PRESENTATION OF THE COURSE

Vera Sacristán
Rodrigo Silveira

Computational Geometry
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

PRESENTATION OF THE COURSE

The kernel of Computational Geometry is the design and analysis of efficient algorithms to solve geometric problems.

PRESENTATION OF THE COURSE

The kernel of Computational Geometry is the design and analysis of efficient algorithms to solve geometric problems.

PRESENTATION OF THE COURSE

The kernel of Computational Geometry is the design and analysis of efficient algorithms to solve geometric problems.

What kind of geometric problems?

PRESENTATION OF THE COURSE

The kernel of Computational Geometry is the design and analysis of efficient algorithms to solve geometric problems.

What kind of geometric problems?

Geometric problems that underlie a wide variety of applications

PRESENTATION OF THE COURSE

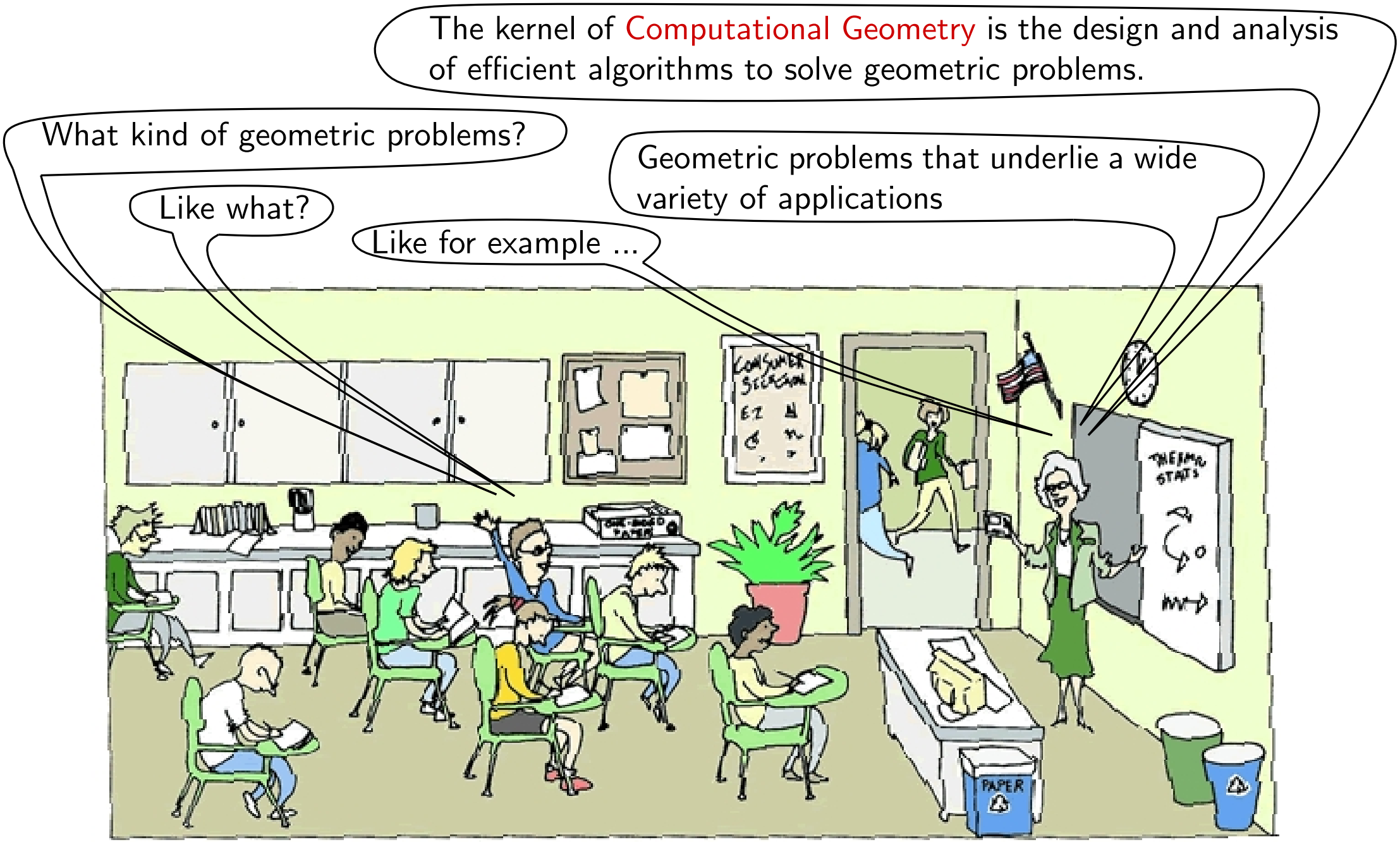
The kernel of Computational Geometry is the design and analysis of efficient algorithms to solve geometric problems.

What kind of geometric problems?

Like what?

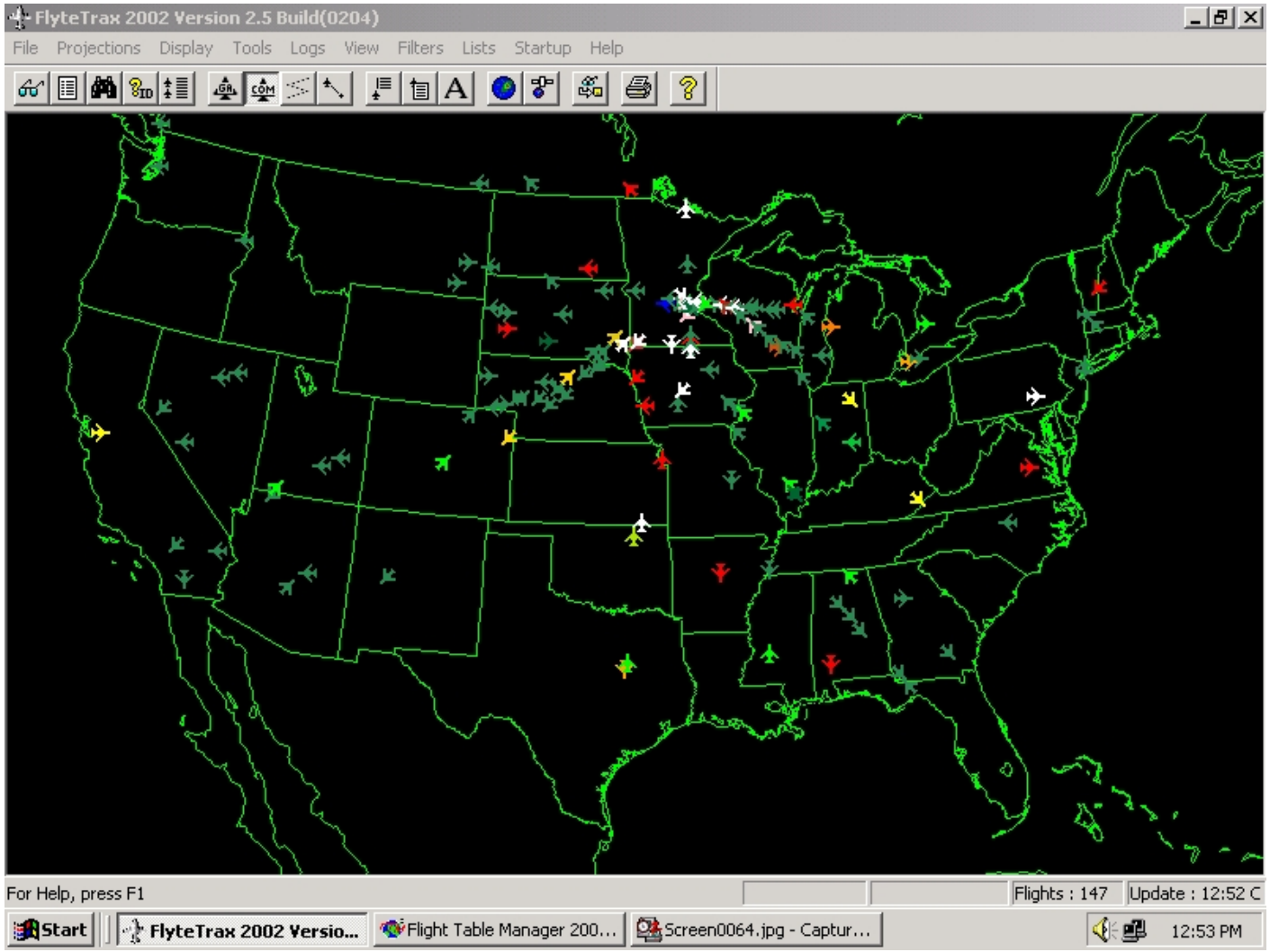
Geometric problems that underlie a wide variety of applications

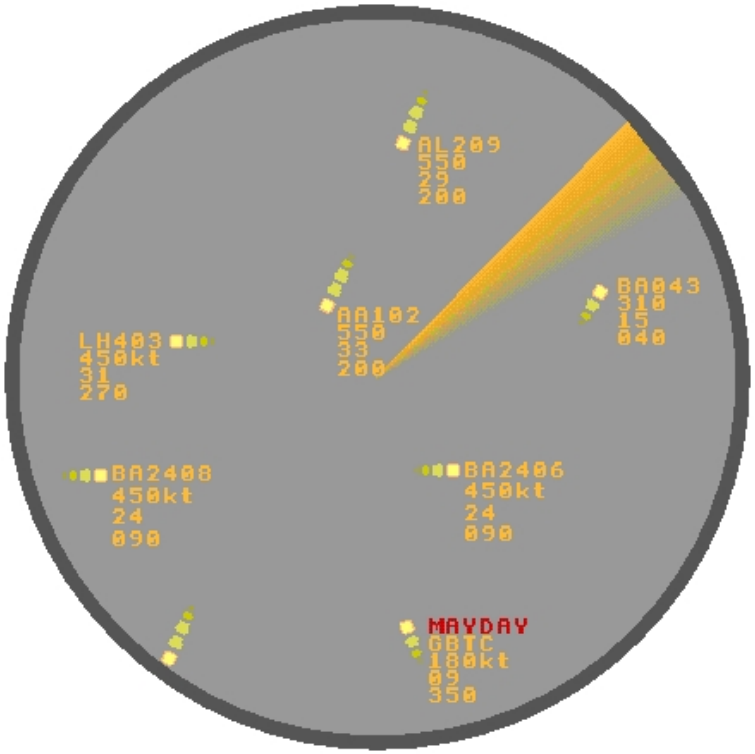
PRESENTATION OF THE COURSE



AIR TRAFFIC CONTROL

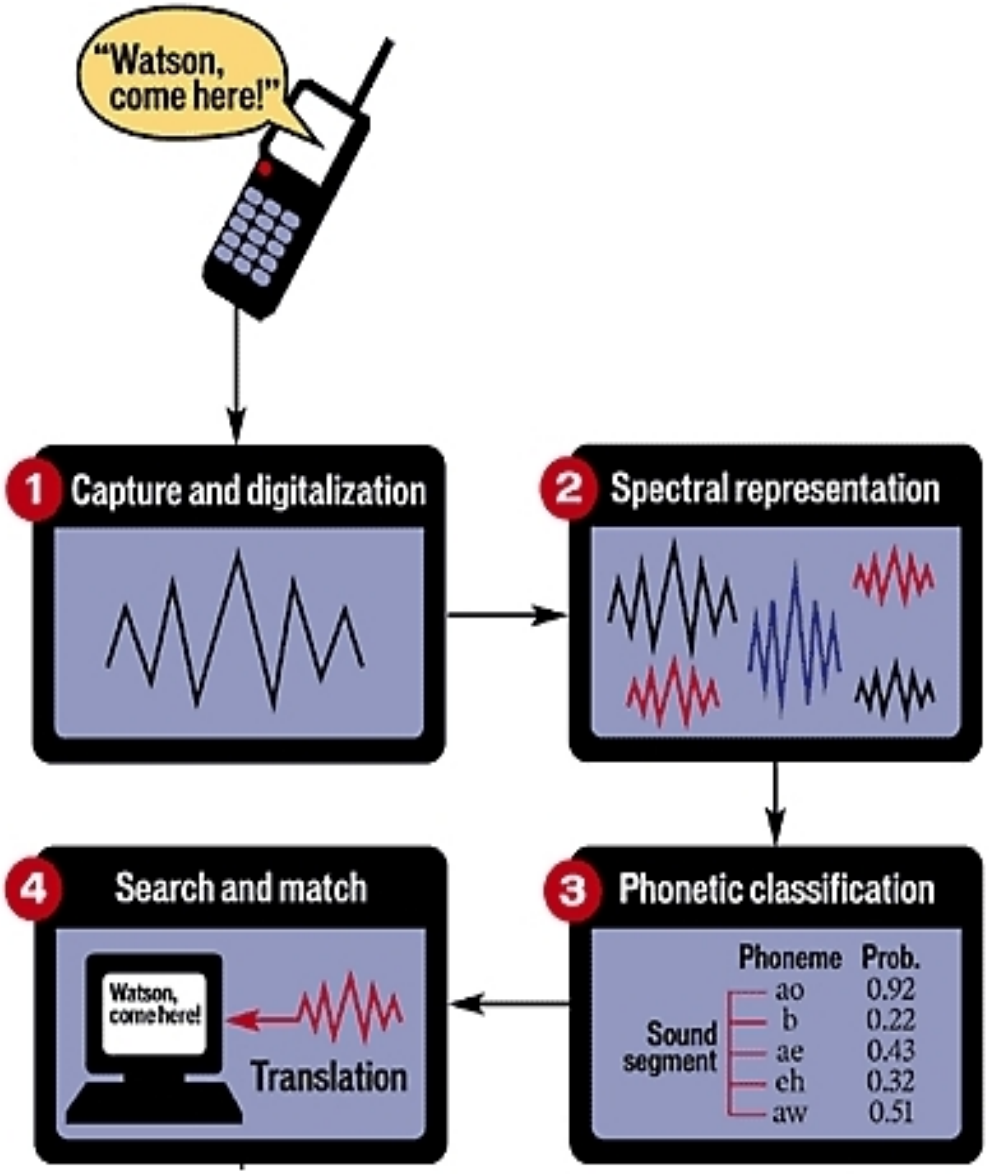
Detect the pair of airplanes that are in most imminent danger of collision among those who show up on the screen of an air controller.





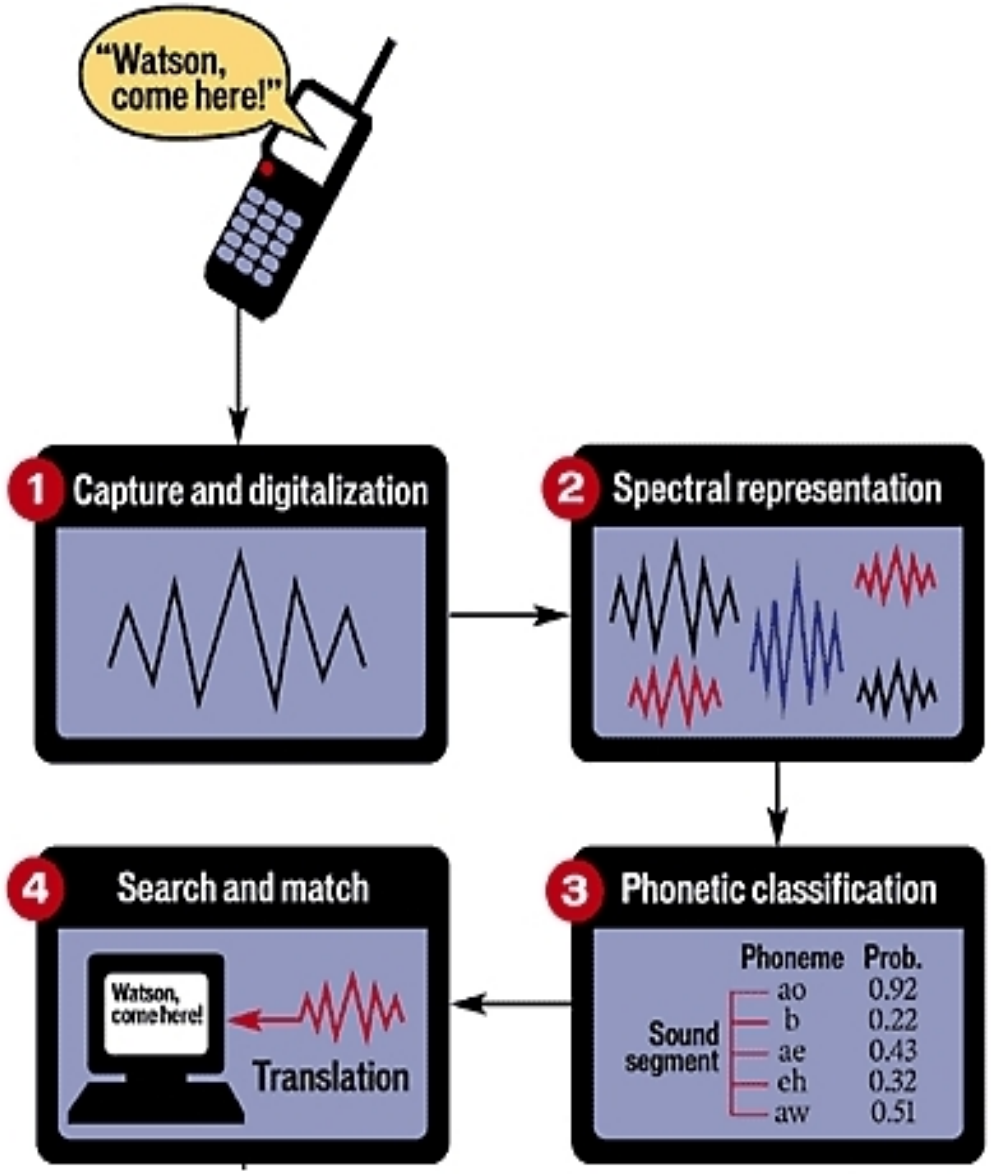
AUTOMATIC VOICE RECOGNITION

AUTOMATIC VOICE RECOGNITION



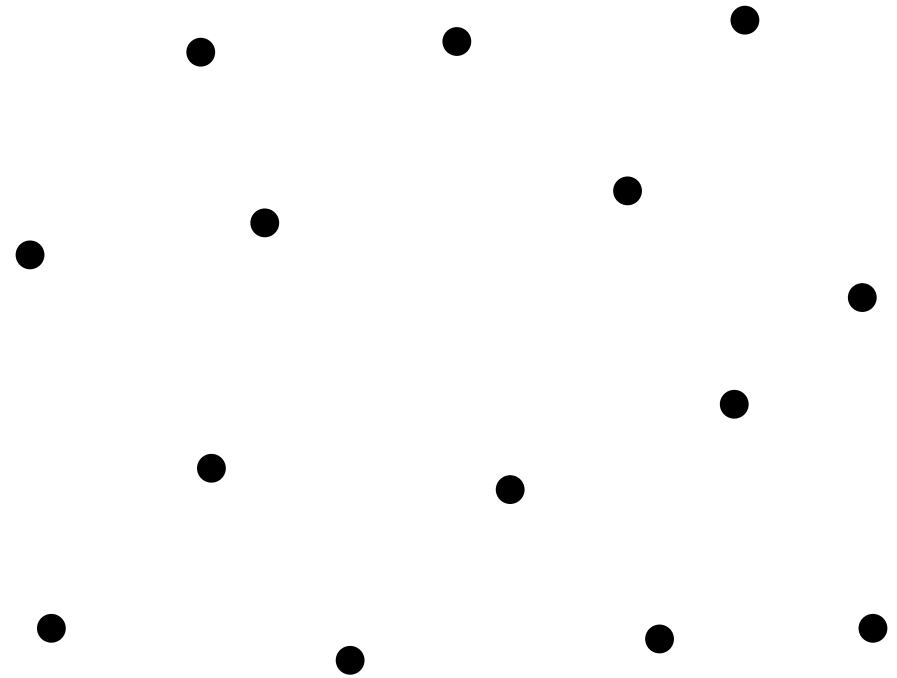
AUTOMATIC VOICE RECOGNITION

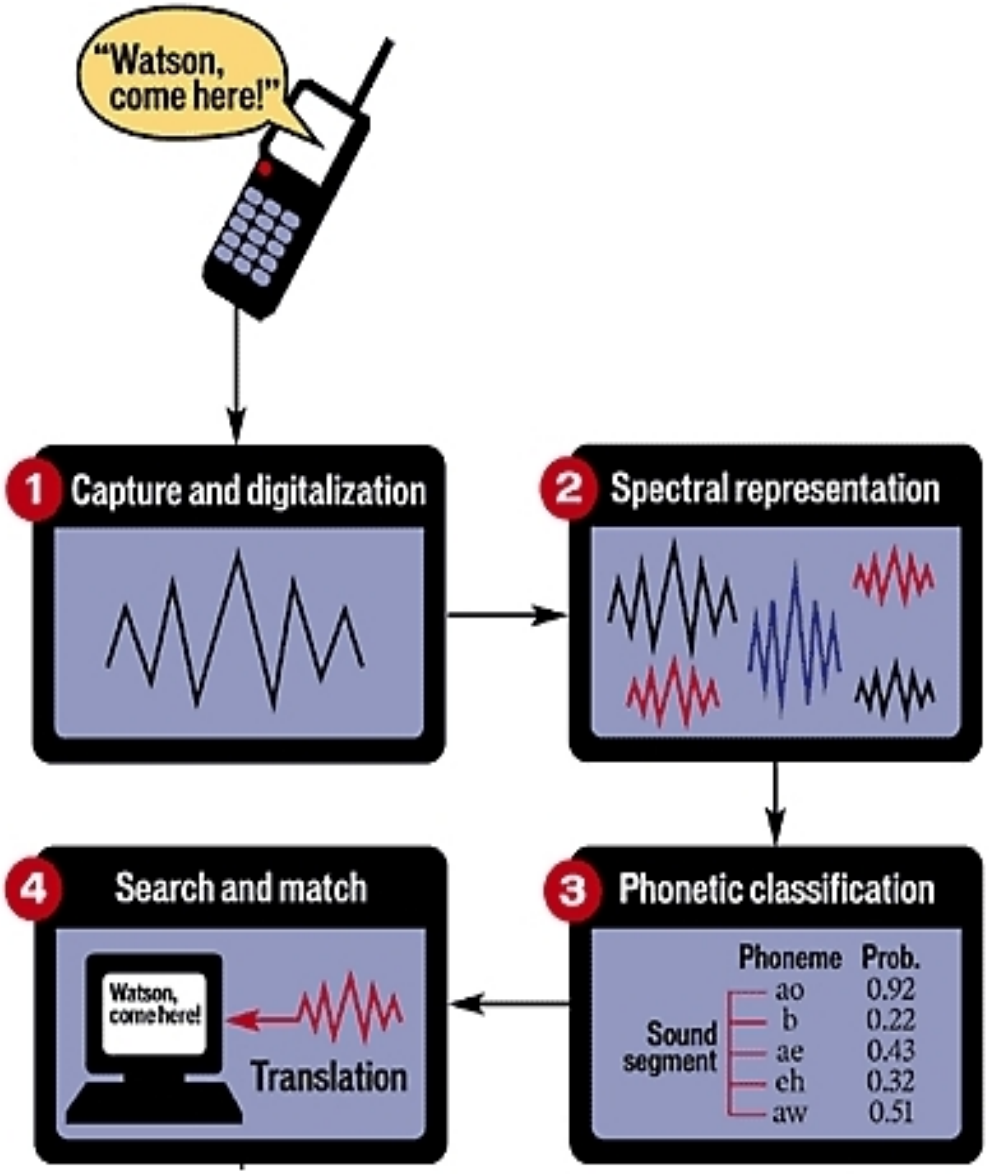
In general, recognizing a pattern consists in classifying an element (in the case of voice recognition, some sounds) detecting the most similar (the closest, in some sense) of some known patterns.



AUTOMATIC VOICE RECOGNITION

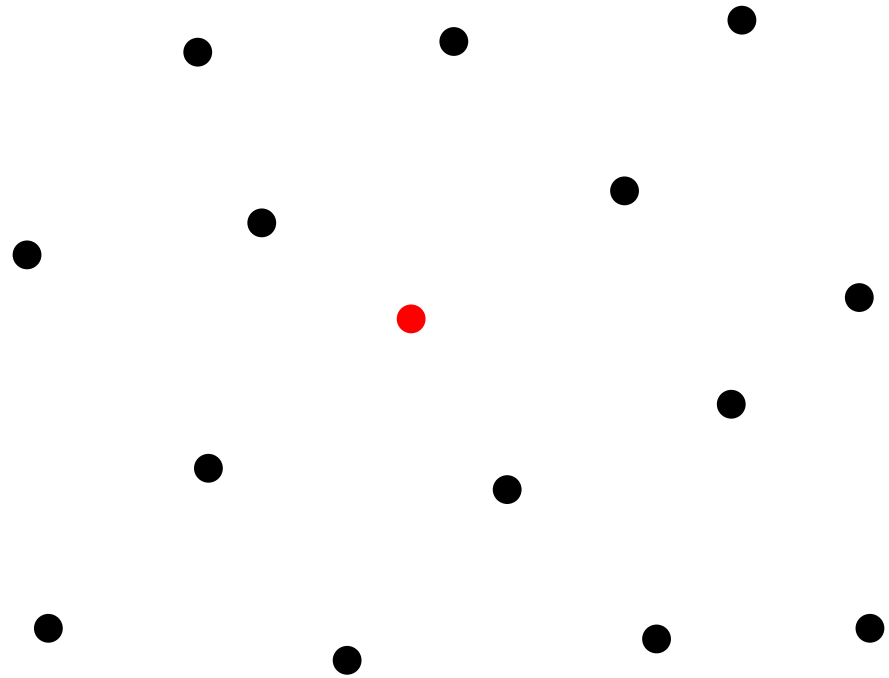
In general, recognizing a pattern consists in classifying an element (in the case of voice recognition, some sounds) detecting the most similar (the closest, in some sense) of some known patterns.

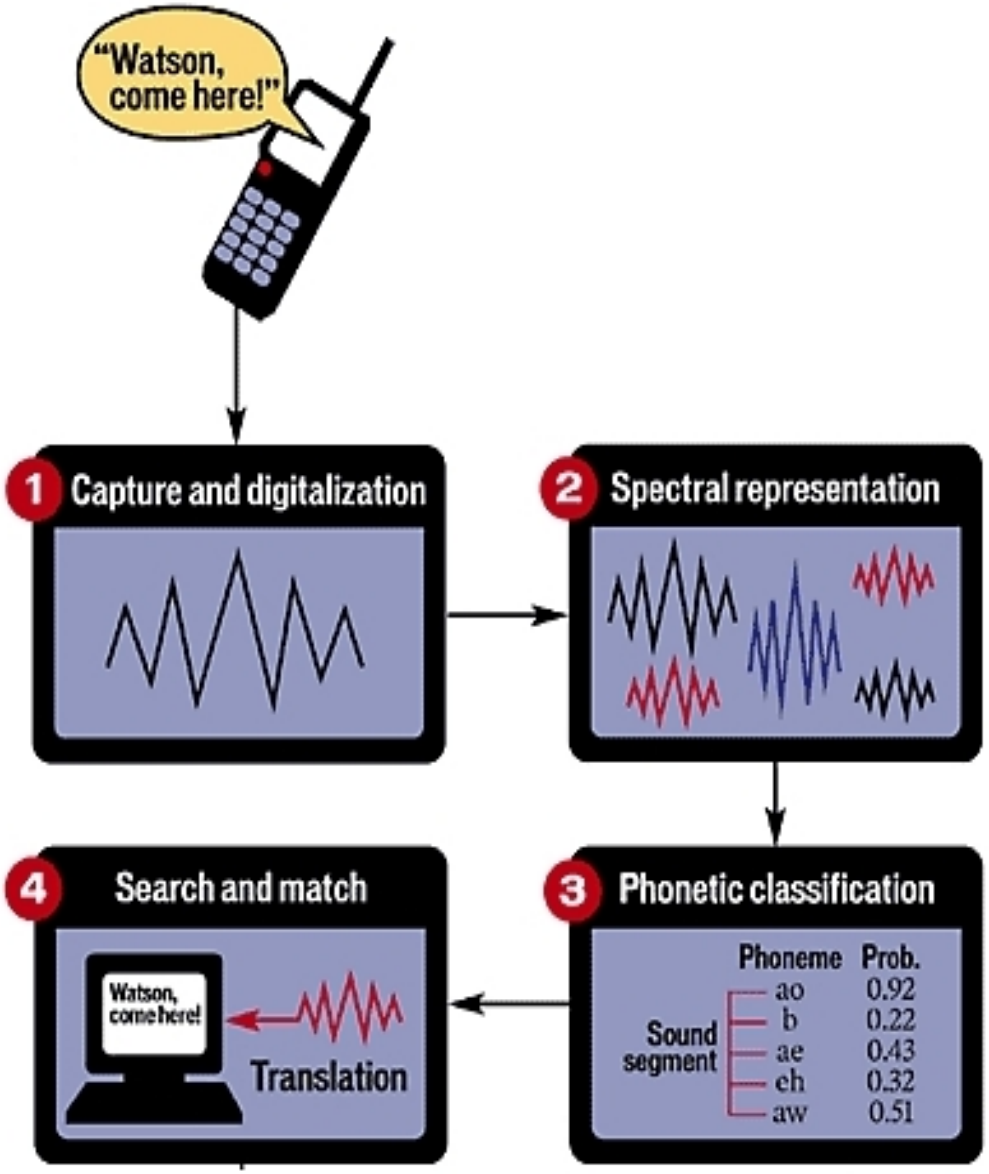




AUTOMATIC VOICE RECOGNITION

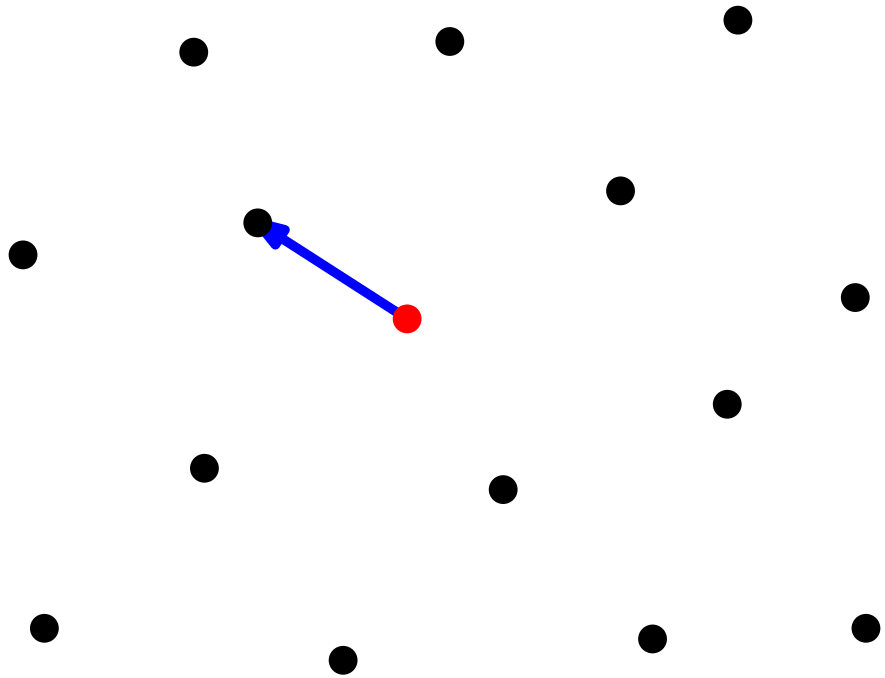
In general, recognizing a pattern consists in classifying an element (in the case of voice recognition, some sounds) detecting the most similar (the closest, in some sense) of some known patterns.

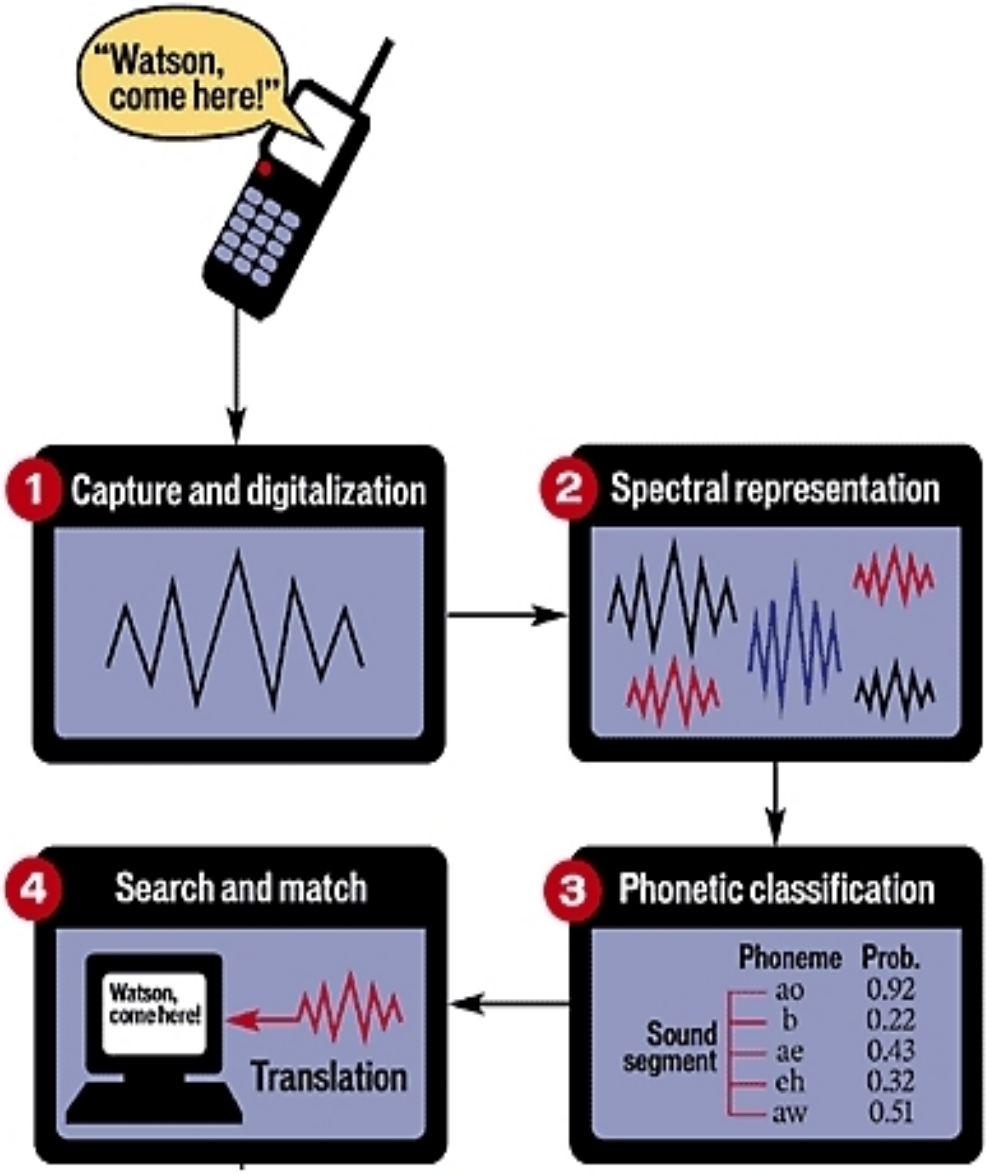




AUTOMATIC VOICE RECOGNITION

In general, recognizing a pattern consists in classifying an element (in the case of voice recognition, some sounds) detecting the most similar (the closest, in some sense) of some known patterns.





PRESENTATION OF THE COURSE

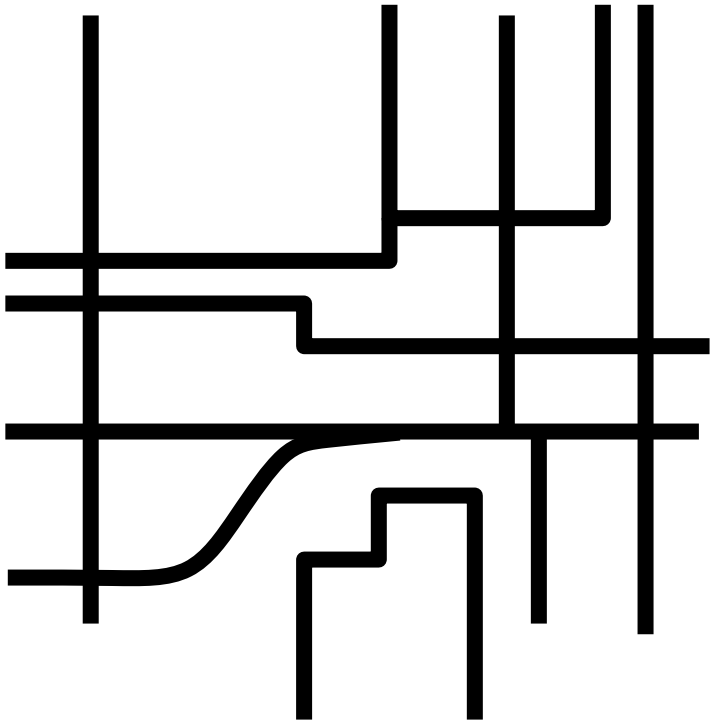
ROAD NAVIGATION (with GPS)

ROAD NAVIGATION (with GPS)

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

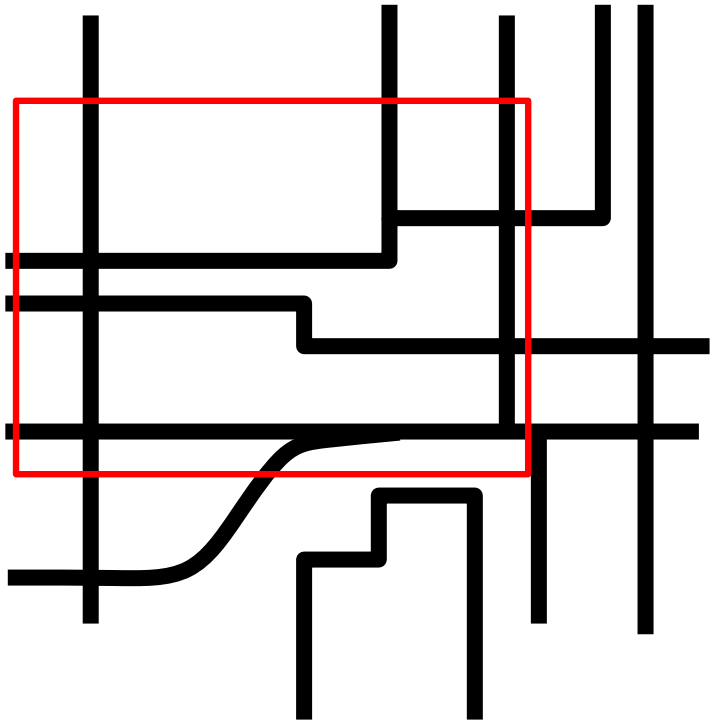
ROAD NAVIGATION (with GPS)

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



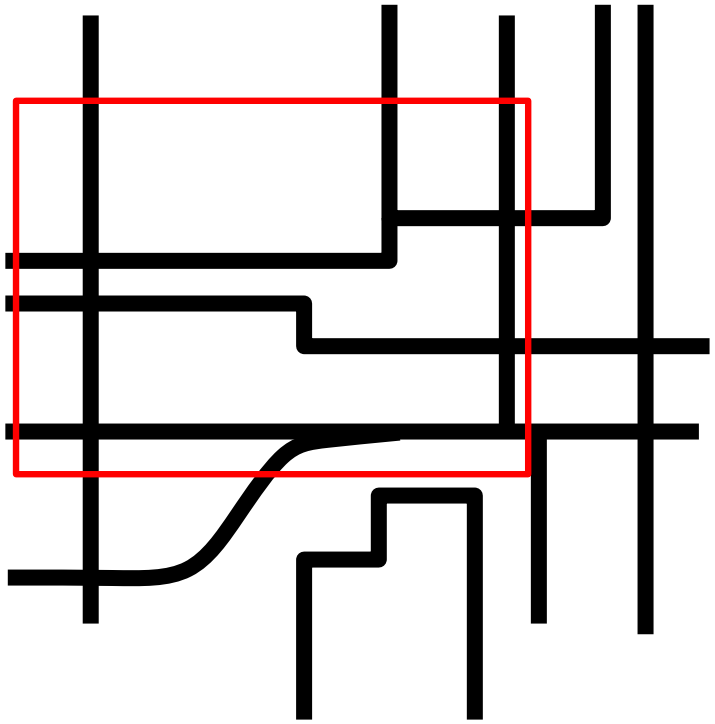
ROAD NAVIGATION (with GPS)

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



ROAD NAVIGATION (with GPS)

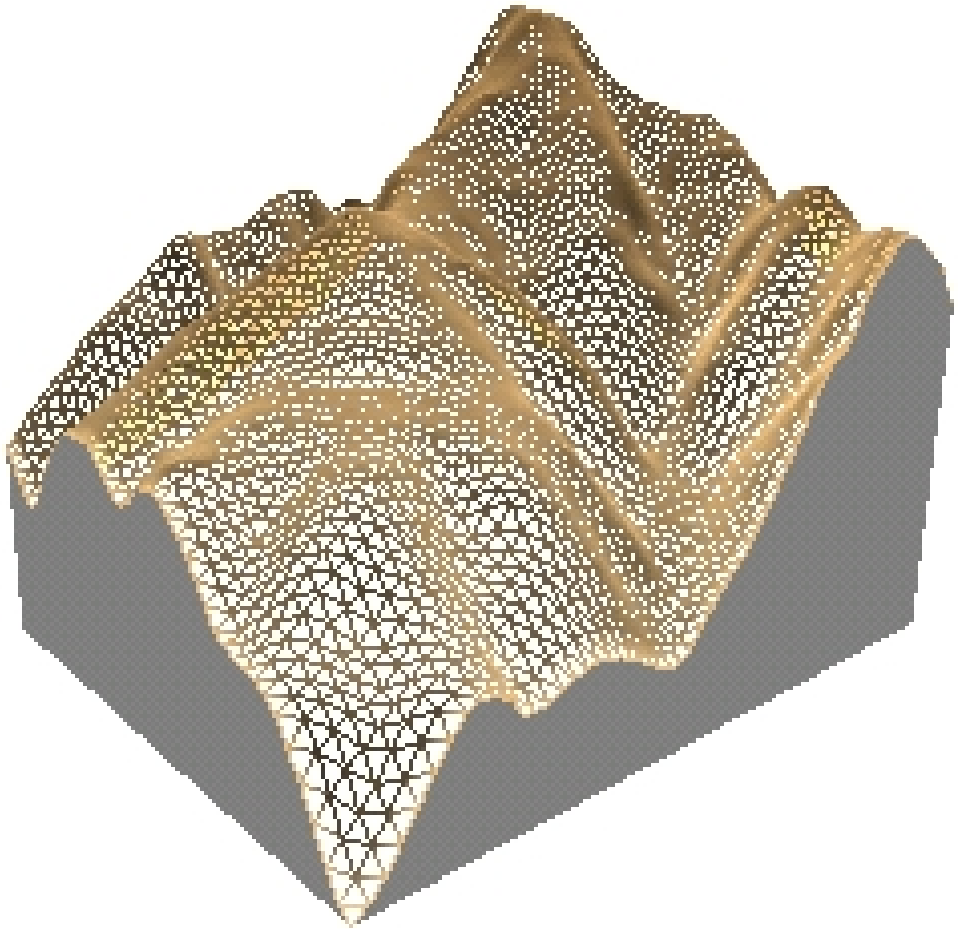
How does your GPS navigator know which objects (streets, places of interest, etc.) to show?

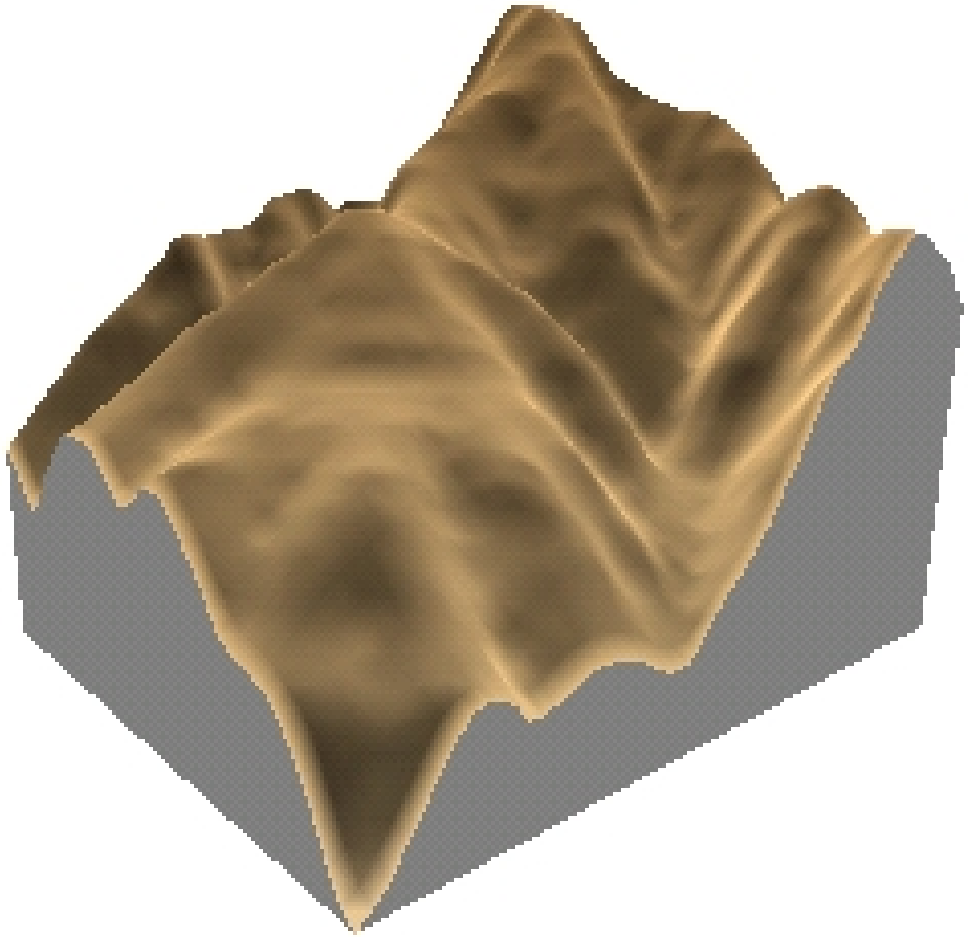


This geometric problem is called *windowing*

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.

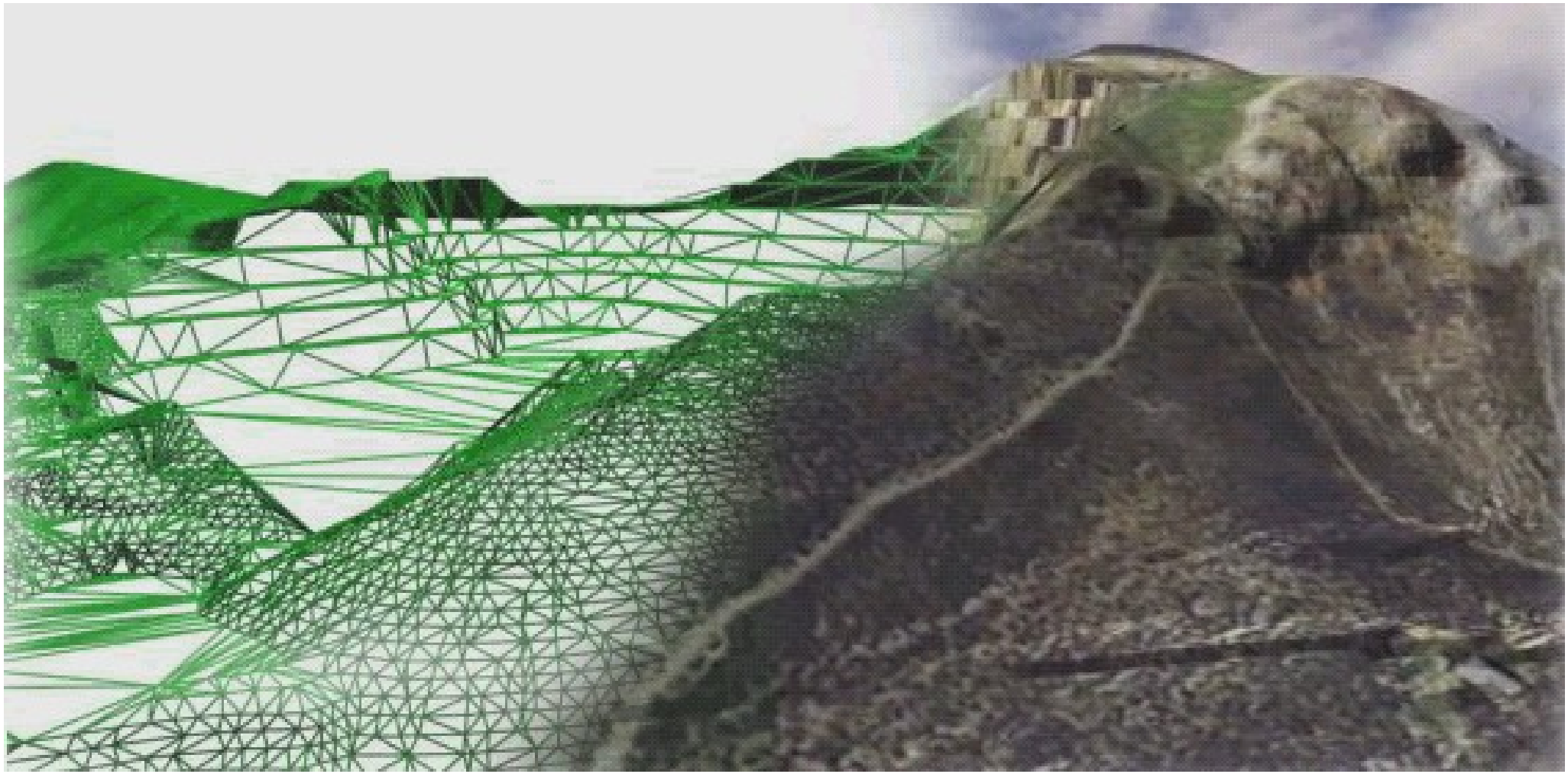




PRESENTATION OF THE COURSE

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.



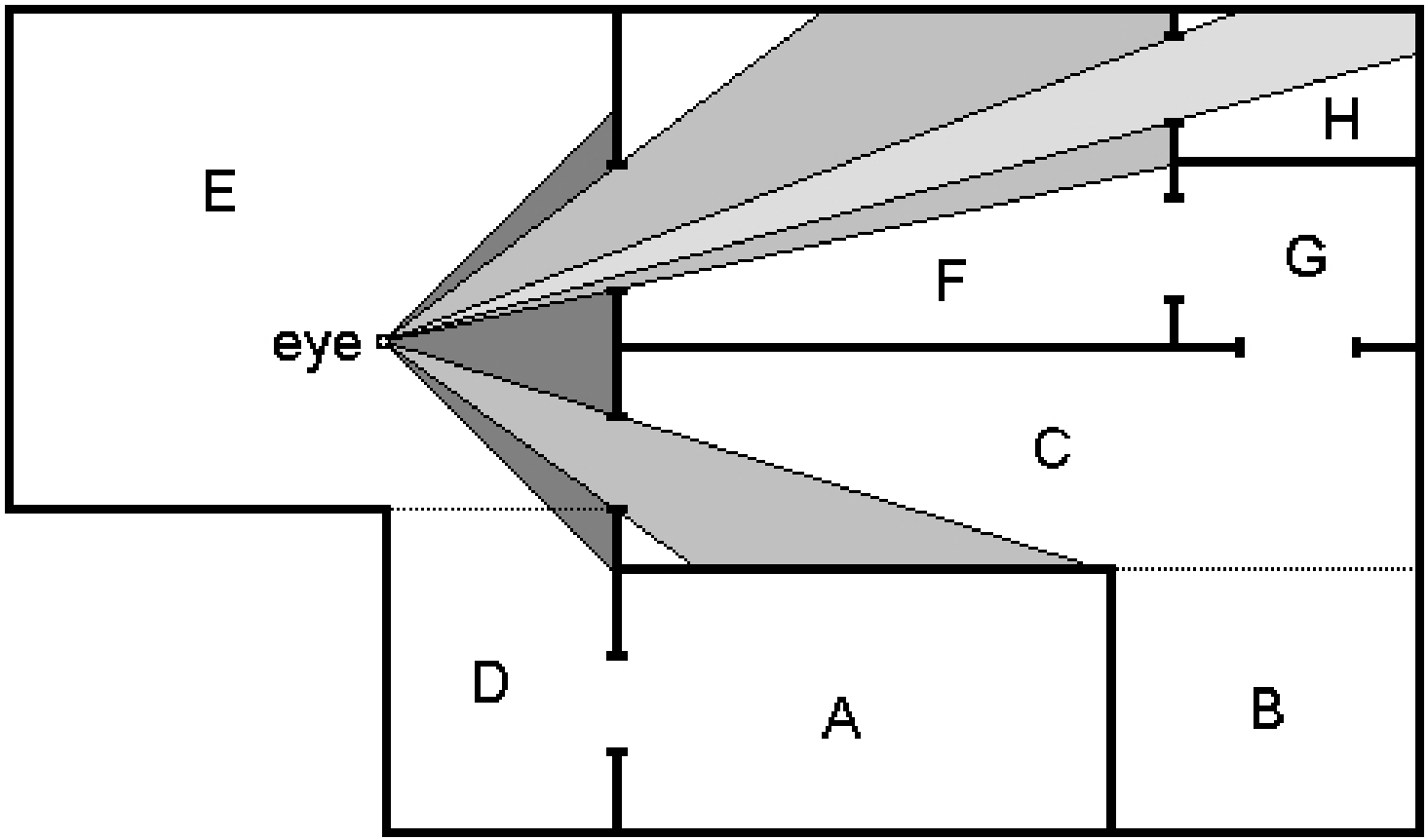
PRESENTATION OF THE COURSE

VIRTUAL NAVIGATION: 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.

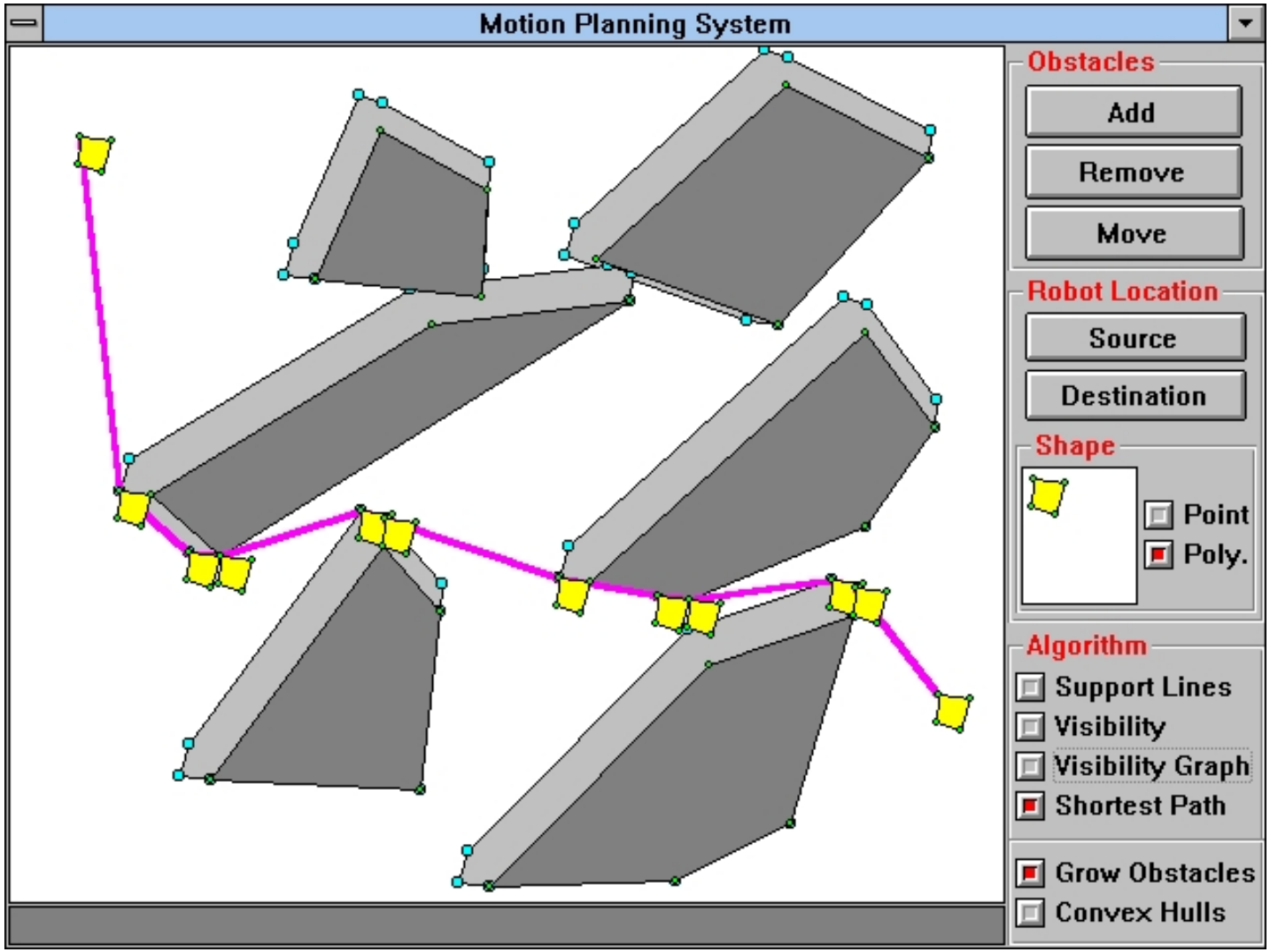
VIRTUAL NAVIGATION: HIDING NON VISIBLE PORTIONS OF A SCENE

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



ROBOTICS: MOTION PLANNING

Find the shortest path between two points, avoiding obstacles

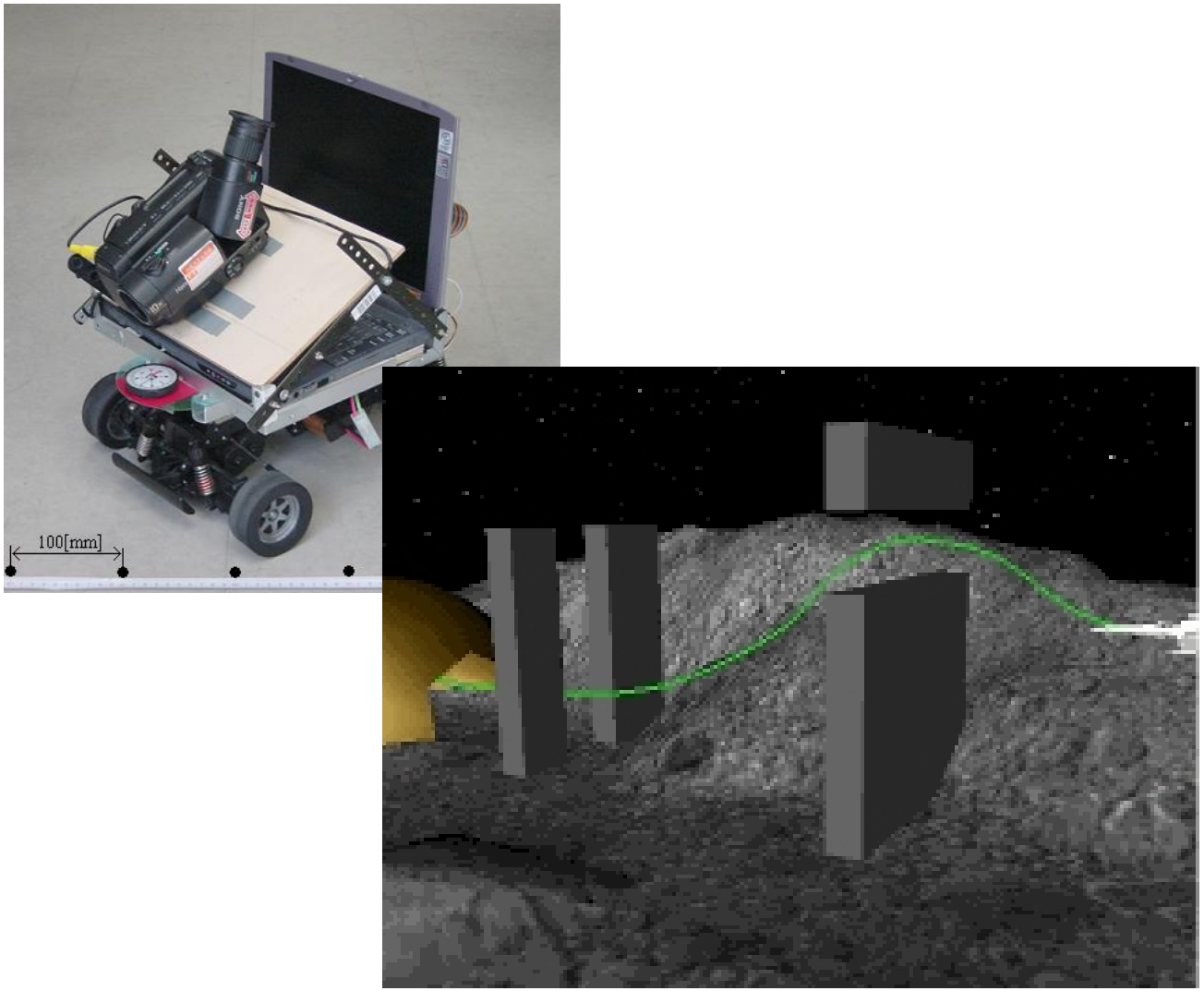


ROBOTICS: MOTION PLANNING

Find the shortest path between two points, avoiding obstacles

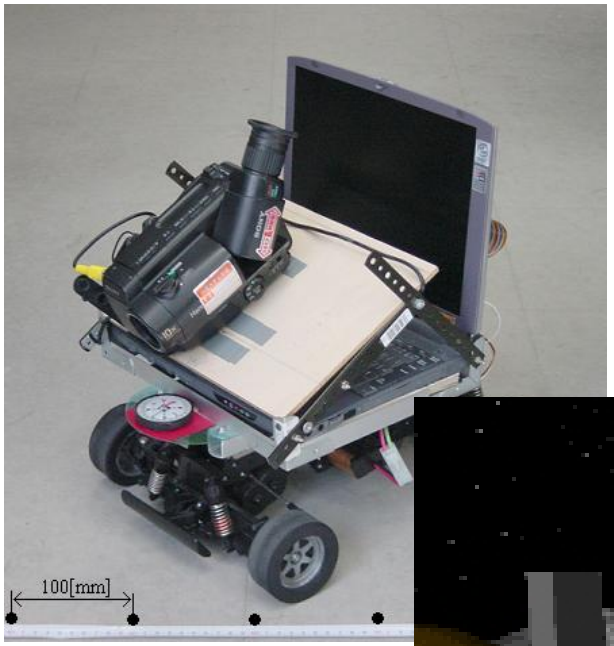
ROBOTICS: MOTION PLANNING

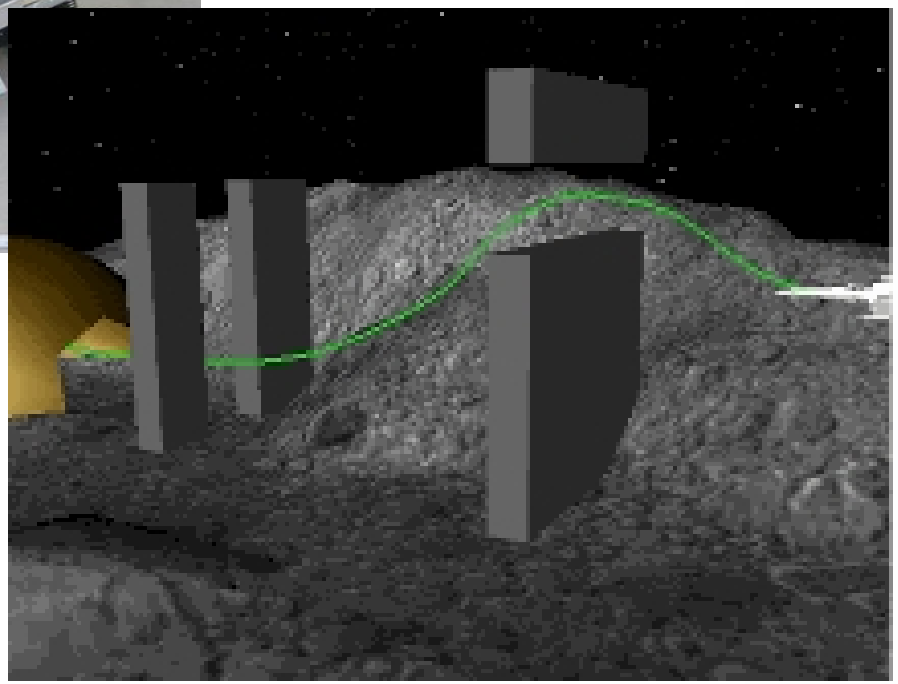
Find the shortest path between two points, avoiding obstacles

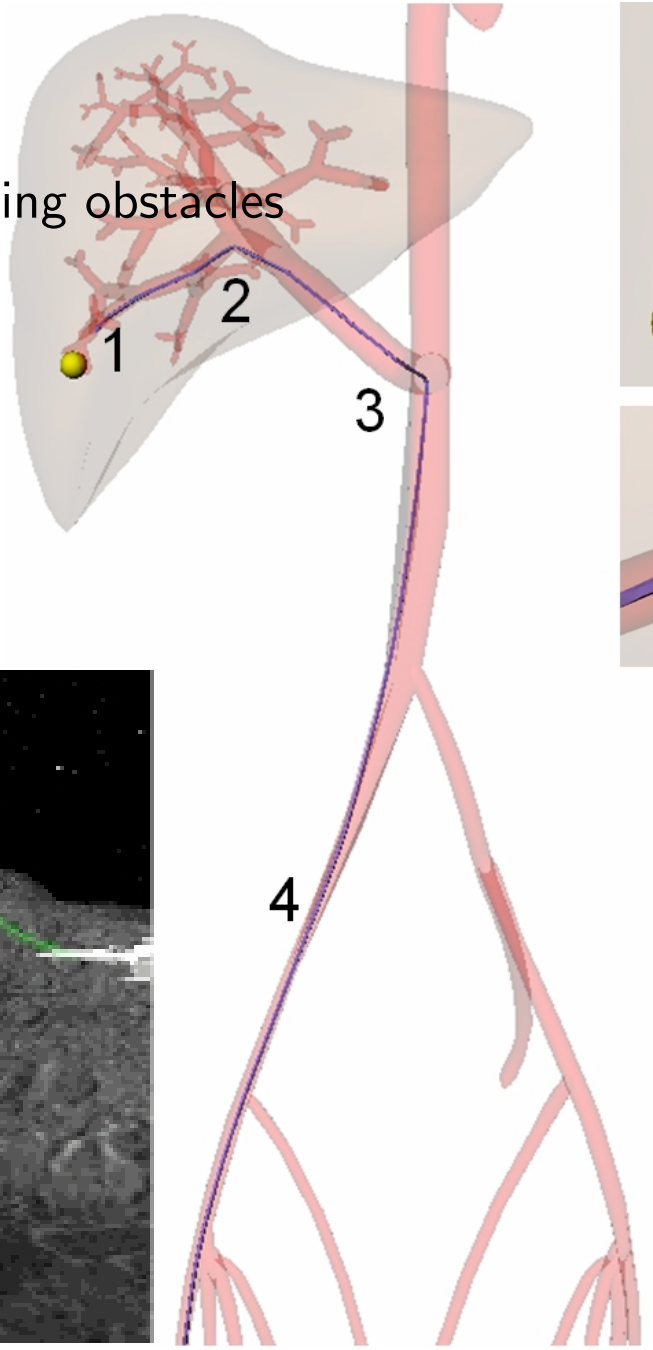


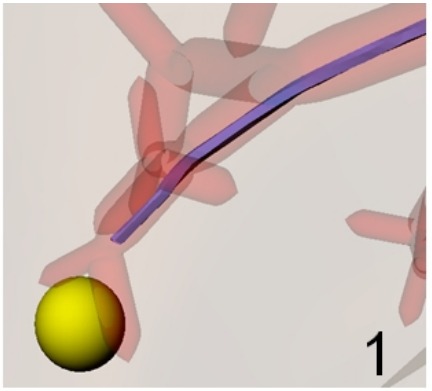
ROBOTICS: MOTION PLANNING

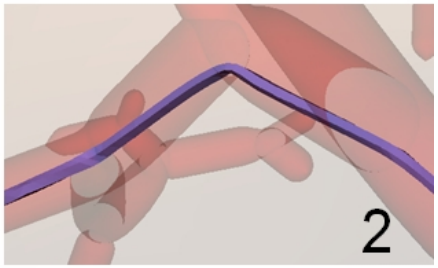
Find the shortest path between two points, avoiding obstacles

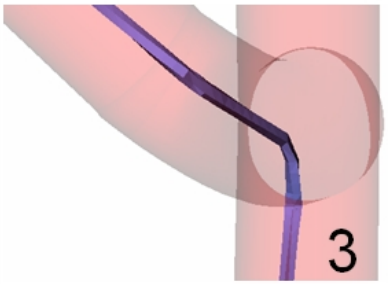


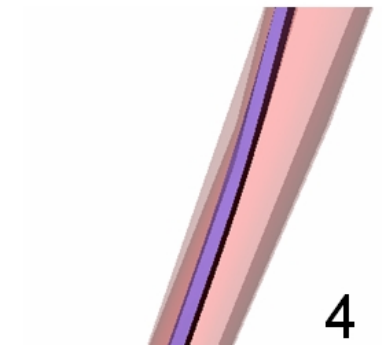






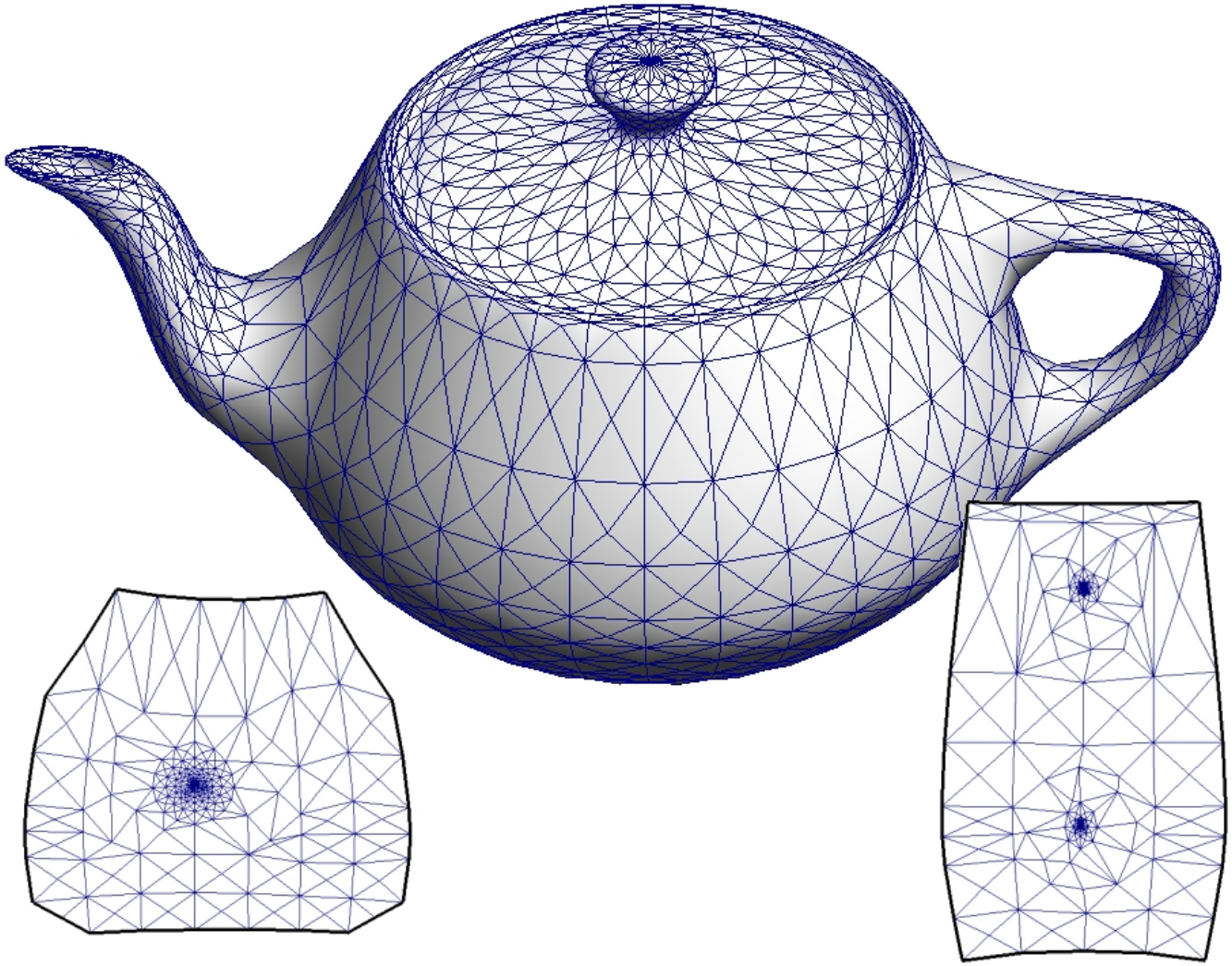






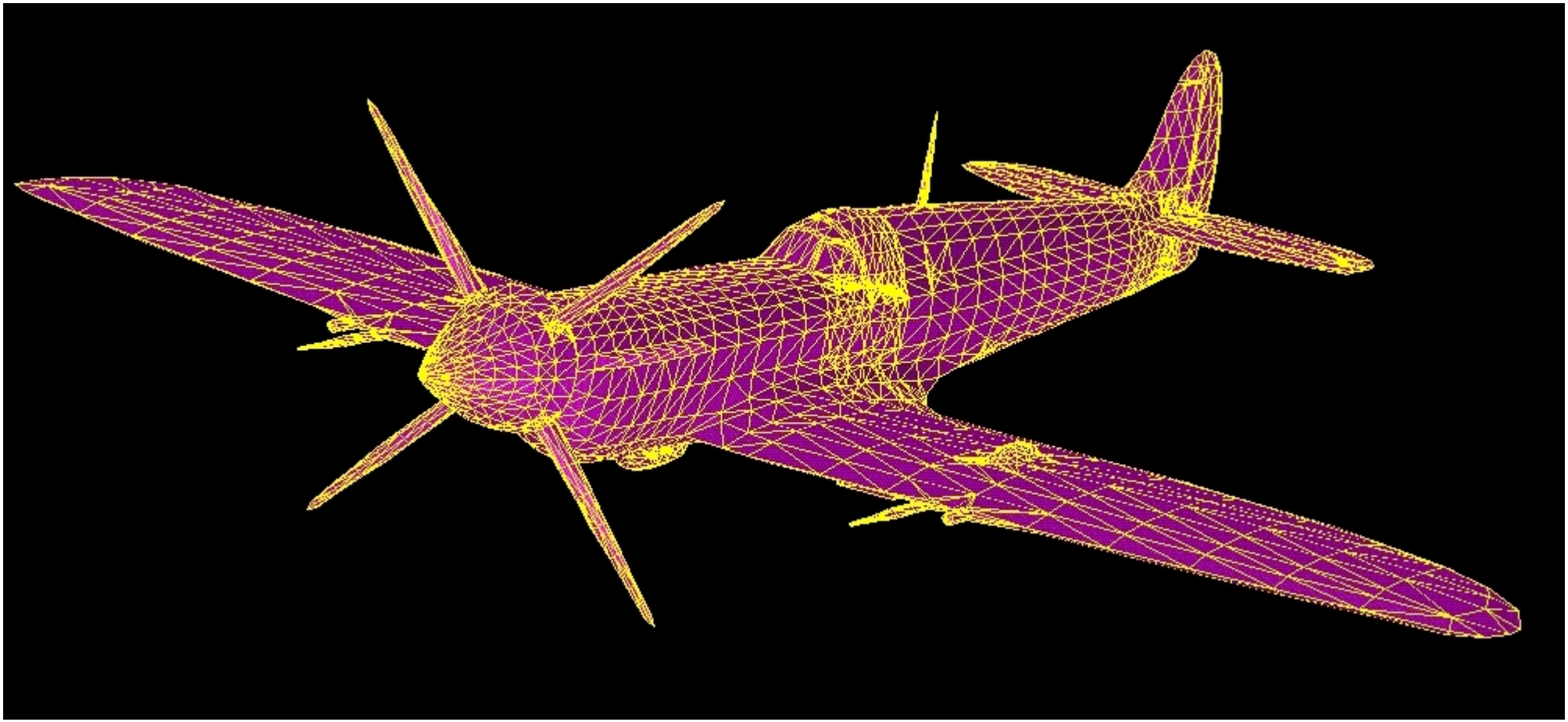
CAD and CAM: MODELLING OBJECTS

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



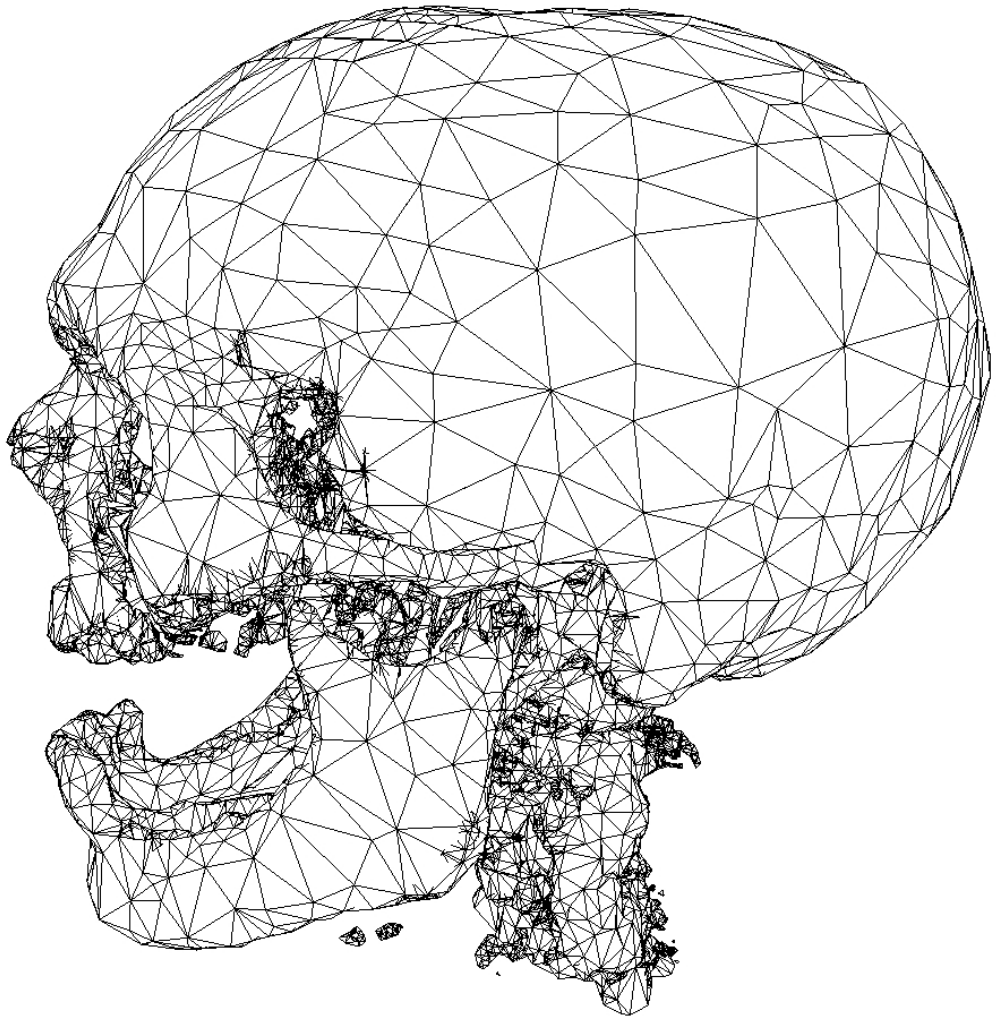
CAD and CAM: MODELLING OBJECTS

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



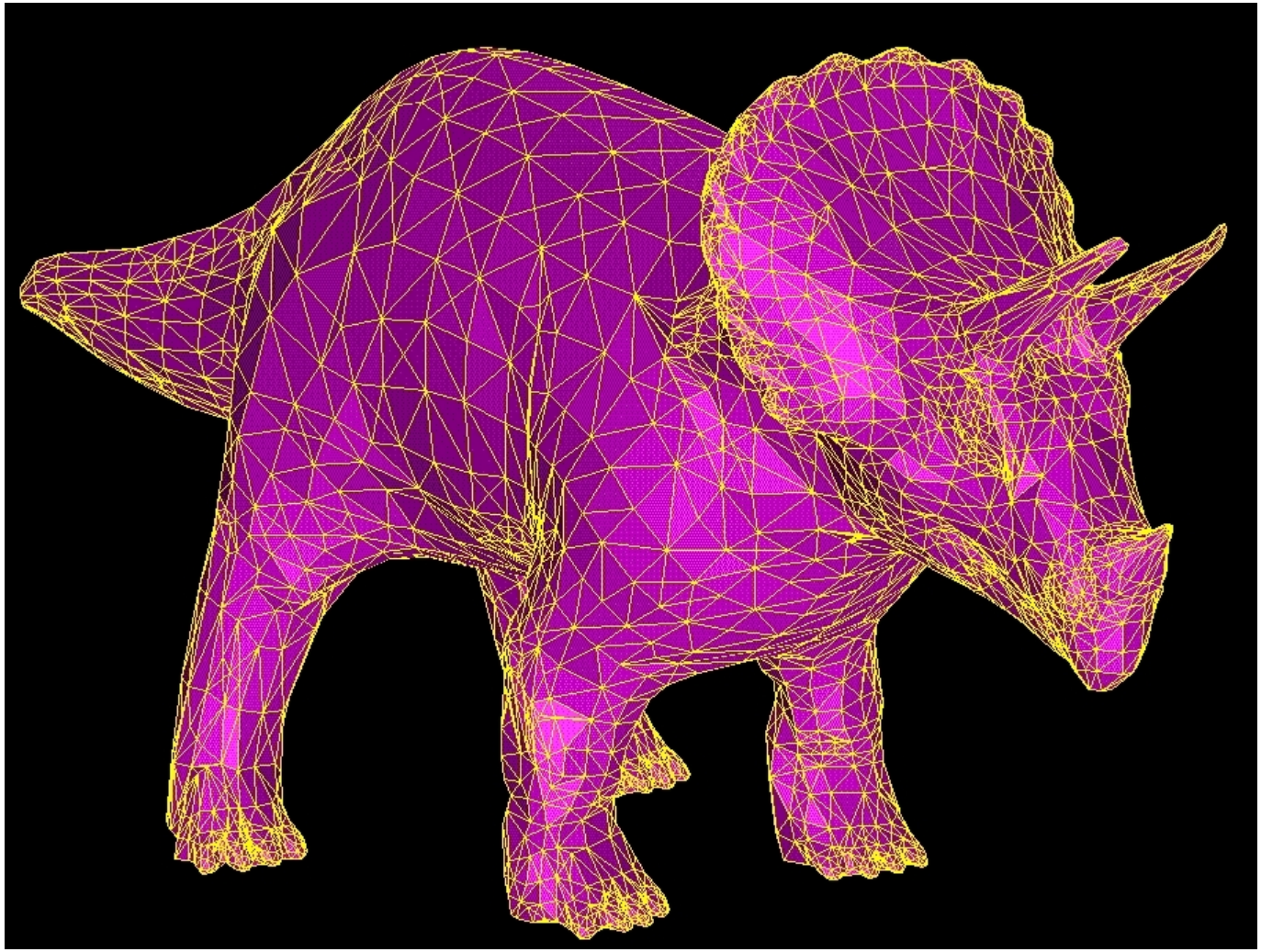
CAD and CAM: MODELLING OBJECTS

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



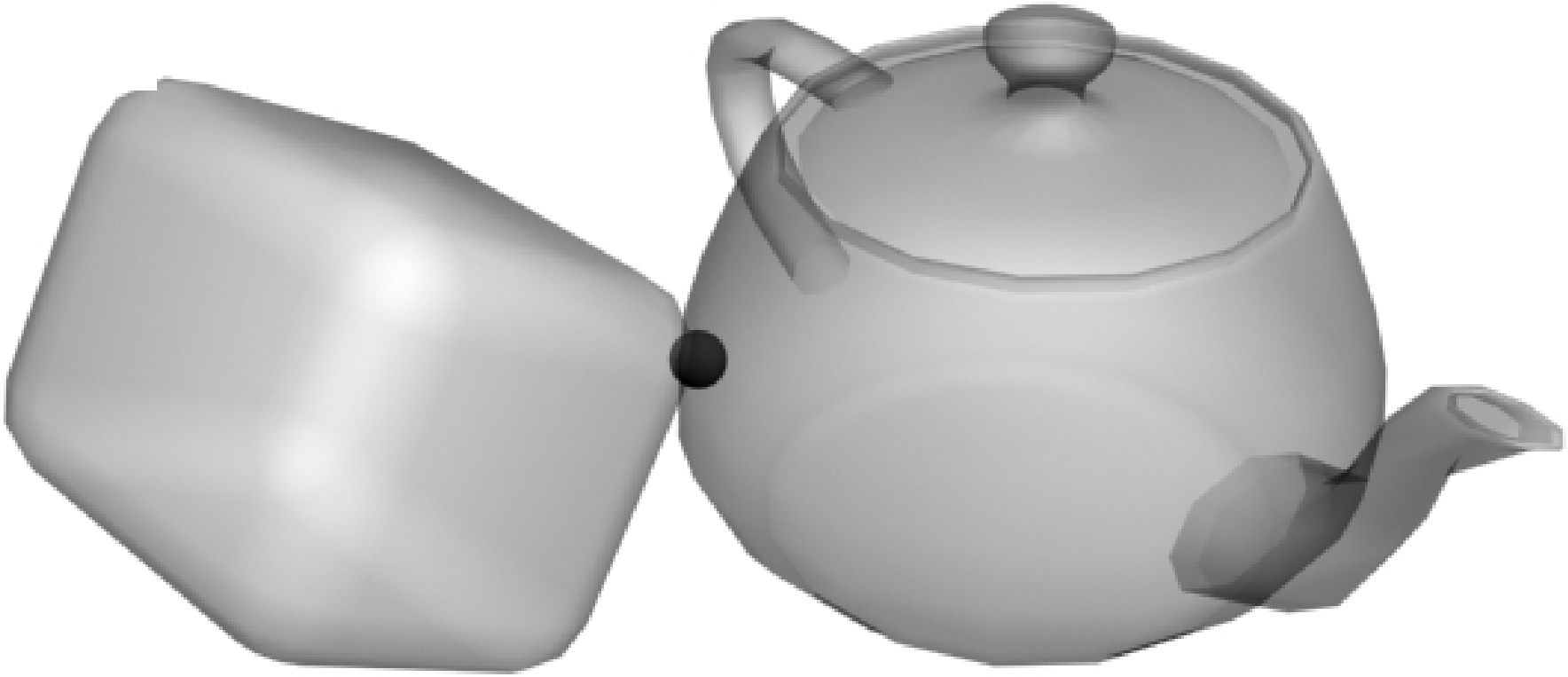
CAD and CAM: MODELLING OBJECTS

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



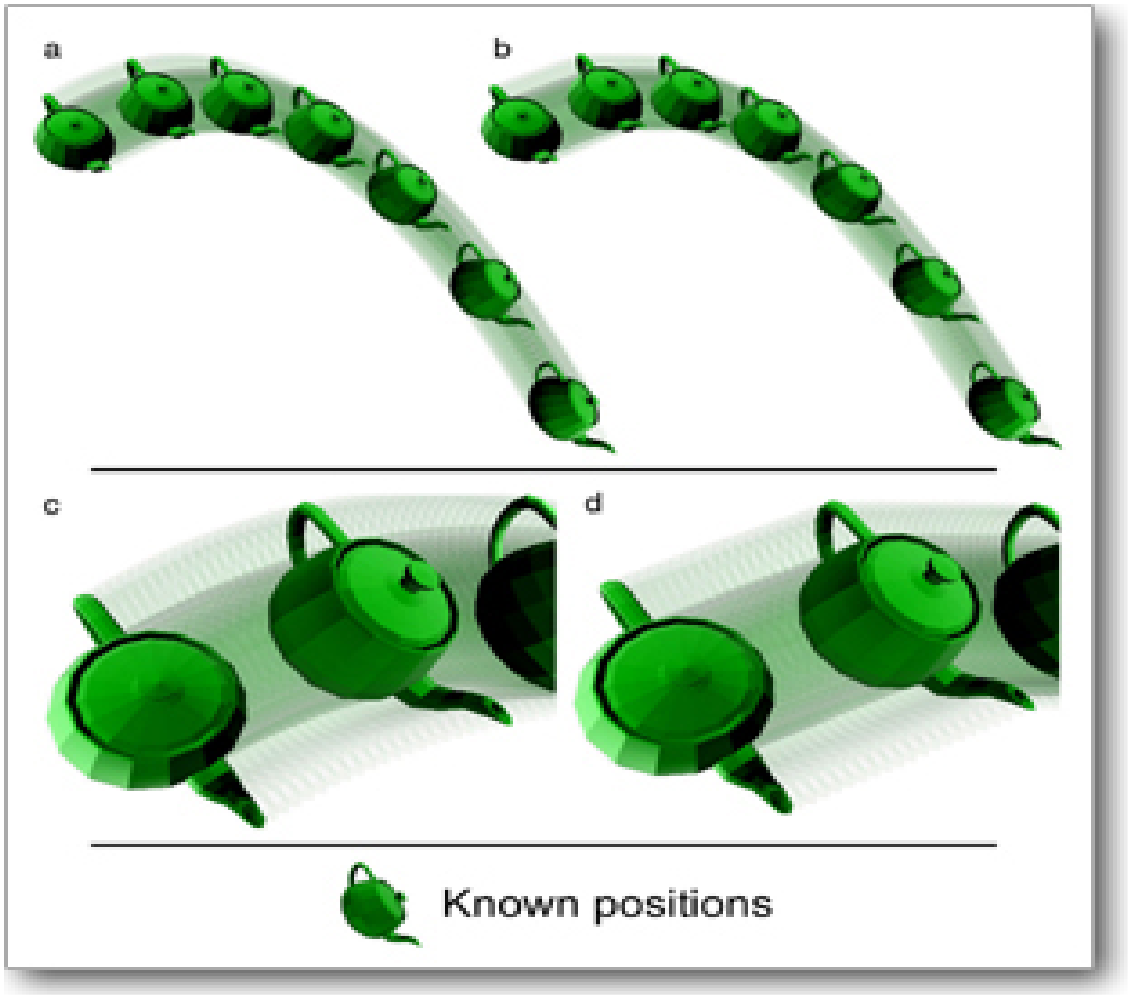
COLLISION DETECTION

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.

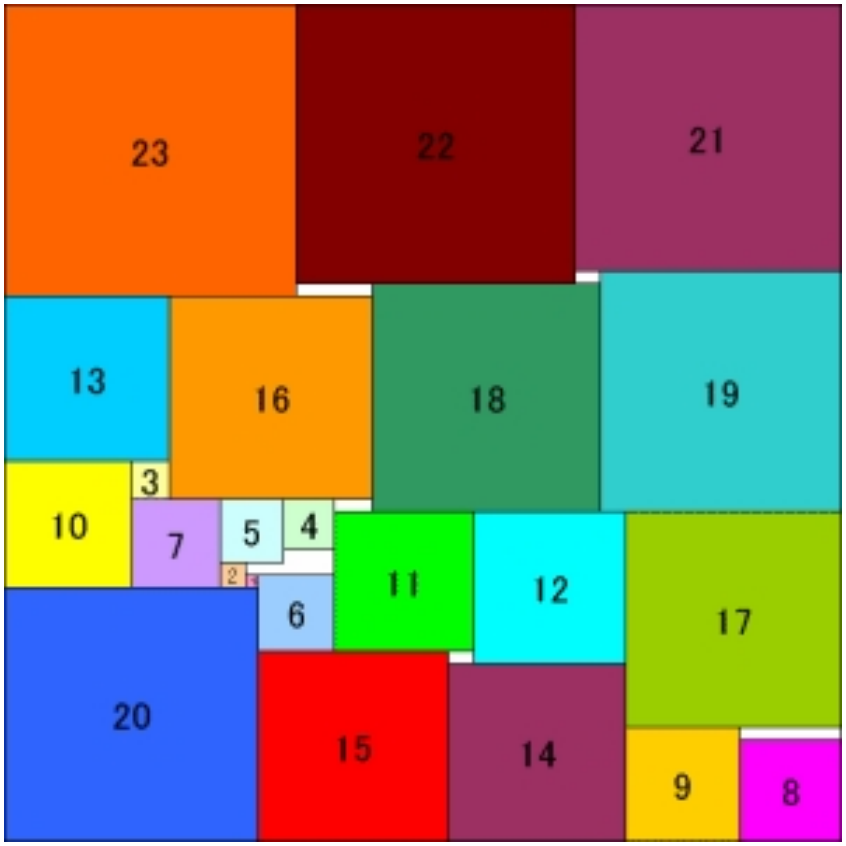


COLLISION DETECTION

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

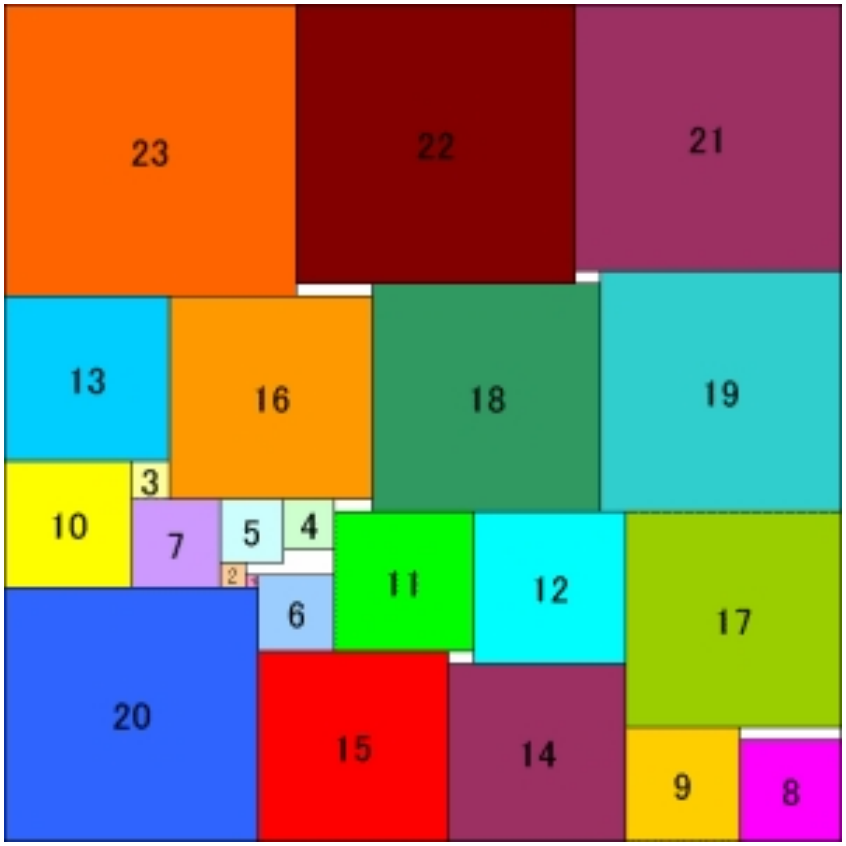
PACKING

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.



PACKING

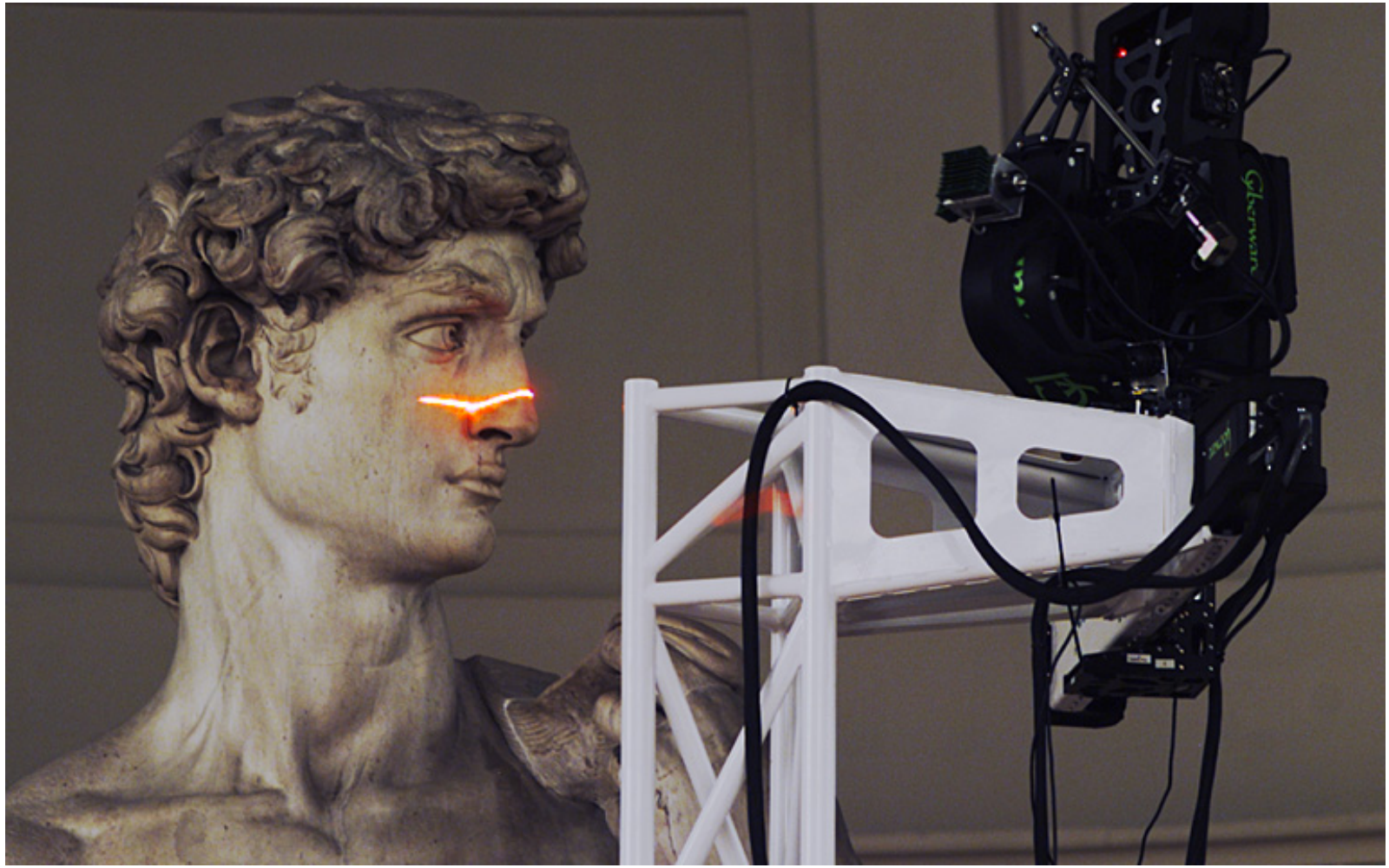
Given a set of objects (luggage), decide whether or not it is possible to pack them in a given space (trunk) and, in the affirmative, compute how this can be done.



PRESENTATION OF THE COURSE

RECONSTRUCTING 3D OBJECTS FROM PLANE SECTIONS

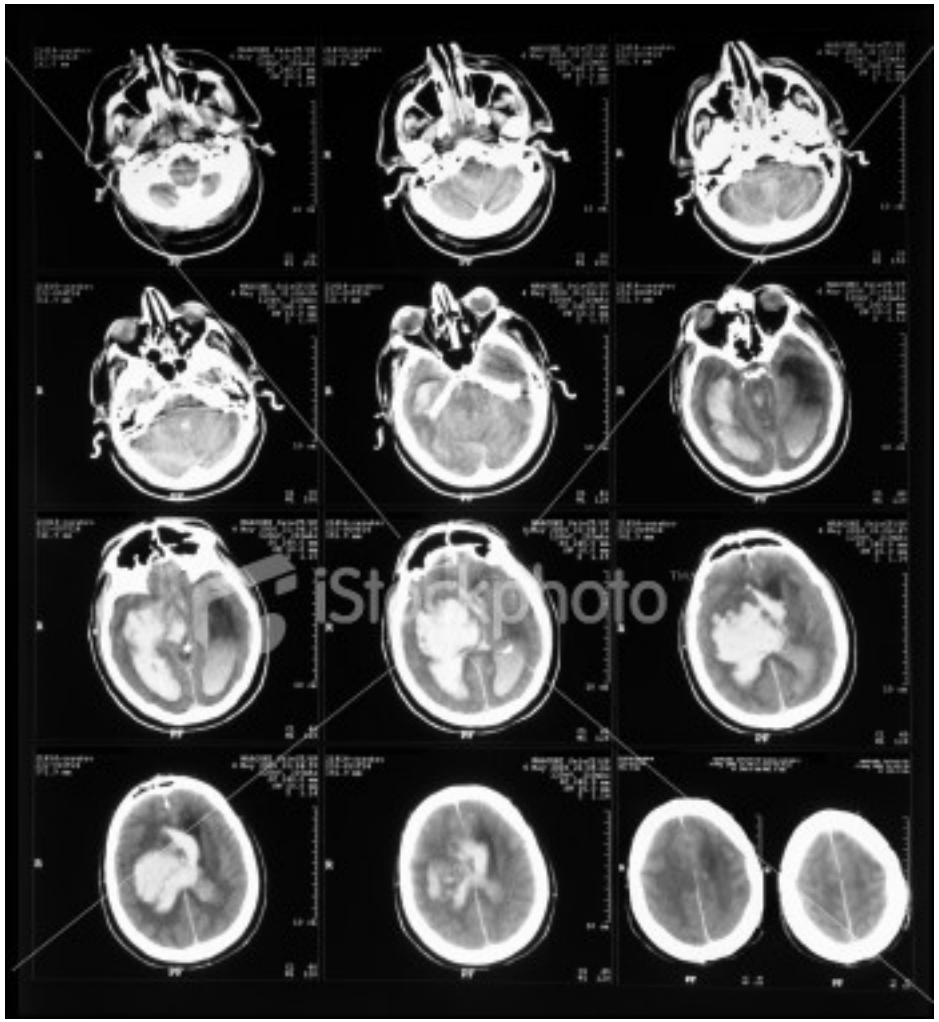
Develop efficient methods to reconstruct three-dimensional shapes from the information of a certain number of **plane sections** of the original objects.



PRESENTATION OF THE COURSE

RECONSTRUCTING 3D OBJECTS FROM PLANE SECTIONS

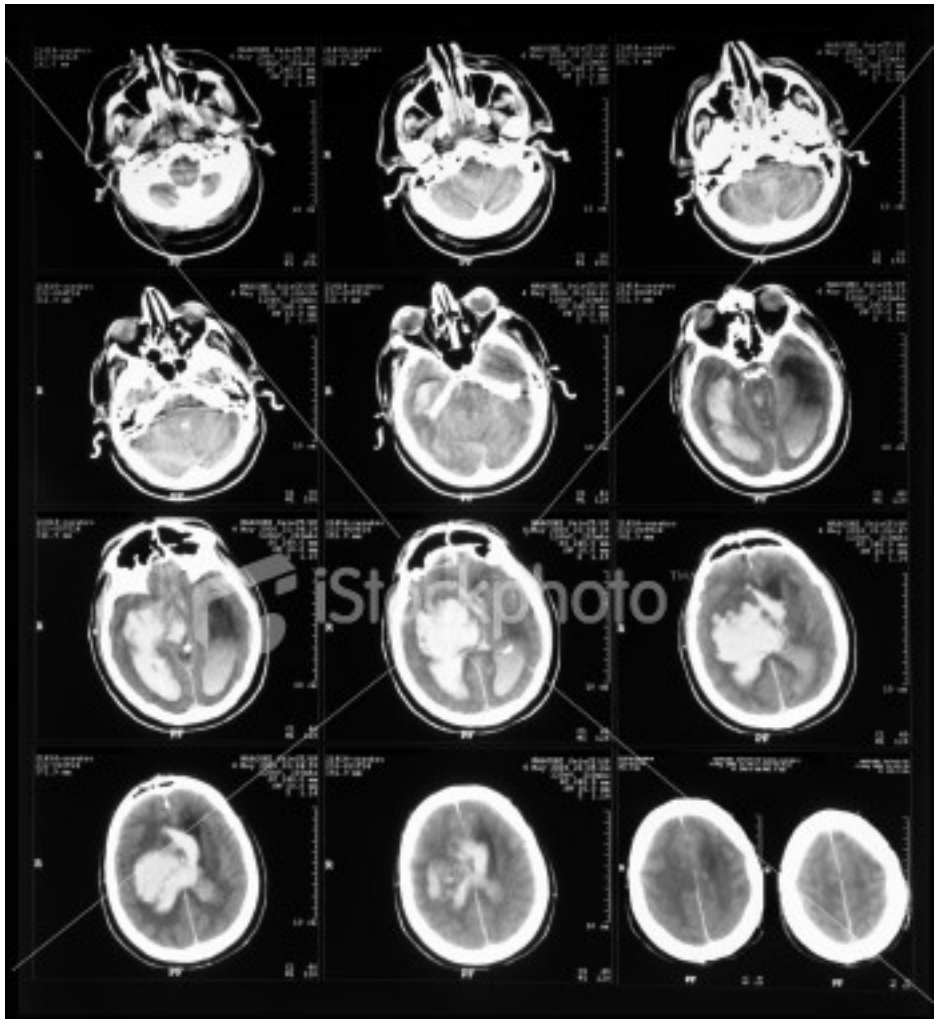
Develop efficient methods to reconstruct three-dimensional shapes from the information of a certain number of **plane sections** of the original objects.

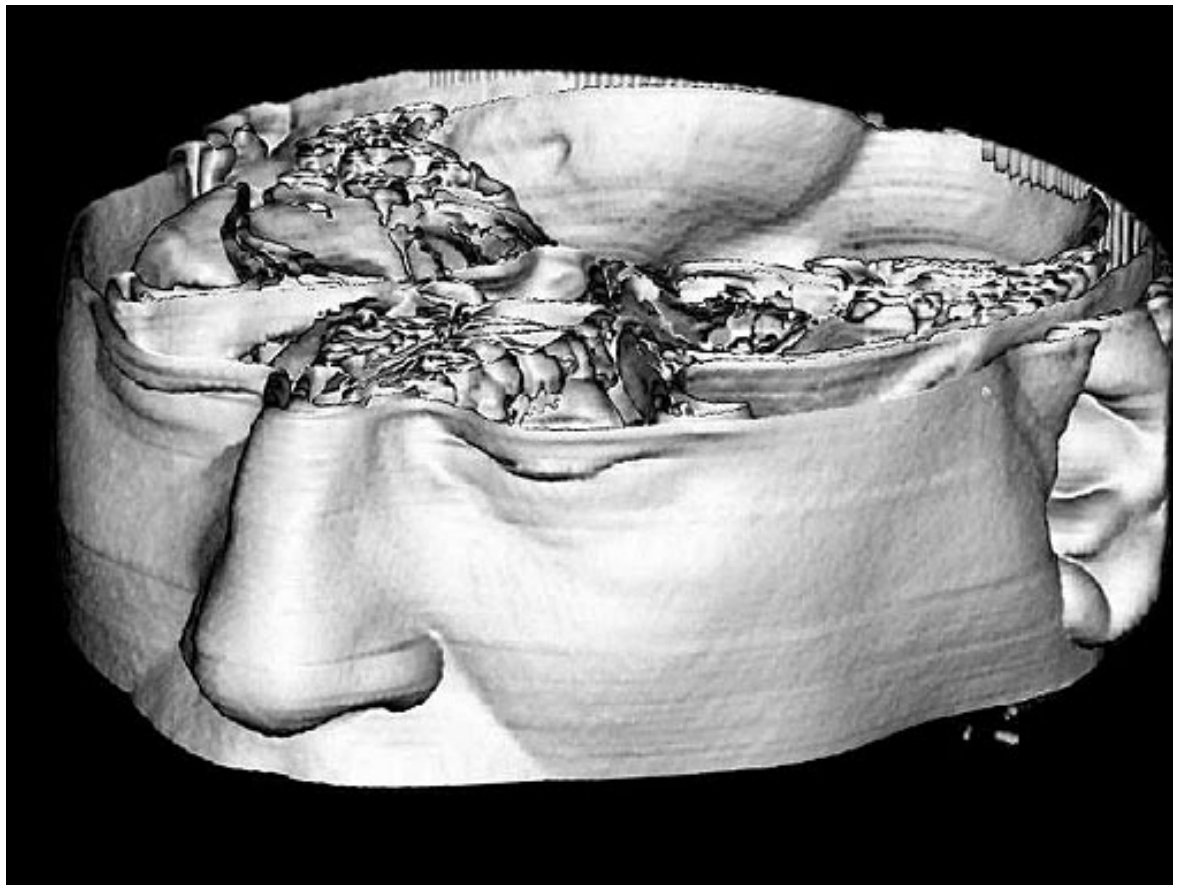


PRESENTATION OF THE COURSE

RECONSTRUCTING 3D OBJECTS FROM PLANE SECTIONS

Develop efficient methods to reconstruct three-dimensional shapes from the information of a certain number of **plane sections** of the original objects.

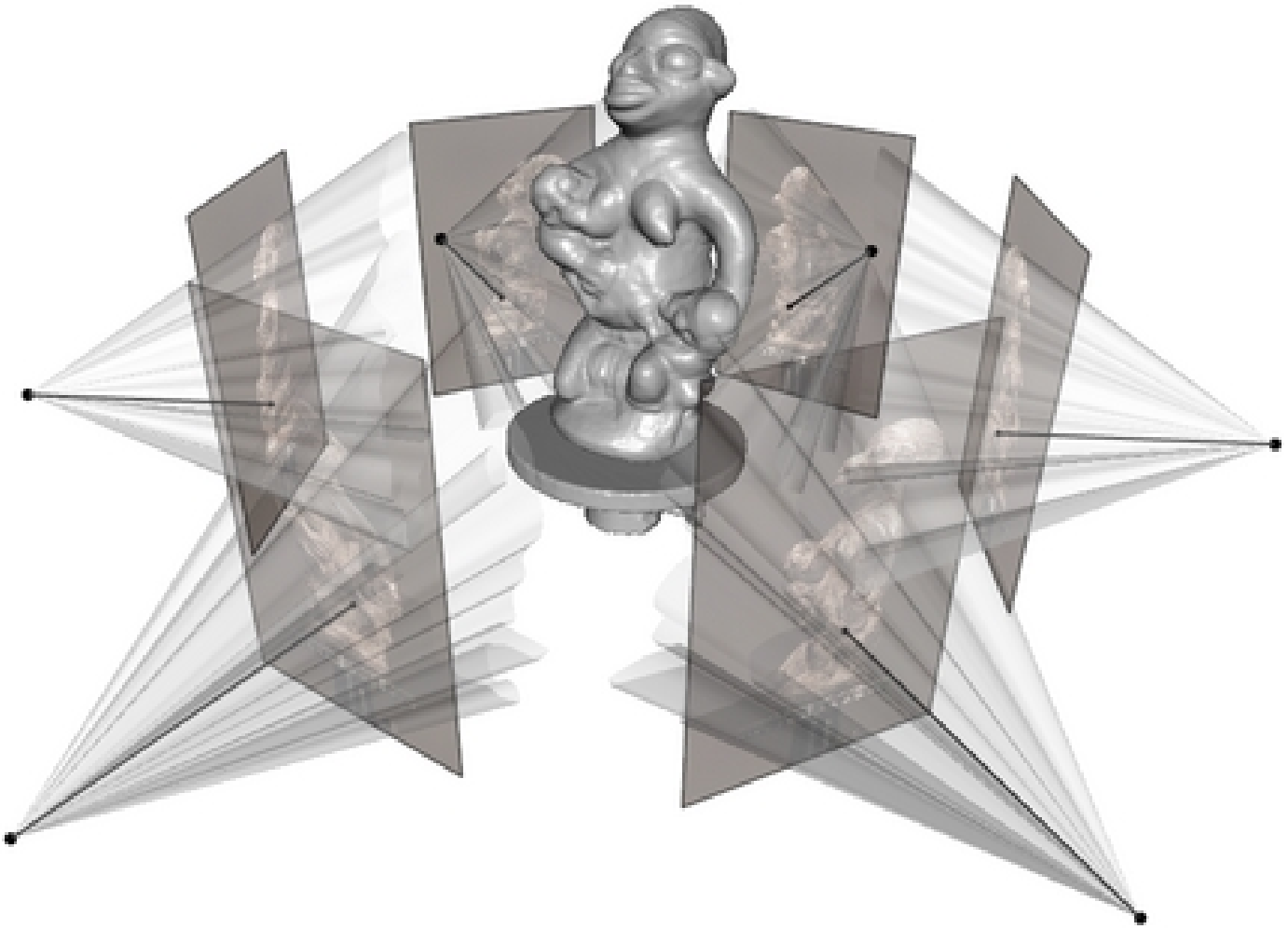




PRESENTATION OF THE COURSE

RECONSTRUCTING 3D OBJECTS FROM PLANE PROJECTIONS

Develop efficient methods to reconstruct three-dimensional shapes from the information of a certain number of **plane projections** of the original objects.



PRESENTATION OF THE COURSE

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
- ...

COMPUTATIONAL GEOMETRY

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

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

PRESENTATION OF THE COURSE

COMPUTATIONAL GEOMETRY

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

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

The kernel of **Computational Geometry** is the design and analysis of efficient algorithms to solve geometric problems.

PRESENTATION OF THE COURSE

COMPUTATIONAL GEOMETRY

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

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

The kernel of **Computational Geometry** is the design and analysis of efficient algorithms to solve geometric problems.

- 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

AN EXAMPLE

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

AN EXAMPLE

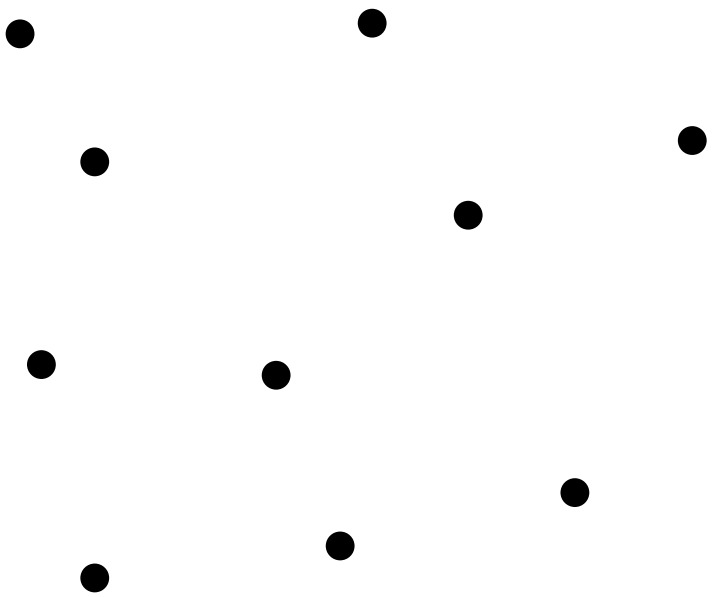
Given n points in the plane (population) find the optimal location of a service to attend this population (antenna, hospital, supermarket,...).

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

AN EXAMPLE

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

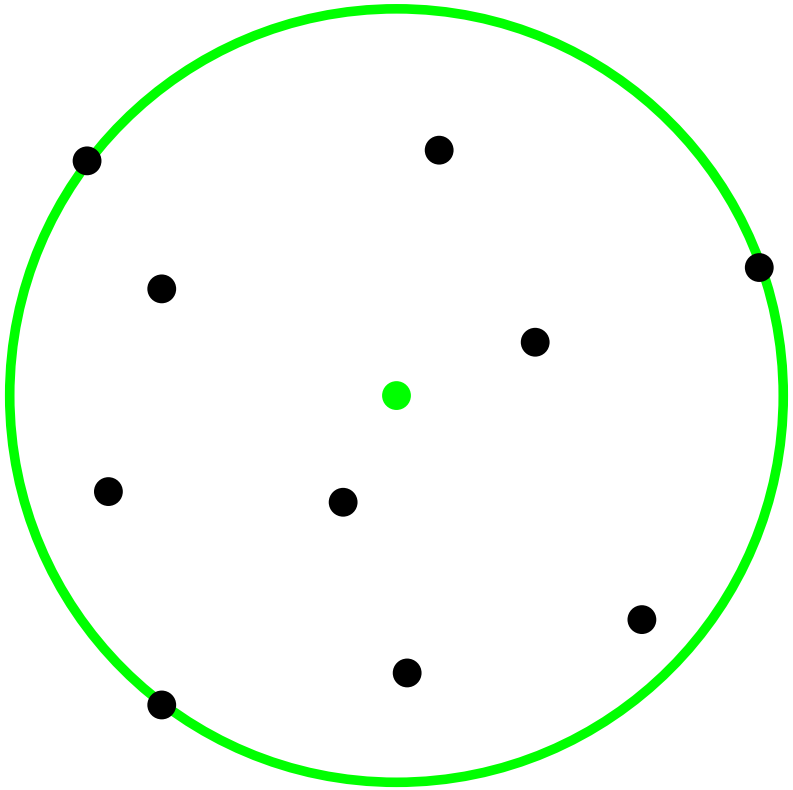
Given $(x_1, y_1), \ldots (x_n, y_n)$, find (x, y) achieving $\min_{(x,y)\in\mathbb{R}^2} \max_{i=1\ldots n} (x - x_i)^2 + (y - y_i)^2$.



AN EXAMPLE

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

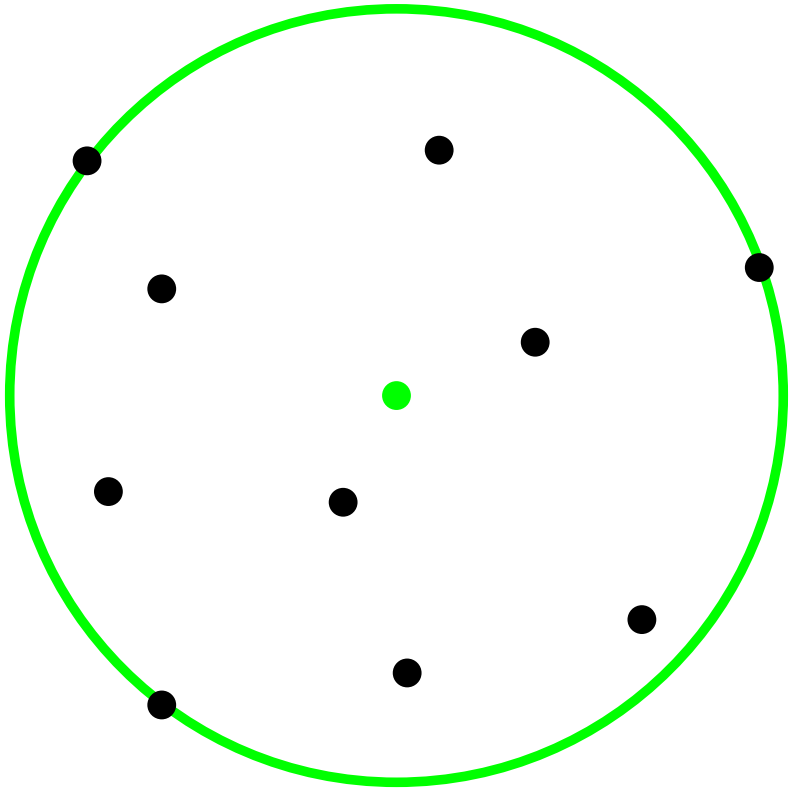
Given $(x_1, y_1), \ldots (x_n, y_n)$, find (x, y) achieving $\min_{(x,y)\in\mathbb{R}^2} \max_{i=1\ldots n} (x - x_i)^2 + (y - y_i)^2$.



AN EXAMPLE

Given n points in the plane (population) 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.

PRESENTATION OF THE COURSE

A BRIEF HISTORY OF COMPUTATIONAL GEOMETRY

Name initially used in different contexts

- Minsky's book "Perceptrons" (1969)
  About pattern recognition
- Forrest paper (1971)
  About curves and surfaces for geometric modeling
- Shamos' PhD Thesis "Computational geometry" (1975)

Nowadays widely recognized as a subfield of **Algorithm Theory**

- Within Theoretical Computer Science (TCS)

  TCS includes many other important areas like complexity theory, analysis of algorithms, formal semantics, etc.

PRESENTATION OF THE COURSE

A BRIEF HISTORY OF COMPUTATIONAL GEOMETRY

Computational geometry today:

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

Better names are possible (and are sometimes used)

- Algorithmic geometry
- Geometric algorithms

A BRIEF HISTORY OF COMPUTATIONAL GEOMETRY

How it started

- Geometry has 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

LA

GEOMETRIE.

LIVRE PREMIER.

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

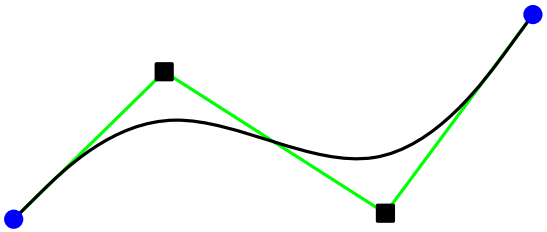
TOus les Problesmes de Geometrie se peuuent facilement reduire a tels termes, qu'il n'eſt beſoin par apres que de connoiſtre la longeur de quelques lignes droites, pour les conſtruire.

Et comme toute l'Arithmetique n'eſt compoſée, que de quatre ou cinq operations, qui ſont l'Addition, la Soustraction, la Multiplication, la Diuiſion, & l'Extraction des racines, qu'on peut prendre pour vne eſpece de Diuiſion : Ainſi n'a'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

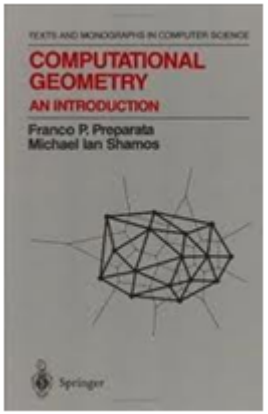
A BRIEF HISTORY OF COMPUTATIONAL GEOMETRY

How it actually started

- In the late 19th Century, Hermann Minkowski puts Delone[a] triangulations and Voronoi Diagrams to use in the Geometry of Numbers

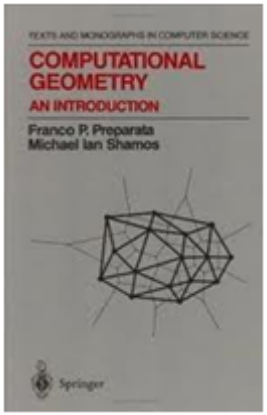


- Scattered work starts appearing in 60's and 70's
  - Bézier splines, combinatorial geometry (Erdös)
- As we know it today, started in the 70's
  - Geometric problems gain attention from algorithms researchers
- Michael Shamos' PhD thesis (1975)
  - "Problems in Computational Geometry"
- Increasing interest during late 70's, 80's
  - Most "basic algorithms" date from this period



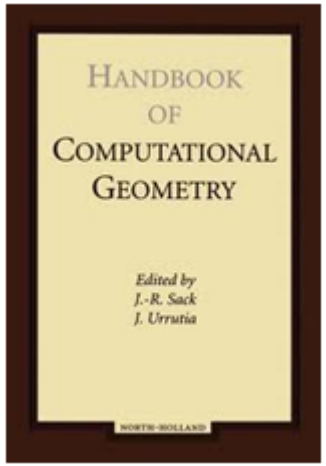
[a] Minkowski used the German transliteration *Delone* of the Russian lastname of Boris Delone. Today, the French transliteration *Delaunay* is more common in Computational Geometry.

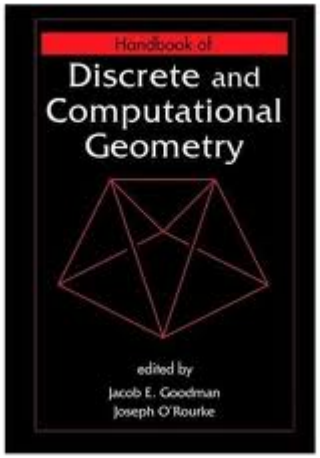
A BRIEF HISTORY OF COMPUTATIONAL GEOMETRY



Taking off: Between late 80's and 90's it becomes a very active field

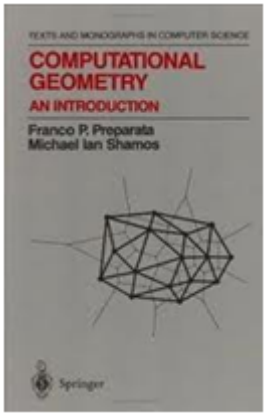
- 1983: First European workshop on the topic
- 1985: First Annual Symposium on Computational Geometry
  Also: first textbook (today, more than 5)
- 1996: CGAL: first serious implementation of a robust geometric algorithms library
- 1997: First handbook on the topic (second in 2000)





PRESENTATION OF THE COURSE

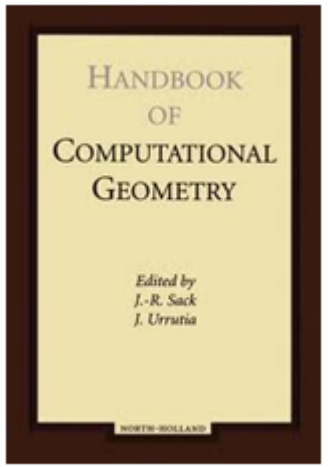
A BRIEF HISTORY OF COMPUTATIONAL GEOMETRY

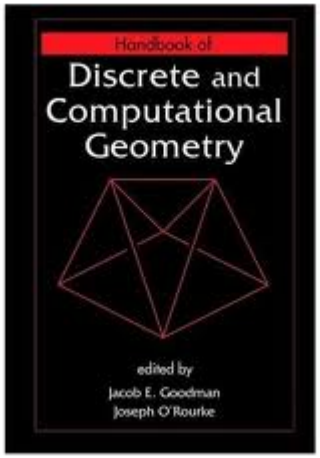


Taking off: Between late 80's and 90's it becomes a very active field

- 1983: First European workshop on the topic
- 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)

Today





- Recognized discipline within TCS
- "Large" active community
  Many research groups in Europe... and Spain!
  (BCN, Madrid, Sevilla, Zaragoza, Girona, Valladolid,...)
- 4 specialized journals devoted to it
- 3 annual specialized conferences, and an important presence in Algorithms conferences

PRESENTATION OF THE COURSE

GEOC

Professors (Q1 2011/2012)

- **Julian Pfeifle** (first half of the course)
- **Rodrigo Silveira** (second half of the course)

Dr. Mohammed Marey

Scientific Computing Department

Goals

- Knowing the wide range of problems studied in Computational Geometry and its solutions, as well as its applications.
- Understanding the power of combining geometric tools with the most appropriate data structures and algorithmic paradigms.
- Seeing in action several algorithmic paradigms and data structures useful in geometric problems.
- Applying geometric results to real problems.

GEOC

Syllabus

1. Introduction to Computational Geometry
2. A basic tool (left and right)
3. Geometric problems on polygons
4. Convex hull
5. Duality. Intersection of halfplanes. Linear programming
6. Triangulating polygons
7. Proximity
8. Triangulating sets of points
9. Arrangements
10. Point location
11. Students' presentations (6 h)

PRESENTATION OF THE COURSE

GEOC

Methodology

- Theory: exposition of the subject by the instructor and, later on, by the students.
- Problems: previously assigned, they will be presented in class by the students.
- Lab: implementing some of the algorithms presented in class (common part) and one algorithm not necessarily presented (to be chosen by each student).

Evaluation

- Final Exam
- Midterm Exam
- Assignment
- Quiz
- activities)
- Tasks
- Projects

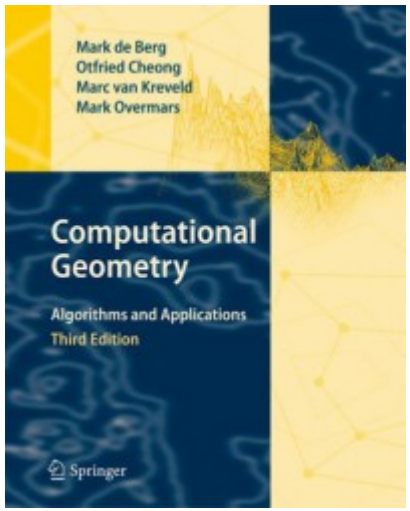
PRESENTATION OF THE COURSE

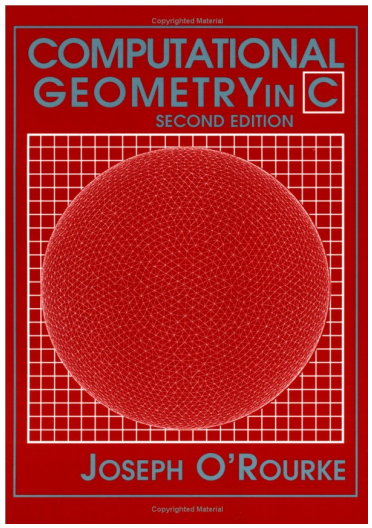
GEOC

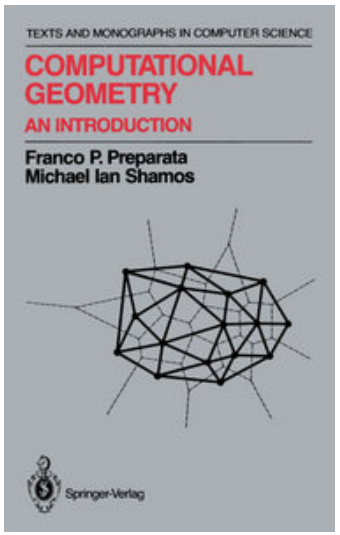
Bibliography

- **Computational Geometry and Computer Graohics in C++**

- M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, **Computational Geometry: Algorithms and Applications (3rd rev. ed.)**, Springer, 2008.
- J. O'Rourke, **Computational Geometry in C (2nd ed.)**, Cambridge University Press, 1998.
- F. Preparata, M. Shamos, **Computational Geometry: An introduction (revised ed.)**, Springer, 1993.







GEOC

Complementary bibliography

- J-D. Boissonnat, M. Yvinec, **Algorithmic Geometry**, Cambridge University Press, 1998.
- J. R. Sack, J. Urrutia, **Handbook on computational geometry**, Elsevier, 2000.
- J. E. Goodman, J. O'Rourke, **Handbook on Discrete and Computational Geometry**, CRC Press, 1997.
- S. Devadoss, J. O'Rourke, **Discrete and Computational Geometry**, Princeton University Press, 2011.

GEOC

Complementary bibliography

- J-D. Boissonnat, M. Yvinec, **Algorithmic Geometry**, Cambridge University Press, 1998.
- J. R. Sack, J. Urrutia, **Handbook on computational geometry**, Elsevier, 2000.
- J. E. Goodman, J. O'Rourke, **Handbook on Discrete and Computational Geometry**, CRC Press, 1997.
- S. Devadoss, J. O'Rourke, **Discrete and Computational Geometry**, Princeton University Press, 2011.

GEOC

Complementary bibliography

- J-D. Boissonnat, M. Yvinec, **Algorithmic Geometry**, Cambridge University Press, 1998.
- J. R. Sack, J. Urrutia, **Handbook on computational geometry**, Elsevier, 2000.
- J. E. Goodman, J. O'Rourke, **Handbook on Discrete and Computational Geometry**, CRC Press, 1997.
- S. Devadoss, J. O'Rourke, **Discrete and Computational Geometry**, Princeton University Press, 2011.

GETTING TO KNOW A BIT MORE

Here are some links that can help you getting an idea of what Computational Geometry is:

For the lazy, a short and clear article explaining what Computational Geometry is, and what it applies to:

http://softsurfer.com/overview.htm

For those who want to see it in action, many applets made by the Computational Geometry students of the Facultad de Informática de Madrid:

http://www.dma.fi.upm.es/docencia/segundociclo/geomcomp/aplicaciones.html

For those who want to know it all, here are some web pages with more information:

Computational Geometry Pages (Jeff Erickson):

http://compgeom.cs.uiuc.edu/∼jeffe/compgeom

Geometry in Action (David Eppstein):

http://www.ics.uci.edu/∼eppstein/geom.html

Computational Geometry links (Godfried Toussaint):

http://www-cgrl.cs.mcgill.ca/∼godfried/teaching/cg-web.html