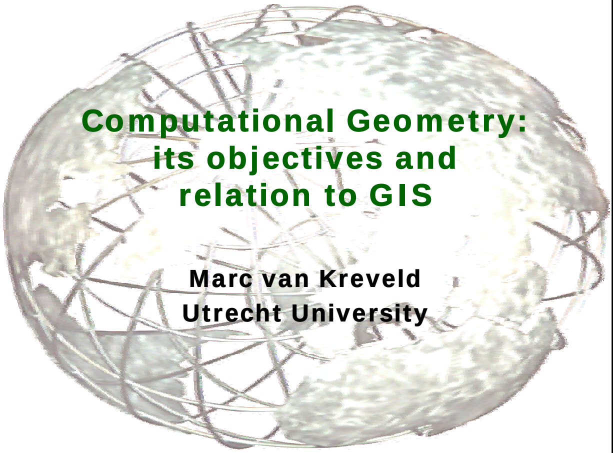


Computational Geometry: its objectives and relation to GIS

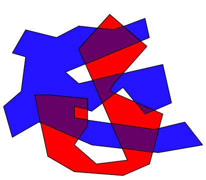
Marc van Kreveld
Utrecht University

Overview

- Brief introduction to computational geometry (CG):
  - Computational geometry
  - Input size and efficiency
  - Some basic problems studied
- Old GIS problems in CG
- New GIS problems for CG?
- Improving the use of CG for GIS

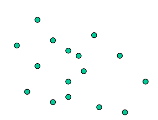
Computational Geometry

- Research area that deals with geometric algorithms (points, lines, distances, ...)
- Provable efficiency, exact problem statements, guarantees in case of approximation
- Applications in robotics, computer graphics, pattern matching, games, CAD/CAM and GIS



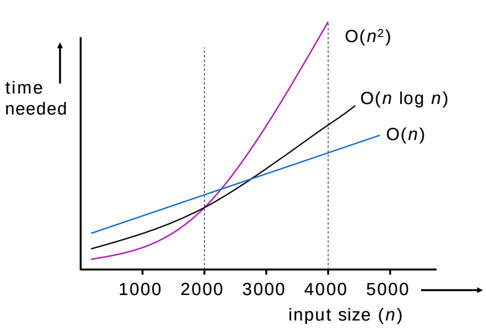
Example: closest pair

- **Problem**: Given a set of n points in the plane, find the two that are closest together
- **Input**: n points, each given by two coordinates
- **Output**: the two closest points
- **Algorithm**: sequence of steps that gives the output
- **Efficiency**: time needed to compute the output



(0.1245, 0.7847)
(3.5209, 2.2146)
(1.0177, 8.9521)
....
....

Input size and efficiency

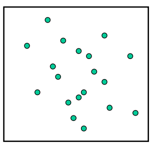
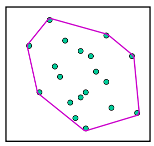


Efficiency

- It takes linear - O(n) - time to look at all objects of the input
- It takes quadratic - O(n^2) - time to consider every pair of input objects
- For query data structures, query times lower than O(n) are the goal (like O(log n) or O($\sqrt{n}$) time): avoid looking at all objects in the data structure to answer a query
  Examples: quadtree, kd-tree, R-tree

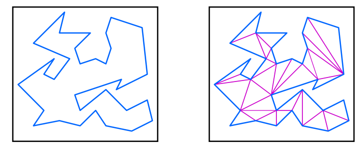
Classical computational geometry

- **Convex hull** of n points in the plane: Compute the smallest convex set that contains the points
- $O(n \log n)$ time, same in 3D
- Higher dimensions: $O(n^{\lceil d/2 \rceil})$ time


Classical computational geometry

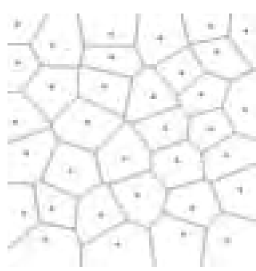
- **Polygon triangulation**: given a simple polygon with n vertices, determine a partition into triangles by connecting vertices by new edges
- $O(n)$ time is possible (but complicated)
- 3D: $O(n^2)$ time



Classical computational geometry

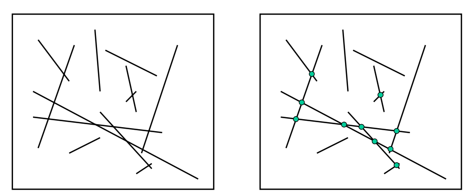
- **Voronoi diagram**: for n points, compute the subdivision of the plane such that in one cell, one point is closest
- $O(n \log n)$ time
- Higher dimensions: $O(n^{\lceil (d+1)/2 \rceil})$ time

Useful for neighborhood analysis in GIS



Classical computational geometry

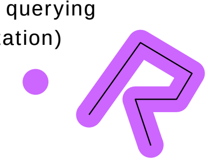
- **Line segment intersection**: for n line segments, determine all intersections
- $O(n \log n + k)$ time, if there are k intersections ($0 \leq k \leq n^2/2 - n/2$) → *output-sensitive algorithm*



Useful for thematic map overlay in GIS

Classical computational geometry

- Delaunay triangulation
- Linear programming in fixed dimensions
- Geometric data structures and querying
- Minkowski sum (buffer computation)
- Arrangements of lines
- ....



- Typically:
  - A well-specified problem that involves geometric objects
  - An algorithm that solves the problem for any instance
  - A time analysis that applies to any instance (upper bound on running time - scalability)

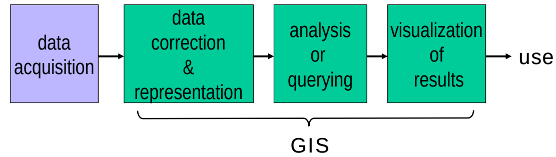
More on efficiency

- Some problems require (too) much time to solve (for the application): $O(n^2)$, $O(n^3)$, $O(2^n)$, ...
- Sometimes due to worst-case analysis; realistic input models may help to explain practical efficiency
- Otherwise, heuristics and approximation algorithms should be used

Example: Traveling sales person

- NP-hard: worse than $O(n^c)$ for any constant c
- An easy, efficient factor-2 approximation exists
- Many heuristics exist

GIS: from data to output



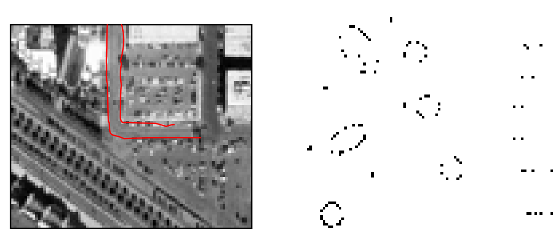
Geographical analysis: process of deriving further information from the data for a task

Old GIS problems in CG

- … in data preprocessing:
  - Automated checking and correction
  - Automated integration
- … in spatial data analysis:
  - Thematic map overlay
  - Spatial interpolation
- … in spatial data visualization:
  - Automated generalization
  - Automated text placement

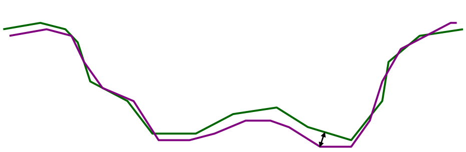
Data acquisition and preprocessing

- Digitizing paper maps or aerial photos:
  - Detect doubly digitized lines, overshoots, …
  - Add topology



Doubly-digitized line detection

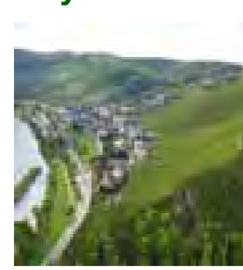
- Need a similarity measure for two polylines
- Need an algorithm to compute this measure



Hausdorff distance: maximum of closest distances to the other polyline; computable in $O(n \log n)$ time

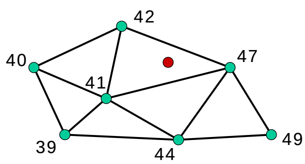
Spatial data analysis

- Suitability analysis, e.g. wine growing requires
  - If hilly, hills facing South
  - Not too steep sloped
  - Good soil type
  - Temperature, precipitation, sun
- Spatial interpolation
- Geo-statistics
- Terrain analysis
- Hazard analysis (flooding, fire, avalanches)
- Trend analysis (for spatio-temporal data)
- Spatial and spatio-temporal data mining



Spatial interpolation

- Piecewise linear (triangulation)
- Moving windows
- Kriging
- Natural neighbor



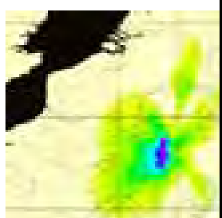
Delaunay triangulation: triangulation that maximizes the smallest occurring angle (over all triangulations); computable in O(*n* log *n*) time

Data storage for efficient querying

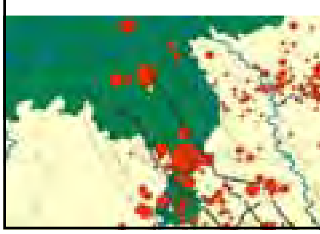
- Geometric data structures allow selection of features in specified areas, without having to examine all features
- Possible query types are windowing queries (rectangular search), nearest neighbor queries, and intersection queries
- More on this in a talk later today (with a focus on I/O-efficiency, by Mark de Berg)

Visualization

- Visualize results of analysis
- Compute a map
  - general/special purpose
  - cartographic generalization
  - label placement



outcome of air pollution analysis



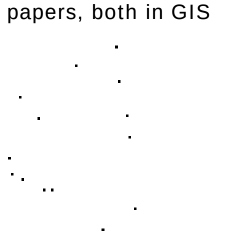
earthquakes and break lines

Label placement

- Placing text with features on the map
  - Point features: cities on small-scale maps
  - Line features: roads and rivers
  - Area features: country names, lakes
- Text should be placed
  - Unambiguously associated to its feature
  - Non-intersecting with other text
  - Readable
  - Non-obscuring important features

Label placement

- Maximizing the number of labels placed without overlap is NP-hard
- There are many research papers, both in GIS and CG (>100)
- Good approximation algorithms and heuristics exist, good software exists



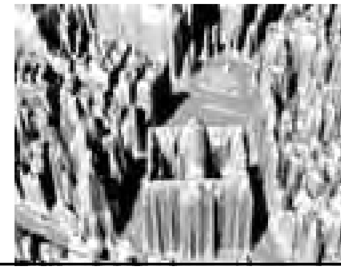
Output of MapText

New GIS problems in CG?

- Still from data preprocessing, analysis, and visualization
- More complex problems:
  - More criteria at once
  - Less clear how to formalize

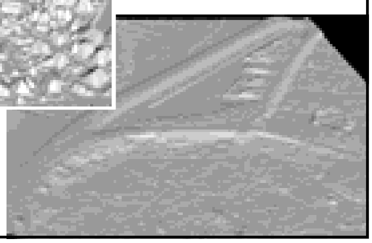
More spatial data correction

- Correction of the Dutch elevation data set AHN (Actueel Hoogtebestand Nederland)

Spatial data correction

Filtering elevation data to remove cars, trees, towns ...



Spatial data correction

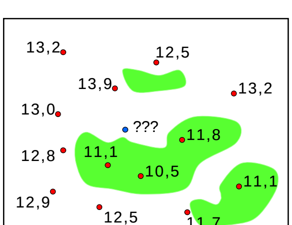
Can we extract building information automatically and improve it?

Can we remove trees, cars, … automatically?

- Use knowledge of the data:
  - An elevation data set in urban areas should have many vertical sides, right angles, and planar areas
- Use other data sets (spatial data conflation):
  - Combine with cadastral information on the outlines of the buildings
  - Combine with road data to help locate roads and therefore cars

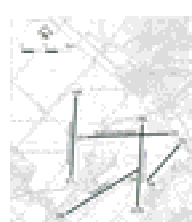
Spatial interpolation

- Natural spatial interpolation may involve obstacles or influencing regions
  - noise spreading is influenced by land cover
  - soil humidity is influenced by soil type



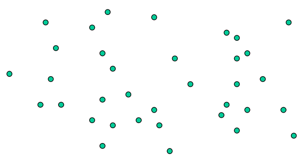
Spatial interpolation

- For noise contour computation, wind also is important (direction and speed; anisotropy)




Spatio-temporal data mining

- In large collections of spatio-temporal data, find patterns, regularities, clusters, …
- E.g. trajectories of hurricanes, epicenters of earthquakes, migration routes of birds, …



Epicenters of earthquakes may show a tendency to move to the Northeast over time

→ can be quantified and computed

Visualization: Cartographic generalization

- Changes to the geometry to display a map at a smaller scale
- Continuous topic of research

1:25,000

→

1:50,000

Visualization: special-purpose maps

- Schematic maps, flow maps, cartograms, statistical maps
- Also multi-criteria problems
- Discussed more in a talk later today (Bettina Speckmann)

Discussion

- There are still many GIS problems to be solved by geometric algorithms. These include:
  - Data correction
  - Cartographic generalization
  - Realistic interpolation
  - Spatio-temporal data mining
  - Visualization

Also of interest, but not discussed:
  - Algorithms for spatial processes
  - Dealing with data imprecision, estimating error

Discussion

- These GIS problems are not easy to state precisely
- How to state (formalize) GIS problems so that they can be solved algorithmically requires expertise from GIS and CG researchers

→ how do different criteria influence the result and efficiency of a GIS problem?