CS-235
Computational Geometry

Subhash Suri

Computer Science Department
UC Santa Barbara

Fall Quarter 2002.
Computational Geometry

- Study of **algorithms for geometric problems**.

- **Deals with discrete shapes**: points, lines, polyhedra, polygonal meshes.

- **Abstraction** of problems in different applied areas.

- **Occlusion, visibility, augmented reality, collision detection, motion or assembly planning, drug design, databases, GIS, layout, fluid dynamics, etc.**
CG and Computer Science

- **CG is a sub-discipline of** algorithms and complexity.

  - **Computability**
    - Hilbert
    - Godel 1900–1940s
    - Turing

  - **Algorithms and Complexity**
    - Rabin 1950s–present
    - Cook/Levin/Karp
    - Knuth
    - Aho–Hopcroft–Ullman–Tarjan

  - **Computational Geometry**
    - Shamos’ Thesis (1975)
    - 1980–present

- Develops fundamental techniques and tools for geometric problems.
- **Motivated** by applications in other CS fields.
- **Significant** interaction with discrete mathematics.
Some Examples

- **Range Searching Data Structures.**

- **Location Queries.**
Some Examples

- Decomposition.

- Is this always possible?

- In three dimension?

- Other examples: Shortest paths, geometric structures, visibility, pattern matching.
Some Examples

• Spatial Data Structures.

• Voronoi diagram, Delaunay triangulation.

• Robot motion planning.
Taste of Comb. Geometry

- **Helly’s Theorem:** Let $C_1, \ldots, C_n$ be a family of convex sets in the plane. If every triple intersects, then $\bigcap C_i$ is non-empty.

- **Center Points:** Given points $p_1, p_2, \ldots, p_n$ in the plane, a point $x$ is called center point if any line through $x$ contains at least $n/3$ points on each side.

- **Ham Sandwich Theorem:** Take $n$ red points and $n$ blue points in the plane. There is a line simultaneously bisecting both red and blue points.

- **Crossing Number Theorem:** If $G$ is a graph with $n$ nodes and $m$ edges, then every drawing of $G$ in the plane contains at least $c \left( \frac{m^3}{n^2} \right) - n$ crossings.
Among any 5 points in the plane in general position, we can find 4 forming a convex polygon.

Erdős-Szekeres Theorem: For every positive integer $k$, there exists a number $F_k$, such that every set of $F_k$ points in the plane contains $k$ that form a convex $k$-gon.

Empty $k$-gon: How large must the set be to guarantee that we can find $k$ points forming a convex polygon, which does not include any other point inside?

Empty $k$-gon: Known values:

$G_3 = 3$.
$G_4 = 5$.
$G_5 = 10$.
$G_6 = ???$.
$G_7 = \infty$!
Spirit of CG

1. **CG** is a product of marriage between classical geometry and computer science.

2. Emphasis on design of efficient algorithms and data structures.

3. In classical approach, reducing the complexity to a finite number of choices was enough.

4. Alas! $10^{100}$ is mathematically finite but computationally infinite.

5. Point of Reference: 1 Year $\approx 3.15 \times 10^7$ seconds.

6. 1 Century $\approx \pi \times 10^9$ seconds.

7. A 1-Giga flop computer does only $10^{20}$ ops in a century!
Model of Computation

1. Assume an abstract programming language model.


3. Each memory cell can hold one int or real geometric coordinate. We will discuss later numerical precision issues.

4. Standard repertoire of operators:
   - Arithmetic: $+, -, \div, *, \sqrt{\cdot}$.
   - Trigonometry: $\sin, \cos, \tan, \exp, \log$.
   - Comparators: $\leq, \geq, =$.
   - Array indexing, pointers.
Elementary Objects

1. **Point** $p = (x, y)$, where $x, y$ reals.

2. **Line** $\ell := ax + by = 1$.

3. **Line segment** $s = [p, q]$.

4. **Circle** $C = (p, r)$. (Center, radius)

5. **Polygon** $< p_1, p_2, \ldots, p_n >$. 

Line Segment  Circle  Polygon
1. Algorithm receives algebraic input. It must perform computation to see the underlying geometric relationships.

2. Is point $p$ on line $\ell$?

3. Is point $p$ inside or outside circle $C$?

4. Do segments $s_1$ and $s_2$ intersect?

5. Is point $p$ inside or outside polygon $P$?
Overview of the Course

1. Convex Hulls.

2. Intersection Detection and Reporting.

3. Triangulation.

4. Range Searching.

5. Point Location.

6. Delaunay Triangulation.


8. Arrangements.


10. Epsilon Net and VC Dimension.

11. Volume and paradoxes in higher dimensions.

12. Misc.