&pid=1-s2.0-S0890540185710498-main.pdf&_valck=1). Inf. Comput., 118(1):34—37,1995 (PDF)
5. S. Fortune and J.E. Hopcroft. "A note on Rabin's nearest-neighbor algorithm." Information Processing Letters, 8(1), pp. 20—23, 1979

Further reading

- List of books in computational geometry

Journals

Combinatorial/algorithmic computational geometry

Below is the list of the major journals that have been publishing research in geometric algorithms. Please notice with the appearance of journals specifically dedicated to computational geometry, the share of geometric publications in general-purpose computer science and computer graphics journals decreased.

- *ACM Computing Surveys*
- *ACM Transactions on Graphics*
- *Acta Informatica*
- *Advances in Geometry*
- *Algorithmica*
- *Ars Combinatoria*
- *Computational Geometry: Theory and Applications*
- *Communications of the ACM*
- *Computer Aided Geometric Design (https://web.archive.org/web/20111122080921/http://www.journals.elsevier.com/computer-aided-geometric-design/#description)*
- *Computer Graphics and Applications*

- *Computer Graphics World*
- *Discrete & Computational Geometry*
- *Geombinatorics*
- *Geometriae Dedicata*
- *IEEE Transactions on Graphics*
- *IEEE Transactions on Computers*
- *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- *Information Processing Letters*
- *International Journal of Computational Geometry and Applications*
- *Journal of Combinatorial Theory, series B*
- *Journal of Computational Geometry*
- *Journal of Differential Geometry (http://intlpress.com/site/pub/pages/journals/items/jdg/_home/_main)*
- *Journal of the ACM*
- *Journal of Algorithms*
- *Journal of Computer and System Sciences*
- *Management Science*
- *Pattern Recognition*
- *Pattern Recognition Letters*
- *SIAM Journal on Computing*
- *SIGACT News*; featured the "Computational Geometry Column" by Joseph O'Rourke
- *Theoretical Computer Science*
- *The Visual Computer*

External links

- Computational Geometry (http://www.computational-geometry.org/)
- Computational Geometry Pages (https://web.archive.org/web/20111106212300/http://compgeom.cs.uiuc.edu/~jeffe/compgeom/)
- Geometry In Action (http://www.ics.uci.edu/~eppstein/geom.html)
- "Strategic Directions in Computational Geometry—Working Group Report" (1996) (http://www.cs.brown.edu/people/rt/sdcr/report/report.html)
- Journal of Computational Geometry (http://jocg.org/)
- (Annual) Winter School on Computational Geometry (https://web.archive.org/web/20140106103137/http://cg.aut.ac.ir/wscg)
- Computational Geometry Lab (https://cglab.ca/)