PRESENTATION OF (THE SECOND HALF OF) THE COURSE

Rodrigo Silveira

Discrete and Algorithmic Geometry
Facultat de Matemàtiques i Estadística
Universitat Politècnica de Catalunya

One course about two (related) topics

One course about two (related) topics

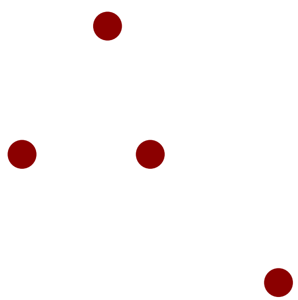
Discrete Geometry | Algorithmic Geometry

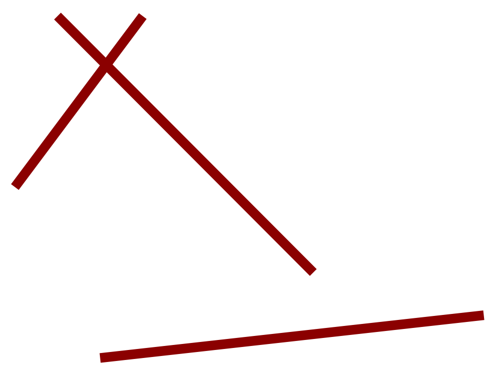
One course about two (related) topics

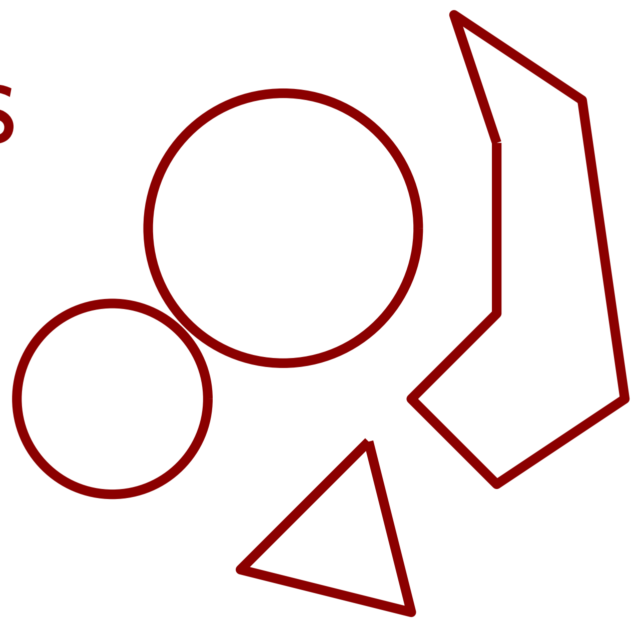
Discrete Geometry

Algorithmic Geometry

Geometric objects







DAG: Discrete and Algorithmic Geometry

One course about two (related) topics

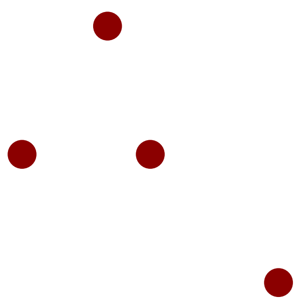
Discrete Geometry

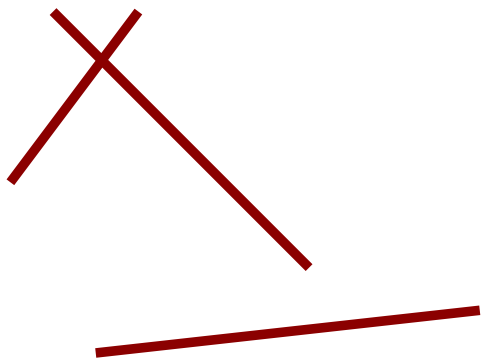
study of combinatorial properties of discrete geometric objects

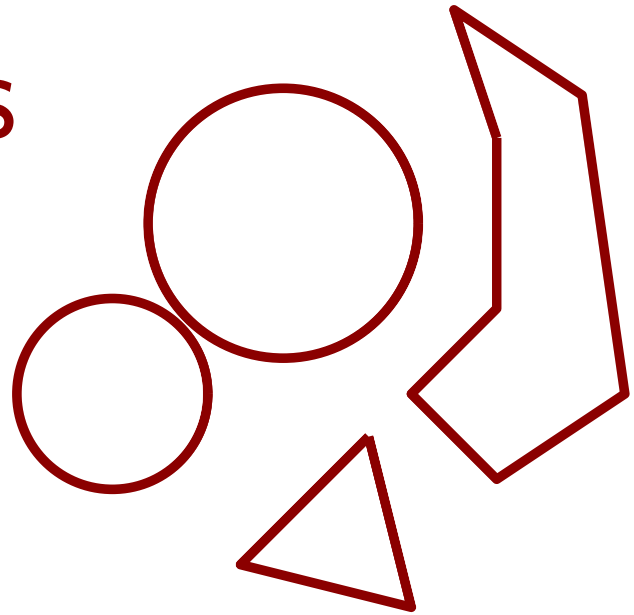
Algorithmic Geometry

design of algorithms that deal with geometric objects

Geometric objects







A course with two parts

Discrete Geometry

Algorithmic Geometry

A course with two parts

Discrete Geometry	Algorithmic Geometry
Part I	Part II

A course with two parts

Discrete Geometry

Part I

Instructor:
Julian Pfeifle

Algorithmic Geometry

Part II

Instructor:
Rodrigo Silveira

Check out the web of our research group: https://dccg.upc.edu

Introduction to algorithmic geometry

Introduction to algorithmic geometry

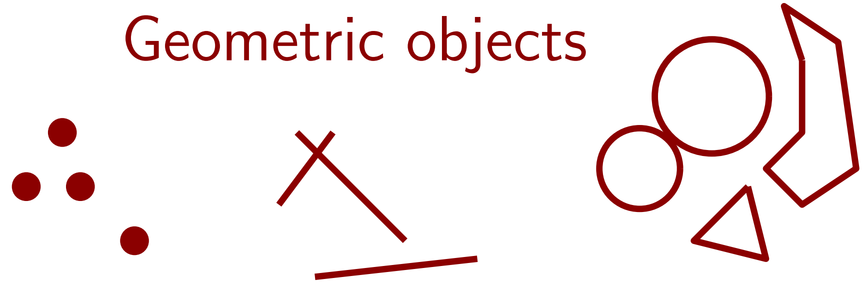
Goal: design and analysis of efficient algorithms to solve geometric problems

Introduction to algorithmic geometry

Goal: design and analysis of efficient algorithms to solve geometric problems

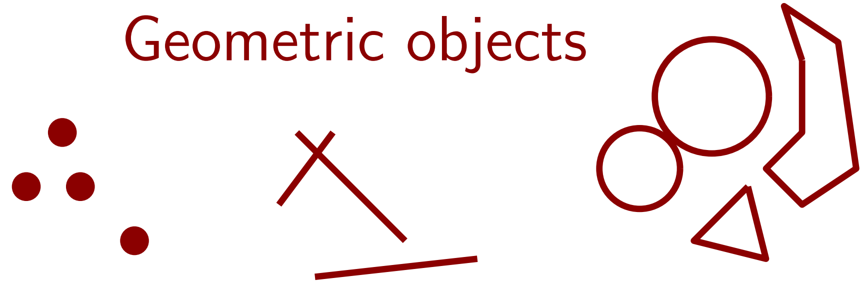
Introduction to algorithmic geometry

Goal: design and analysis of efficient algorithms to solve geometric problems



Introduction to algorithmic geometry

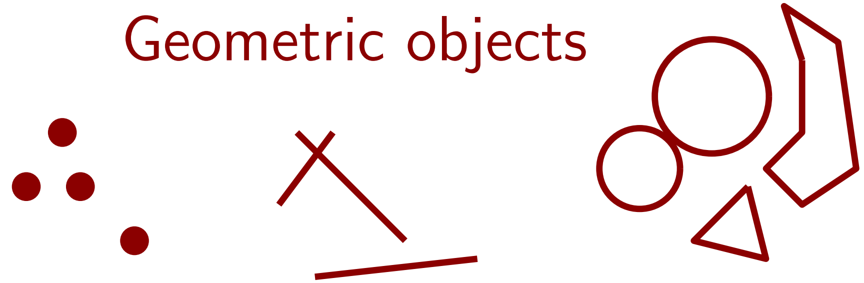
Goal: design and analysis of efficient algorithms
to solve geometric problems



- Keep in mind: *Algorithmic geometry*, as a field, is more well-known as *Computational Geometry*

Introduction to algorithmic geometry

Goal: design and analysis of efficient algorithms to solve geometric problems

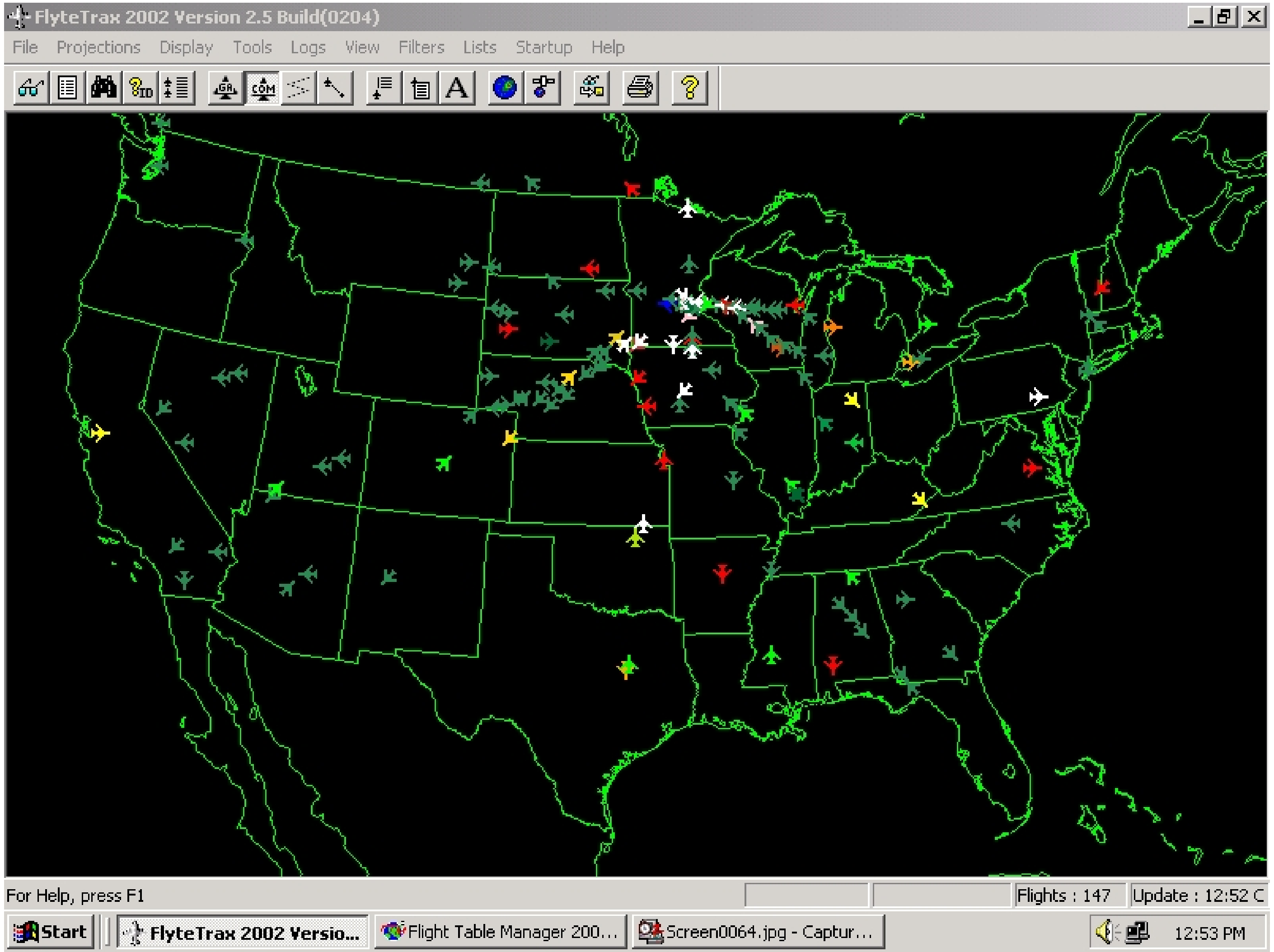


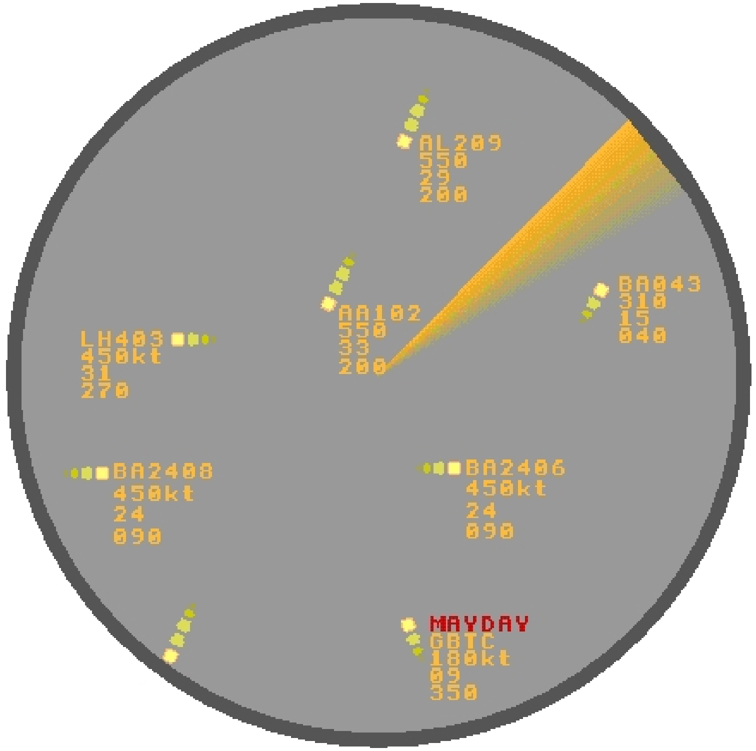
- Keep in mind: *Algorithmic geometry*, as a field, is more well-known as *Computational Geometry*

Let's take a look at some examples of applied geometric problems

AIR TRAFFIC CONTROL

Detect the pair of airplanes that are in most imminent danger of collision among those who show up on the screen of an air controller





What are the geometric objects here?

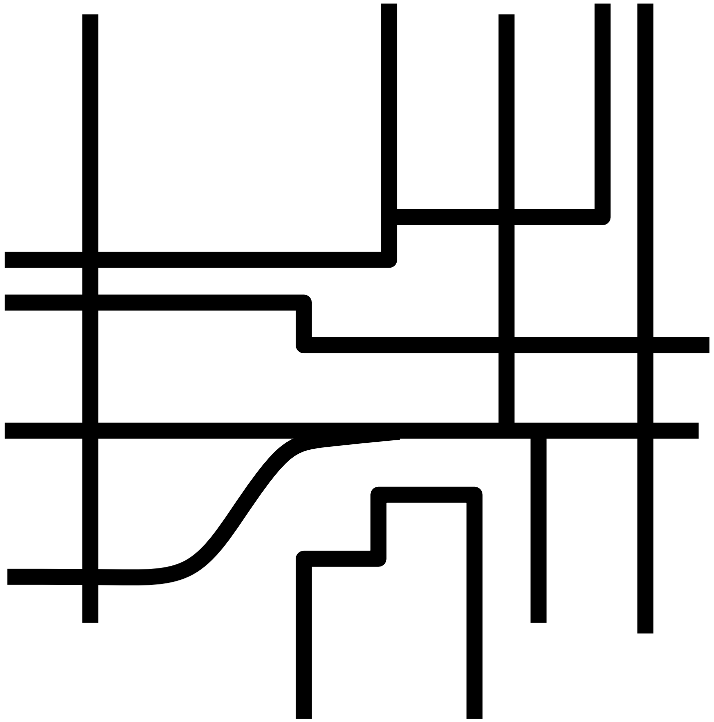
ROAD NAVIGATION (with GPS)

ROAD NAVIGATION (with GPS)

How does your GPS navigator know which objects (streets, places of interest, etc.) to show?

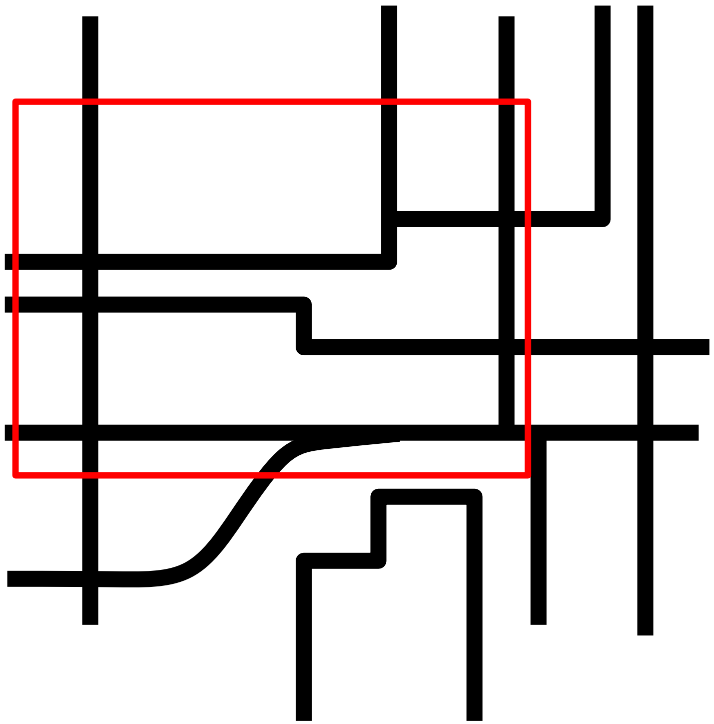
ROAD NAVIGATION (with GPS)

How does your GPS navigator know which objects (streets, places of interest, etc.) to show?



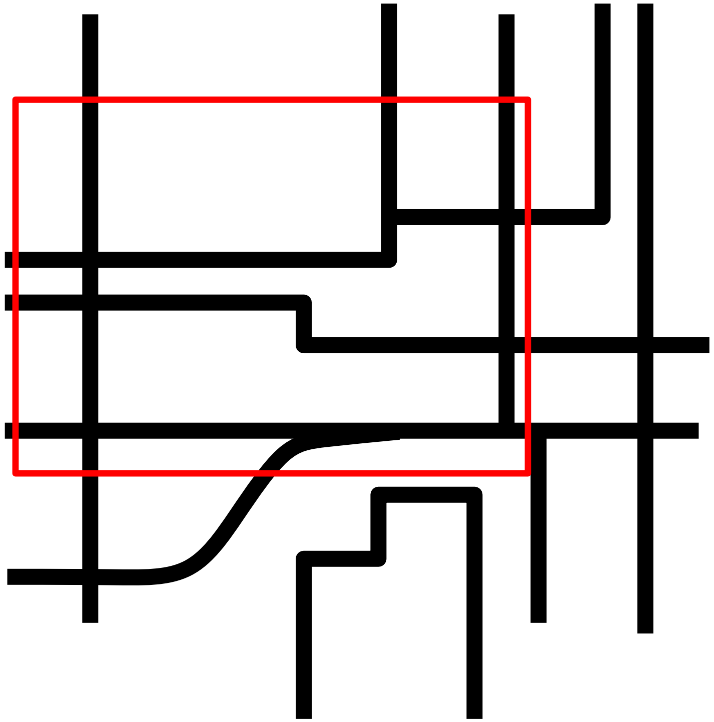
ROAD NAVIGATION (with GPS)

How does your GPS navigator know which objects (streets, places of interest, etc.) to show?



ROAD NAVIGATION (with GPS)

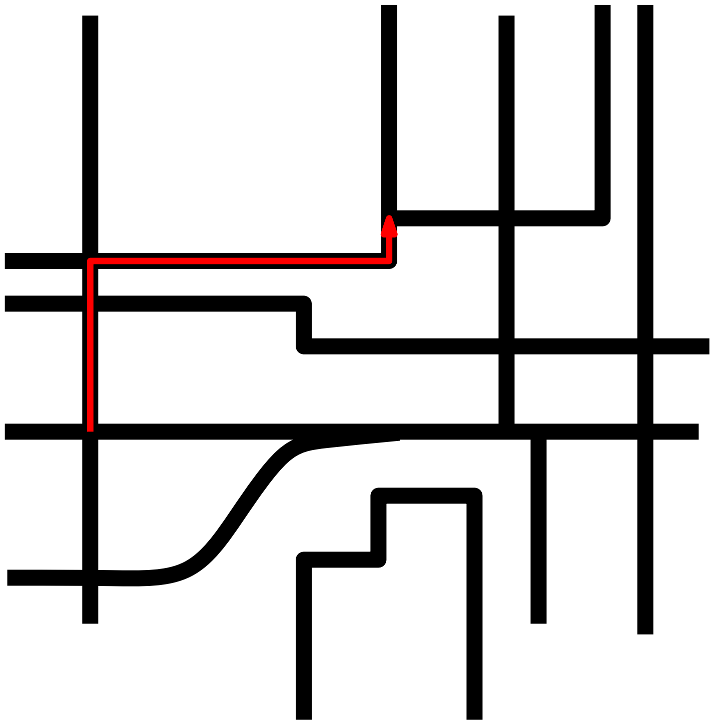
How does your GPS navigator know which objects (streets, places of interest, etc.) to show?



This geometric problem is called *windowing*

ROAD NAVIGATION (with GPS)

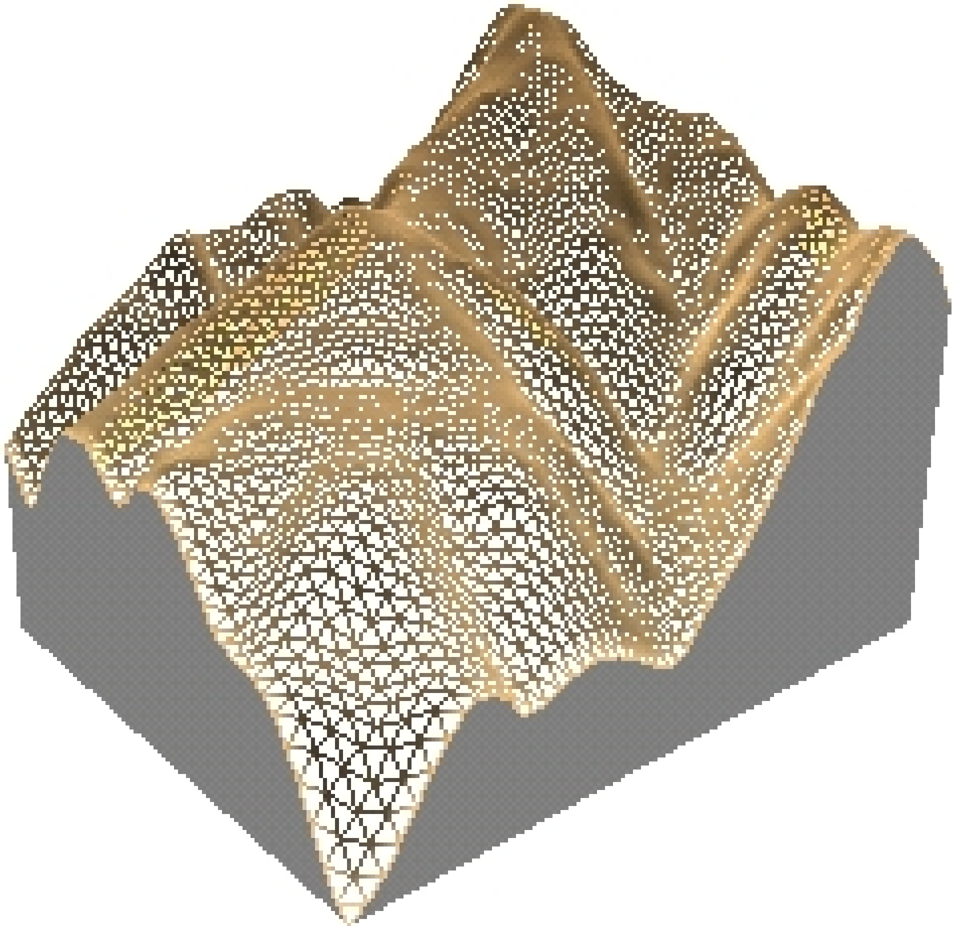
How does it compute the best way to get from one place to another?

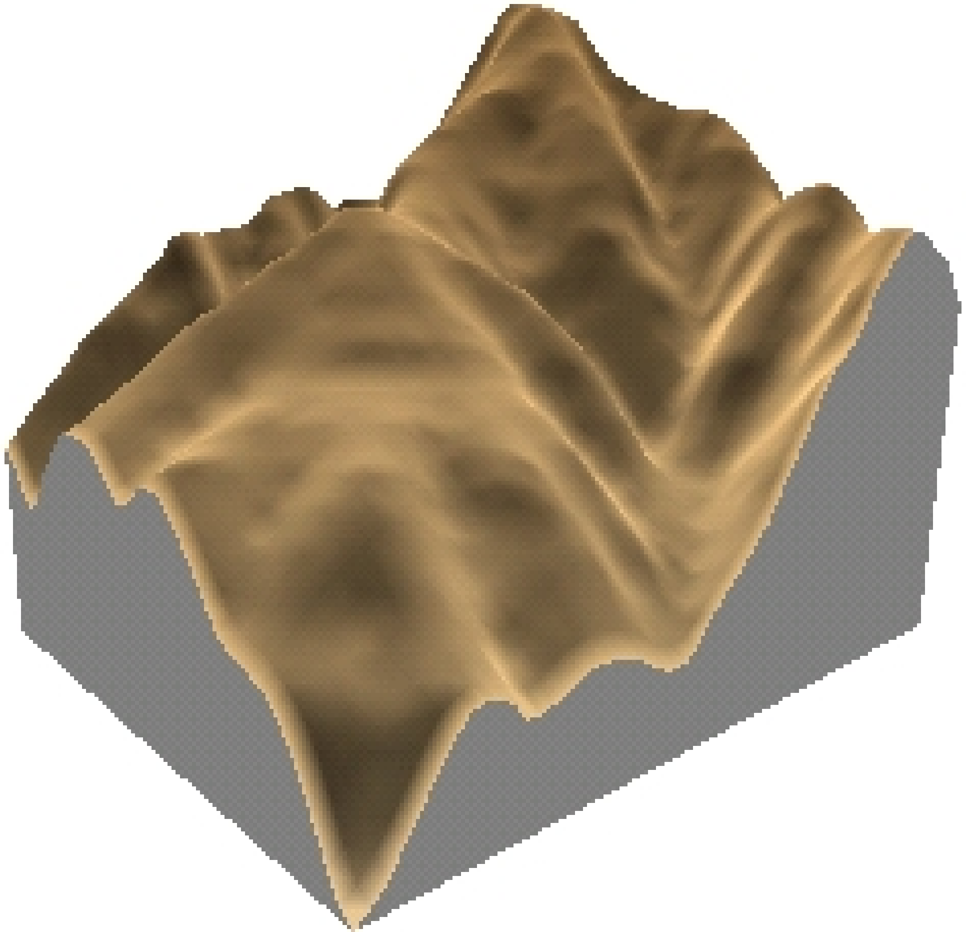


This is a type of geometric *shortest path problem*

TOPOGRAPHICAL DATA INTERPOLATION

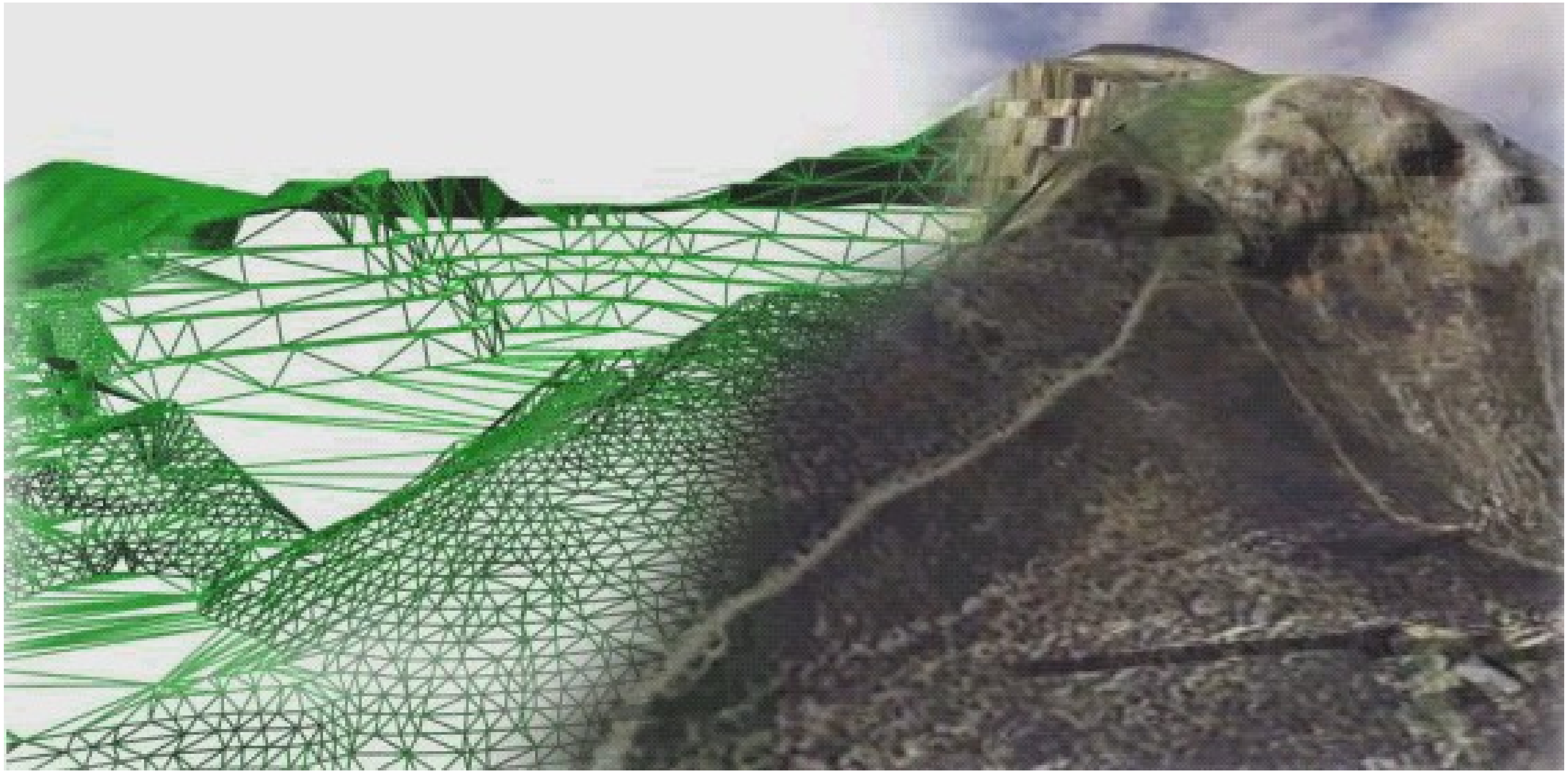
From the altitude data of a sample of points in a terrain, obtaining an approximation of the terrain as a continuous function, by interpolating the values of the altitude of the remaining points





TOPOGRAPHICAL DATA INTERPOLATION

From the altitude data of a sample of points in a terrain, obtaining an approximation of the terrain as a continuous function, by interpolating the values of the altitude of the remaining points

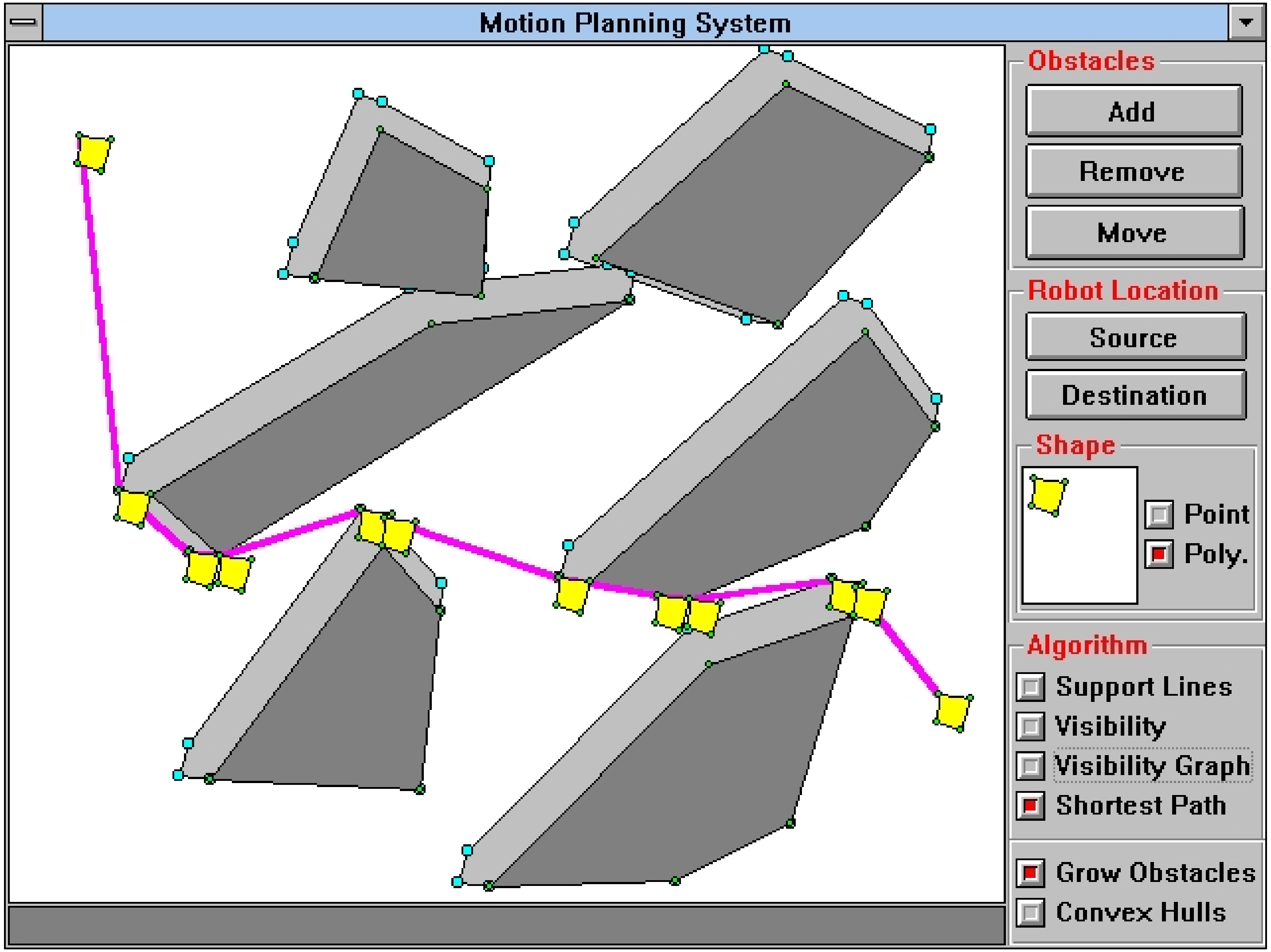


GRAPHICS: HIDING NON-VISIBLE PORTIONS OF A SCENE

Represent a realistic scene, by showing on the screen only the portions of the scene which are visible from the viewpoint

ROBOTICS: MOTION PLANNING PROBLEM

Find the shortest path between two points, avoiding obstacles



ROBOTICS: MOTION PLANNING PROBLEM

Find the shortest path between two points, avoiding obstacles

ROBOTICS: MOTION PLANNING PROBLEM

Find the shortest path between two points, avoiding obstacles

ROBOTICS: MOTION PLANNING PROBLEM

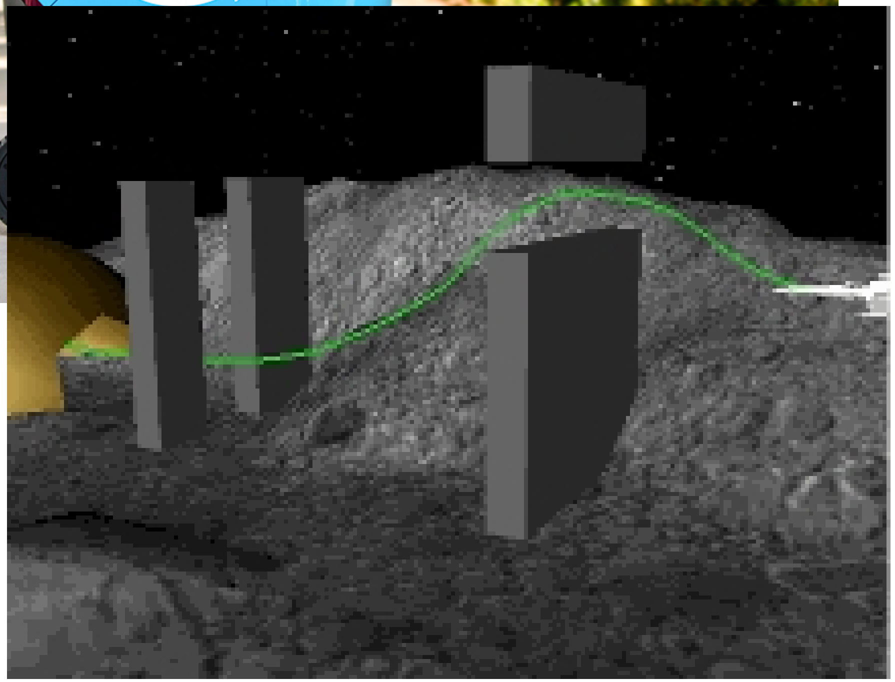
Find the shortest path between two points, avoiding obstacles

Scout, Amazon's self-driving delivery robot

ROBOTICS: MOTION PLANNING PROBLEM

Find the shortest path between two points, avoiding obstacles

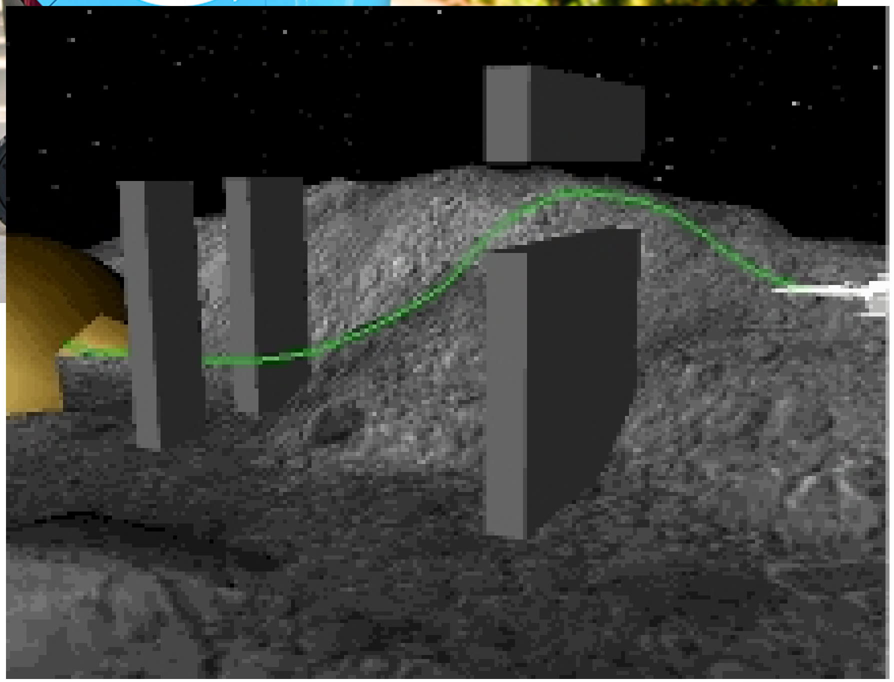
Scout, Amazon's

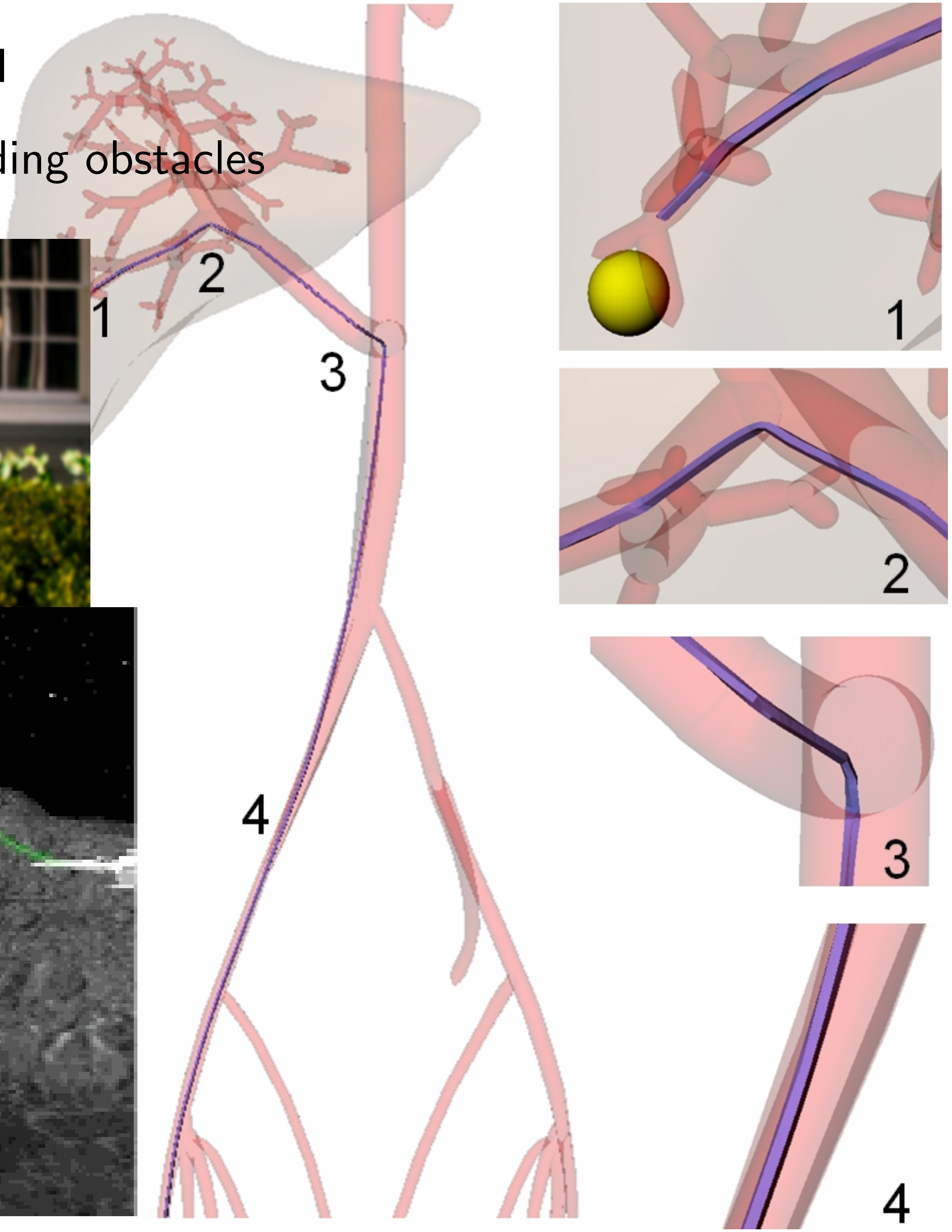


ROBOTICS: MOTION PLANNING PROBLI

Find the shortest path between two points, avoiding obstacles

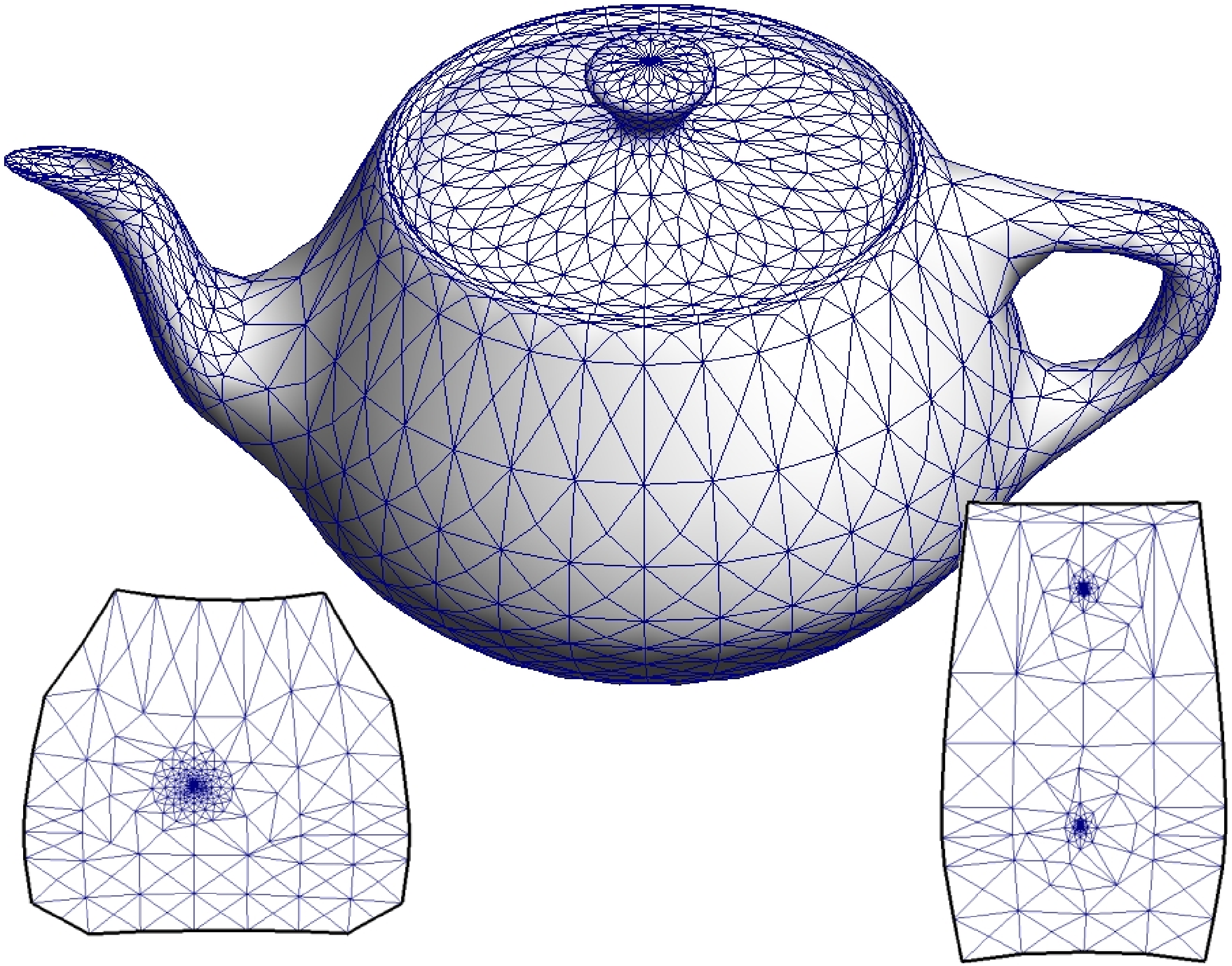
Scout, Amazon's





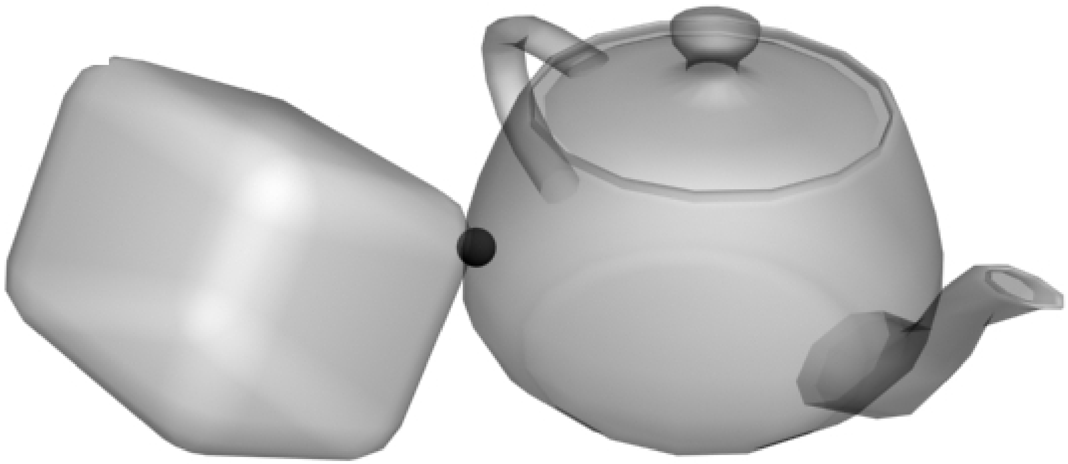
CAD and CAM: MODELING OBJECTS

Design geometric objects by an appropriate discretization. Store the geometric information in a structured and efficient way



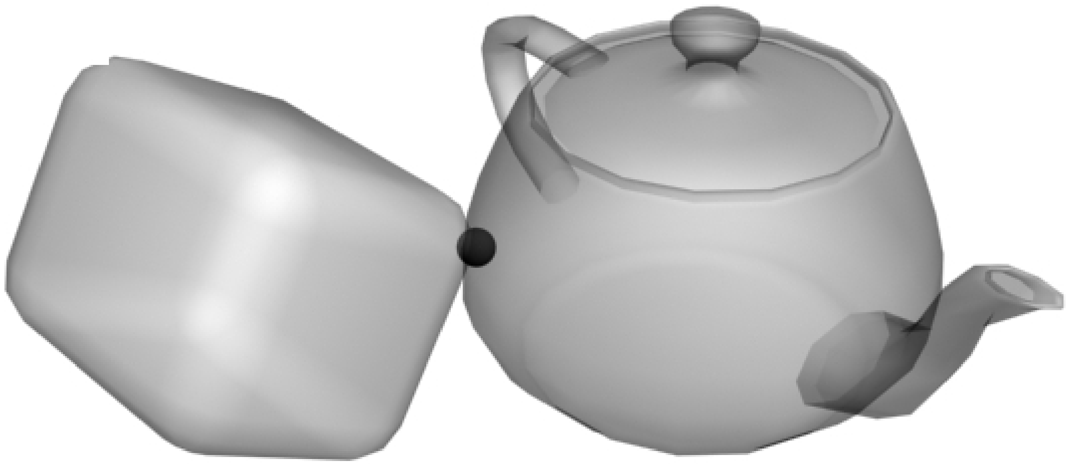
COLLISION DETECTION

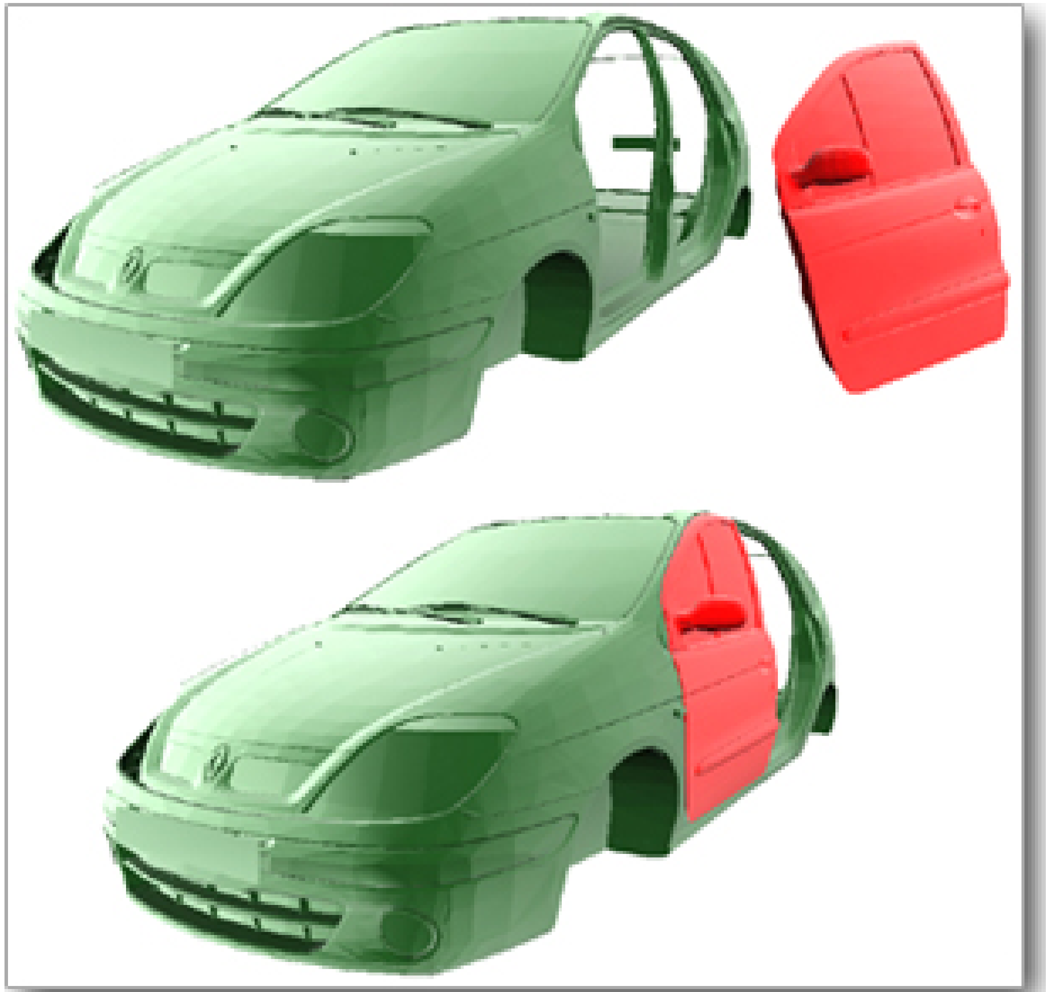
Detect whether or not the intersection of two objects in the plane or in 3D space is empty



COLLISION DETECTION

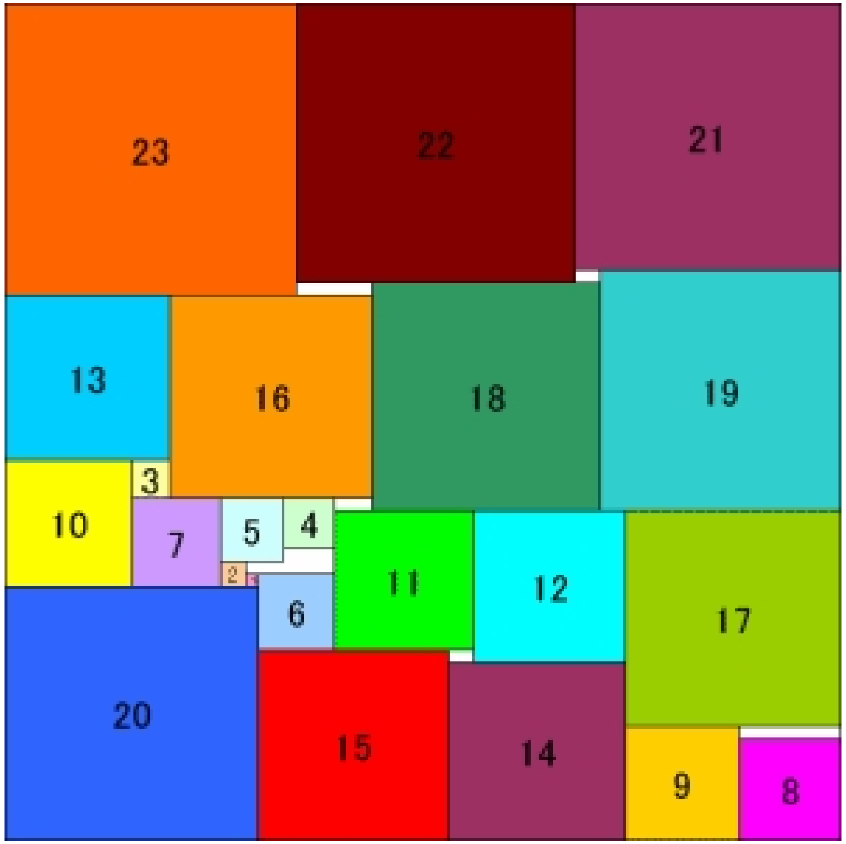
Detect whether or not the intersection of two objects in the plane or in 3D space is empty





PACKING

Given a set of objects (luggage), decide whether or not it is possible to pack them in a given space (trunk) and, in the affirmative, compute how this can be done



Applications of Algorithmic Geometry

FIELDS OF APPLICATION

- Computer Graphics: realistic visualization, modellling, ...
- Automated processes: computer vision, voice recognition, automatic reading, robotics, ...
- Geographics information systems, air traffic control
- Design and manufacturing
- 3D reconstruction from 2D information
- Molecular biology
- Astrophysics
- VLSI
- Statistics, operations research
- ...

Geometric problems

The problems posed in the previous applications have some elements in common:

- Geometric nature of the information
- The geometric problem is discrete
- Big amount of data: requires efficient solutions (in time and space)

Geometric problems

The problems posed in the previous applications have some elements in common:

- Geometric nature of the information
- The geometric problem is discrete
- Big amount of data: requires efficient solutions (in time and space)

> Algorithmic Geometry: design and analysis of efficient algorithms to solve geometric problems

Geometric problems

The problems posed in the previous applications have some elements in common:

- Geometric nature of the information
- The geometric problem is discrete
- Big amount of data: requires efficient solutions (in time and space)

> Algorithmic Geometry: design and analysis of efficient algorithms to solve geometric problems

Typical steps in solving a geometric problem

- Analyze the problem and understand its geometric component
- Discretize the problem (if it is not discrete)
- Exploit the geometric characteristics of the problem
- Find efficient algorithms
- Store in appropriate data structures

Example of a *clean* geometric problem

Example of a *clean* geometric problem

Given n points in the plane (e.g., villages) find the optimal location of a service to attend this population (antenna, hospital, supermarket,...).

Example of a *clean* geometric problem

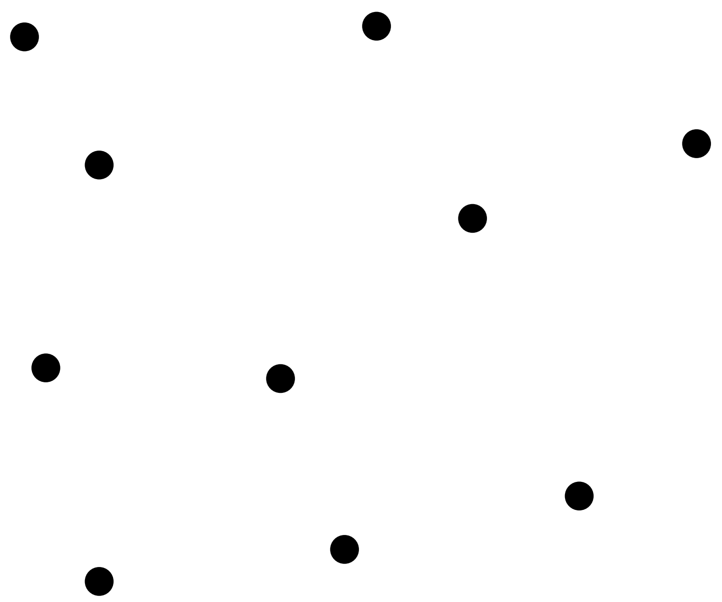
Given n points in the plane (e.g., villages) find the optimal location of a service to attend this population (antenna, hospital, supermarket,...).

Given $(x_1, y_1), \ldots (x_n, y_n)$, find (x, y) achieving $\min_{(x,y)\in\mathbb{R}^2} \max_{i=1\ldots n} (x - x_i)^2 + (y - y_i)^2$.

Example of a *clean* geometric problem

Given n points in the plane (e.g., villages) find the optimal location of a service to attend this population (antenna, hospital, supermarket,...).

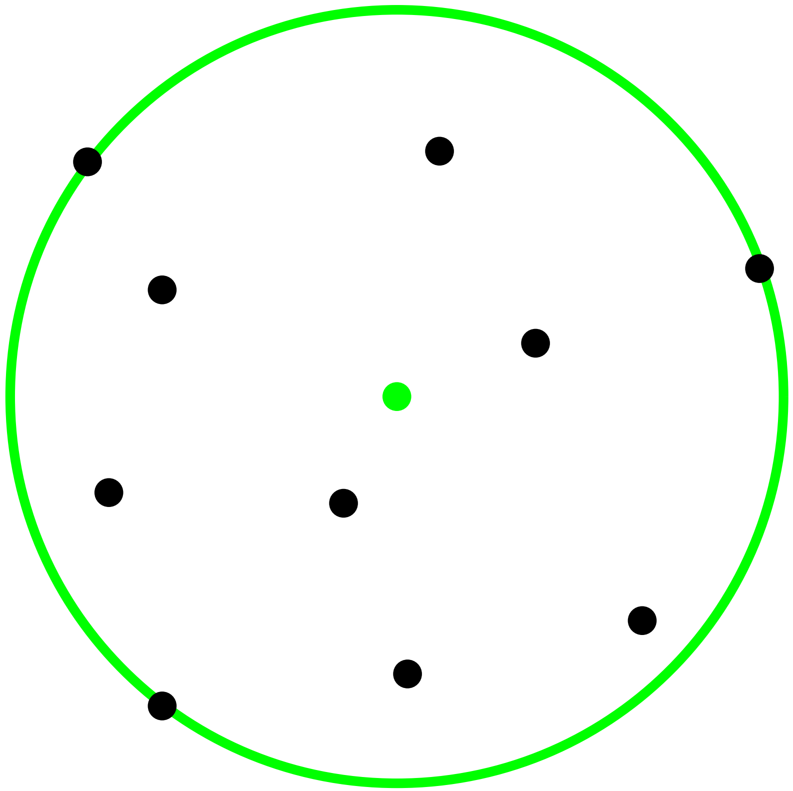
Given $(x_1, y_1), \ldots (x_n, y_n)$, find (x, y) achieving $\min_{(x,y)\in\mathbb{R}^2} \max_{i=1\ldots n} (x - x_i)^2 + (y - y_i)^2$.



Example of a *clean* geometric problem

Given n points in the plane (e.g., villages) find the optimal location of a service to attend this population (antenna, hospital, supermarket,...).

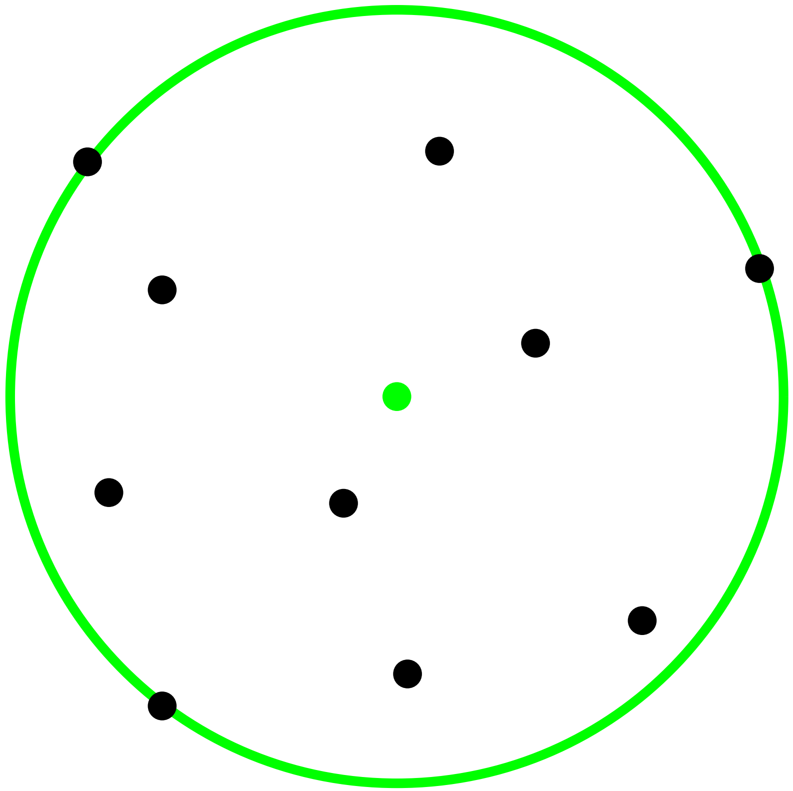
Given $(x_1, y_1), \ldots (x_n, y_n)$, find (x, y) achieving $\min_{(x,y)\in\mathbb{R}^2} \max_{i=1\ldots n} (x - x_i)^2 + (y - y_i)^2$.



Example of a *clean* geometric problem

Given n points in the plane (e.g., villages) find the optimal location of a service to attend this population (antenna, hospital, supermarket,...).

Given $(x_1, y_1), \ldots (x_n, y_n)$, find (x, y) achieving $\min_{(x,y)\in\mathbb{R}^2} \max_{i=1\ldots n} (x - x_i)^2 + (y - y_i)^2$.



This problem is difficult to solve using analytical techniques.

But it can be discretized.

This allows an algorithmic solution.

The cost of the algorithm depends upon the number of input points.

A brief history of computational geometry

Name initially used in different contexts

- In Minsky's book "Perceptrons" (1969) it meant pattern recognition.
- In Forrest paper (1971) it meant curves and surfaces for geometric modeling.
- Shamos' PhD Thesis "Computational geometry" (1975) uses the name for the first time as we understand it today:

Design and analysis of **efficient algorithms** to solve **geometric problems**.

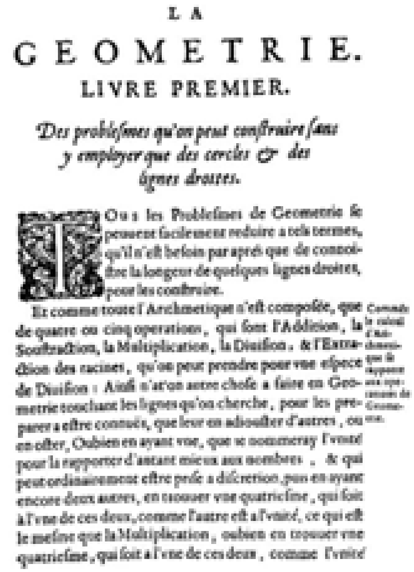
Better names are possible (and are sometimes used, like in this course!)

- Algorithmic geometry
- Geometric algorithms

A brief history of computational geometry

How it started

- Geometric algorithms have been around for a while
  - Euclid constructions in The Elements (300 BC)
  - Descartes Cartesian geometry (17th century)
- Computers brought renewed interest
  - 50's first graphics program (for hidden line removal)
  - First CAD (Computer-aided design) programs
- ... and modified its characteristics
  - massive amount of data
  - need of efficiency



LA

GEOMETRIE.

LIVRE PREMIER.

Des problesmes qu'on peut construire sans y employer que des cercles & des lignes droites.

TOus les Problesmes de Geometrie se peuvent facilement reduire a tels termes, qu'il n'est besoin par aprés que de connoistre la longeur de quelques lignes droites, pour les construire.

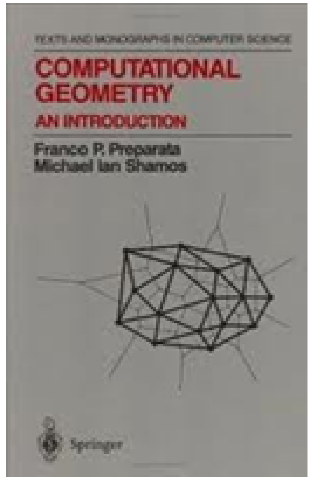
Et comme toute l'Arithmetique n'est composée, que de quatre ou cinq operations, qui sont l'Addition, la Soustraction, la Multiplication, la Division, & l'Extraction des racines, qu'on peut prendre pour vne espece de Division : Ainsi n'at'on autre chose a faire en Geometrie touchant les lignes qu'on cherche, pour les preparer a estre connuës, que leur en adiouster d'autres, ou en oster, Oubien en ayant vne, que ie nommeray l'vnité pour la rapporter d'autant mieux aux nombres, & qui peut ordinairement estre prise a discretion, puis en ayant encore deux autres, en trouuer vne quatriesme, qui soit a l'vne de ces deux, comme l'autre est a l'vnité, ce qui est le mesme que la Multiplication, oubien en trouuer vne quatriesme, qui soit a l'vne de ces deux, comme l'vnité

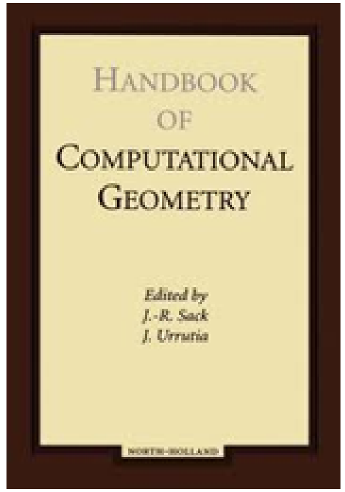
Comment le calcul d'Arithmetique se rapporte aux operations de Geometrie.

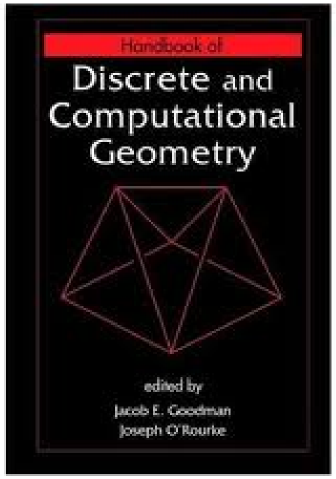
A brief history of computational geometry

Some early milestones

- 1975: Michael Shamos' PhD thesis *Problems in Computational Geometry*.
- 1975-1985: Increasing interest. Most basic algorithms date from this period.
- 1983: First European Workshop on Computational Geometry
- 1985: First Annual Symposium on Computational Geometry
  Also: first textbook (today, more than 5)
- 1996: CGAL: first serious implementation of a robust geometric algorithms library
- 1997: First handbook on the topic (second in 2000)







A brief history of computational geometry

Today

- Recognized discipline within *Theoretical Computer Science* and *Discrete Mathematics*
- "Large" active community. Many research groups in the USA, Canada, Europe, México, Australia,...
  In Spain: Barcelona, Sevilla, Santander, Alcalá, Zaragoza, Girona, ...
- 5 specialized journals devoted to it
- 3 annual specialized conferences
- An important presence in algorithms and discrete math conferences

Goals for Part II (algorithmic geometry)

- Know some of the main problems studied in computational geometry and its solutions, as well as its applications
- Understand the power of combining geometric tools with the suitable data structures and algorithmic paradigms
- See in action several algorithmic paradigms and data structures useful in geometric problems
- Apply geometric results to solve new problems

Syllabus

(Part II)

1. Background tools
2. A basic tool (relative position)
3. Computing segment intersections
4. Convex hulls in 2D, 3D and higher dimensions
5. Convex hulls. Duality. Intersection of halfplanes. Linear programming
6. Triangulating polygons
7. Voronoi diagrams
8. Delaunay triangulations

This course

Methodology

- Theory lectures
- Problem sessions

This course

Methodology

- Theory lectures → by the instructor
- Problem sessions → by the students
  - 2 problem sessions planned
  - each student will deliver and present two problem (in total)

This course

Methodology

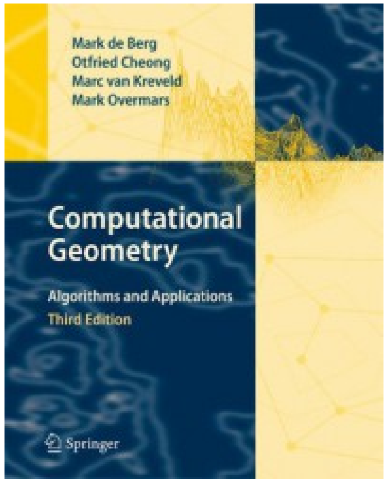
- Theory lectures → by the instructor
- Problem sessions → by the students
  - 2 problem sessions planned
  - each student will deliver and present two problem (in total)

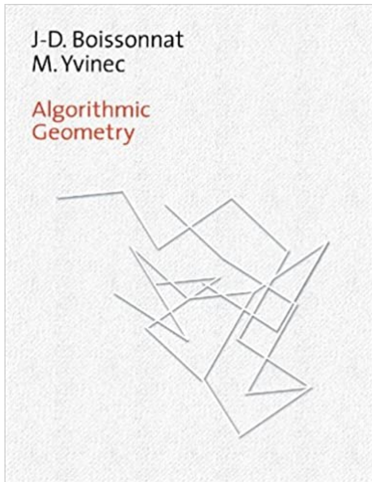
Evaluation

- Course grade: 0.5 (grade of Part I) + 0.5 (grade of Part II)
- Grade of each Part II: 0.5 (first problem) + 0.5 (second problem)
- *Optional* partial exam for Part II

Main bibliography

- M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, **Computational Geometry: Algorithms and Applications (3rd ed.)**, Springer, 2008.
- J.-D. Boissonat. M. Yvinec, **Algorithmic Geometry**, Cambridge University Press, 1998.





Part I

Contact and further information

- Course **web page** with slides and material (Part II): https://dccg.upc.edu/courses-dag
- **Atenea** for announcements and additional material

This course

Contact and further information

- Course **web page** with slides and material (Part II): https://dccg.upc.edu/courses-dag
- **Atenea** for announcements and additional material

Office hours

- By appointment. You can find me at the Omega building (Campus Nord), 4th floor
- To visit me, or for any other matter, send me an email:
    - **Rodrigo Silveira:** rodrigo.silveira@upc.edu

(*) But not at *any* time of *any* day, so always contact by email first