Applications of computational geometry R. Inkulu

- robotics
    * path planning, while in turn relying on visibility
    * positioning and fixing some parts (arms) of a robot so that it can work on a part being manufactured
- computer graphics
    * hidden surface removal
    * determining the geometric primitive pointed to with the mouse in a collection of objects being displayed
    * assessing the collision between geometric objects
- data visualization
    * drawing a graph on the plane with minimum number of crossings
- geographic information systems
    * storing and efficiently extracting geographic data
    * digitizing a paper map leading to digitized boundaries; determine whether two digitizations obtained for the same map are same
    * locating the nearest public phone
    * map overlay
- computer aided design
    * geometric modeling to represent real-world objects
    * intersection and union of objects
    * simulating heat emission of a printed circuit board via meshing
    * simulating to calculate stress at every point of a structure via meshing
- computer aided manufacturing
    * machining along a tour that touches a set of points
    * design a mold so that the manufactured product can be removed from the mold by a single translation
- electronic design automation
    * packing a set of chips onto a mother board while accommodating pairwise connections between chips
    * partitioning an orthogonal polygon into (apprx) minimum number of rectangles
    * form a simple polygon of smallest perimeter with a given set of points as its vertices
- wireless network design
    * placing a minimum number of sufficiently powered antennas to cover every point of every region of interest
- pattern recognition
    * match the image i of a symbol against a collection of stored symbols so that to identify i

- machine learning
    * partition data points into clusters; and, to determine outliers

- data analysis
    * store data points so that to efficiently answer range queries

- data science
    * (approximate) nearest neighbors
    * dimension reduction

- computational learning theory
    * ϵ-nets and VC-dimension

- guarding
    * survey a museum using minimum number of guards

- molecular modeling
    * compute the outer surface of a homogeneous mixture formed by molecules, wherein molecules of it are represented by balls with a few pairwise intersections

- efficient communication
    * level-of-detail based information sharing

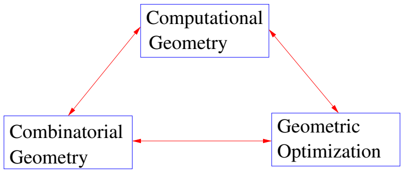
- computational statistics
    * determine how central the given point is inside a cloud of points

- linear programming
    * notions of convexity and polytopes to find an optima

- rigidity analysis
    * given a discrete configuration of a set of points, position them on a plane while satisfying the given set of pairwise distances; also, determine whether the set of solutions is unique

- crystallography
    * measuring the similarity between two arbitrary crystal structures

etc.



The objective of this course is to devise efficient data structures and algorithms to several discrete geometric problems as well as optimization problems that commonly occur in these domains. Besides, we also touch upon analyzing elementary combinatorics associated with these problems. Algorithmic solutions (i.e., with provable guarantees) are established for several evident problems with sets of points, lines, line segments, and simple polygons in the plane.

Model of computation

The *real RAM model of computation*, which allows each memory location to hold a single real number, and does each of the following primitive operations in $O(1)$ time:

- arithmatic $(+, -, *, /)$
- comparison $(<, \leq, \neq, \geq, >)$
- indirect addressing with integer address
- k^{th} root, logarithm to the base 2, $sine$, and $cosine$

Motivation: This model (esp., in comparison to the usual word-RAM model of computation) serves to get rid off (i) round-off errors in the approximate representation of real numbers, and (ii) the irrational numbers that could result from algebra with real numbers.

General position assumptions

To avoid handling degeneracies, in this course, we assume the following:

- no three points are collinear
- no two points have the same x_i^{th}-coordinate, for any i
- no four points are cocircular

— depending on the problem, some more could be included