COMPUTATIONAL GEOMETRY

Anil Maheshwari
School of Computer Science
Carleton University
E-mail:-*maheshwa@scs.carleton.ca*

95.508

WINTER 1996-97

What is Computational Geometry?

Design & Analysis of Algorithms for Geometric Problems.
(Preparata-Shamos).

Study of sorting and searching problems in Higher Dimensions
(Mulmuley).

Application Areas

- Computer Graphics
- VLSI
- Robotics
- Geographic Informations Systems
- CAD
- Data Bases

- Motion Planning

- Vector Quantization

Look into WEB PAGES

History & Development

- Egyptians/Greeks.
- Euclid (Axiomatic Framework; Constructive Proofs)
- Descartes (Coordinate System)
- Lemoine (Complexity Notions, 1902)
- Hilbert (Lower Bounds)
- Klein (Geometric Transformations and Projective Geometry)

- Geometric Modeling using Spline Curves (1970s)
- Perceptrons – Complexity of predicates recognizing geometric objects (Minsky/Papert 1969)
- Graphics Software/Geometric Editors
- Computational Geometry, Ph.D. Thesis by M.I. Shamos, 1975.

Problem Areas (O'Rourke)

1. Triangulation and Partitioning
2. Convex Hulls
3. Voronoi Diagrams and Delaunay Triangulations
4. Arrangements
5. Search
6. Motion Planning and Visibility

Techniques

1. Plane Sweep
2. Randomized Incremental Algorithms
3. Random Sampling
4. Divide and Conquer
5. Fractional Cascading

Algorithmic Development

- Deterministic Algorithms 1975.
- Dynamic Algorithms
- Randomized Algorithms 1980
- Derandomization

– 2 & 3 Dimensional

– Application and Real Application Motivated

References

1. Preparata & Shamos: Introduction to Computational Geometry, Springer-Verlag..

2. Mulmuley: Computational Geometry: An Introduction through Randomized Algorithms.

3. O'Rourke: Art-Gallery Theorems and Algorithms, Cambridge University Press.

4. Mount: Class Notes, University of Maryland, College Park.

5. Several Research Articles.

BackGround Requirement

1. 95.384.
2. Notations: O, o, Ω, Θ.
3. Lower Bounds.
4. Data Structures: Priority Queue, Segment Trees.

DCEL Data Structure

Muller & Preparata, 1978

- To represent connected planar graphs embedded in the plane.

- Graph:: $G = (V, E)$

- Vertices:: V=$\{v_1, v_2, \cdots, v_n\}$

- Edges:: E=$\{e_1, e_2, \cdots, e_m\}$

$\forall e_i \in E$

$$e_i = (v_1, v_2, f_1, f_2, p_1, p_2)$$

- v_1 & v_2 are tail and head vertices.

- f_1 & f_2 are left and right faces to this edge.

- p_1 & p_2 are pointers to clockwise edges.

Sorting and Searching

Input: A set N of n-points on the real line R.

Sorting Problem: Find the partition $H(N)$ of R formed by the given set of points N.

Searching Problem: Associate a search structure $\tilde{H}(N)$ with $H(N)$ so that, given any point $q \in R$, one can locate the interval in $H(N)$ containing q quickly, i.e., in logarithmic time.

Dynamic Variant: Set N could be dynamic.

Higher Dimensional Sorting/Searching Problem

$\Re^d$: d-dimensional Euclidean Space

Let $H(N)$ denote the partition of the whole or part of $\Re^d$ induced by N.

Example 1: N= Set of lines and $H(N)$= Partition of the plane.

Example 2: N= Set of hyperplanes.
$H(N)$= Subfaces of the hyperplanes forming the convex polytope.

In general, $H(N)$= $Geometric\ Complex$.

i.e., Collection of disjoint regions in $\Re^d$ of varying degree, e.g.,
"0"- dimension $\rightarrow$ Vertices.
"1"- dimension $\rightarrow$ Edges.

Higher Dimensional Sorting/Searching Problem

Sorting Problem: Given a set N of objects in $\Re^d$, construct the induced geometric complex $H(N)$ quickly.

Searching Problem: Construct a search structure $\tilde{H}(N)$ that can be used to answer queries over N quickly.

Example: Point location queries over the partition induced by a set N.

Classical Problems: Range Queries

N= Set of objects in $\Re^d$.

Query= A connected region in $\Re^d$.

Search Problem: Report/Count all objects intersected by the query.

Applications in Databases, Speech processing, computer graphics, etc.

Example: N= Set of points. Query= Half-Space or a Simplex.

Classical Problems: Arrangements

N= Set of lines in the plane.

- lines divide the plane into cells (2-faces)

- lines are divided into intervals (1-face/edge)

- intersection are vertices (0-face)

Sorting Problem: Given N, construct the arrangement $H(N)$.

Searching Problem: Construct a search structure $\tilde{H}(N)$ that can be used to locate the face of $H(N)$ contating a given query point quickly.

Classical Problems: Trapezoidal Decomposition

N= Set of segments.

$H(N)$= Trapezoidal Decomposition.

Sorting Problem: Given a set N of segments in the plane, construct the trapezoidal decomposition $H(N)$.

Searching Problem: Construct a search structure $\tilde{H}(N)$ so that the face of $H(N)$ containing any query point can be located quickly.

Classical Problems: Convex Polytopes

Sorting Problem: Given a set of half-spaces, construct the facial lattice of the convex polytope formed by them.

Search Problem: (e.g. Linear Programming Problem)

Given a set N of half-spaces and a linear objective function z, find the coordinate of a z-maximum on $H(N)$.

Build a search structure $\tilde{H}(N)$, so that any given linear function z, a z-maximum on $H(N)$ can be found quickly.

Classical Problems: Duality

Convex Polytope Problem: $N =$ *Half Spaces and* $H(N) =$ *Convex Polytope*

Convex Hull Problem: $\hat{N} =$ *Points and* $H(\hat{N}) =$ *Convex Hull*

These problems are Dual of each other!!

Assume that the origing lies strictly in the interior of $H(N)$.

Consider the following transformation that takes a point to a line or vice-versa.

$$T : (p_1, p_2, \cdots, p_d) \rightarrow p_1 x_1 + \cdots + p_d x_d = 1$$

Classical Problems: Voronoi Diagrams

N= Sites (points) in $\Re^d$.

Post-Office Problem: Build a search structure so that, given any query point p, the site in N nearest to p can be located quickly.

Voronoi Region of a site $S \in N$ is defined to be the region consisting of the points in $\Re^d$ whose nearest neighbor in N is S.

Sorting Problem: Given a set of sites N, construct its Voronoi Diagram $H(N)$.

Search Problem: Construct a search structure $\tilde{H}(N)$ so that, given a query point p, the nearest site of p can be located quickly.

Classical Problems: Hidden Surface Removal

N= Set of opaque polygons.

$H(N)$= Visibility Map, i.e., what is visible and what portion.

Sorting Problem: Construct the visibility map.

Classical Problems: Path Problems

N= Constrained Regions.

$H(N)$= Euclidean Shortest Path, Minimum Link Path, Weighted Shortest Paths, etc.

Sorting Problem: Given N, compute ESP, MLP.

Searching Problem: Compute the Search Structure $\tilde{H}(N)$ such that shortest paths between query points can be answered quickly.

Numerical Precision and Degeneracies

- Real RAM Model of Computation.
- Each memory word can hold an *arbitrary long precision real numbers*.
- Each basic operation (add, divide, subtract, multiply) is a *unit cost operation*.

Marking Scheme

1. Seminar

2. Report

3. Finals