Chapter III

Geographic Information Systems: Data Processing

GIS: data processing

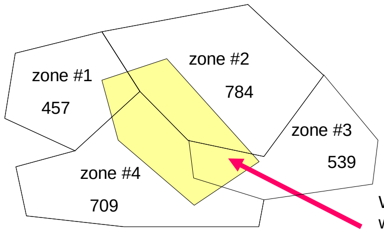
3.1 – Spatial queries

- 1. Example of spatial queries
- 2. Elementary spatial queries
- 3. Queries of spatial analysis
- 4. Topological queries
- 5. Conclusion

3.1.1. Example of spatial queries

- What do we have in this point?
- What do we have in this zone ?
- What is the best path from Lisbon to Warsaw
- What are the countries at the border of Austria?
- What are the states crossed by the Mississippi river?
- Where is the more polluted zone?

Example of spatial query



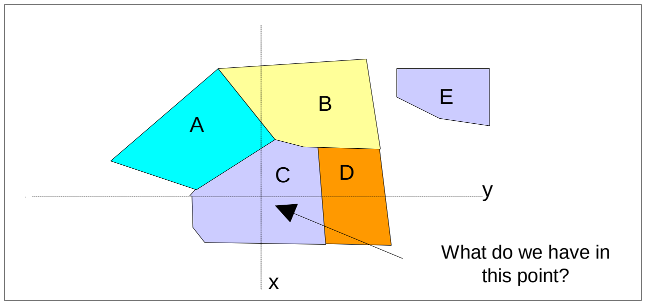
#zone	Number of trees
1	457
2	784
3	539
4	709

What is the number of trees within the zone arbitrarily designed?

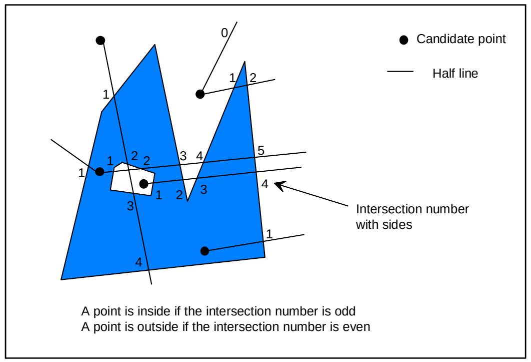
3.1.2. Elementary spatial queries

- Point query
- Line query
- Region query
- 3D query
- Buffer zones

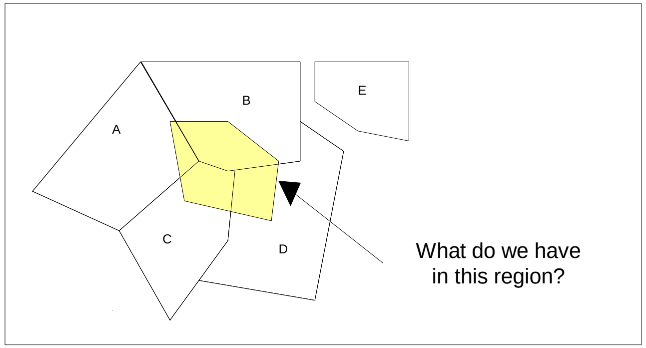
Point query



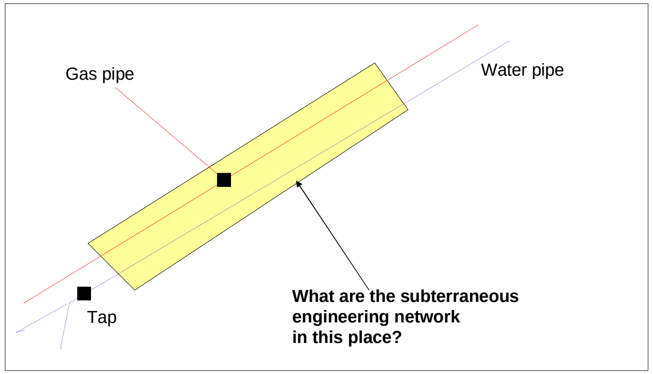
Jordan half-line theorem



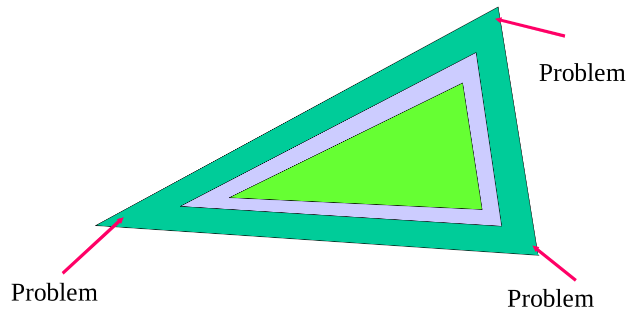
Region query



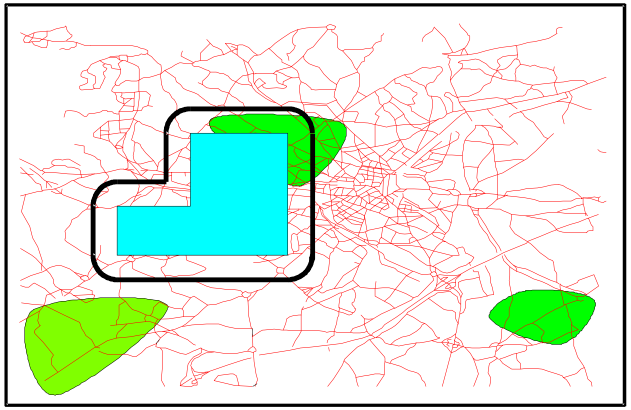
Trench query



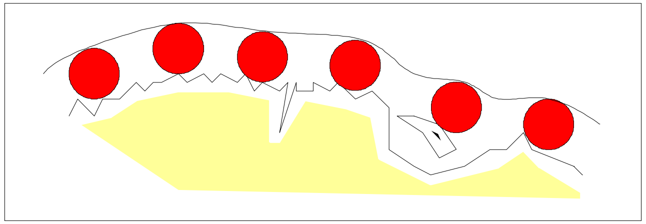
Buffer zones defined by parallels



Buffer zone



Definition of a buffer zone along a jagged polygon

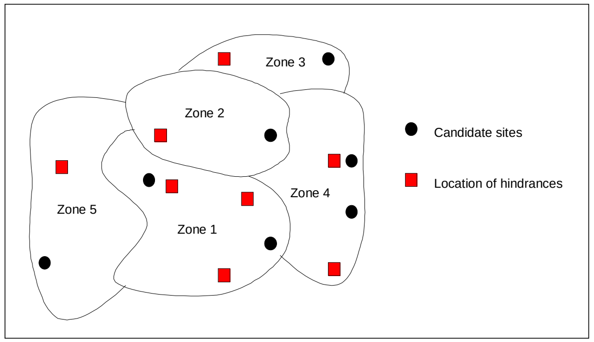


Example: delimitation of sea territorial waters

3.1.3. Queries of spatial analysis

- Optimal point
- Optimal zone
- Optimal path
- Districting

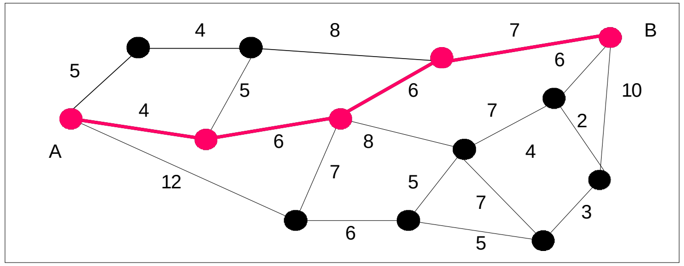
Locating a new hospital



Optimization queries

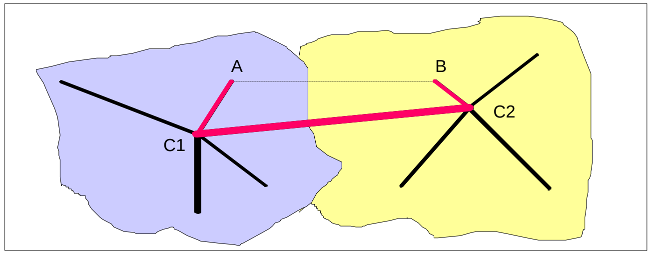
- Usually solved by operation research methods
- Definition of one or several criteria
- Finding for the optimum
  - Gradient (hill-climbing) algorithm
  - Muticriteria methods

Optimal path in a graph



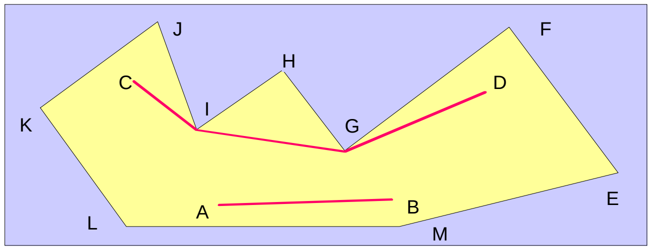
Solved by Dijkstra algorithm or variants

Path in a hierarchical graph



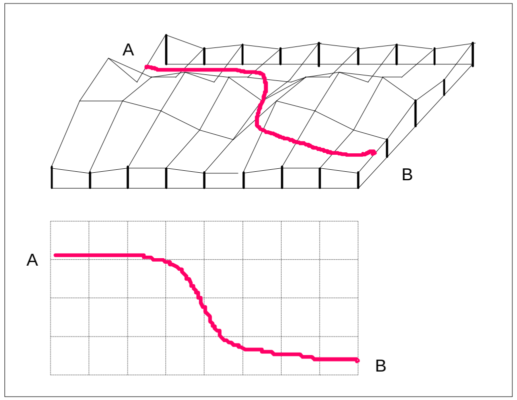
How to go from A to B?

Minimum path in a polygon

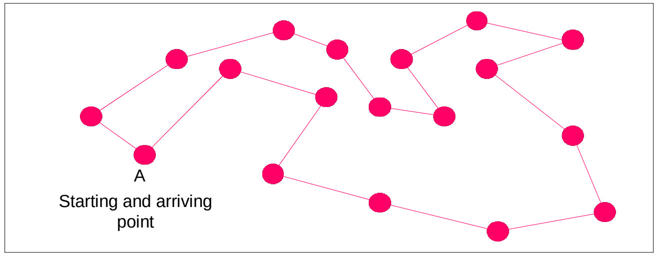


How to go from A to B?
How to go from C to D?

Minimum path in a terrain



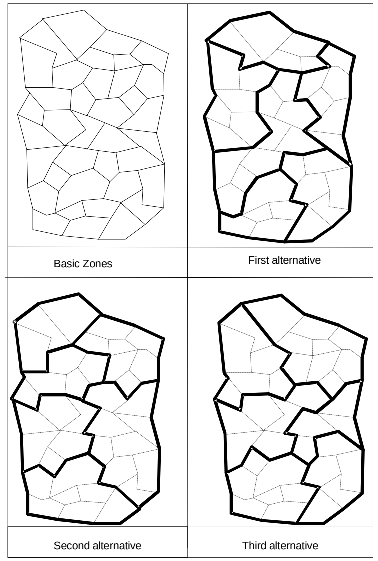
Salesman circuit



Districting

- Objective: find a tessellation following some criteria
- Example:
  - political election districts
  - local branch (subsidies) of a company

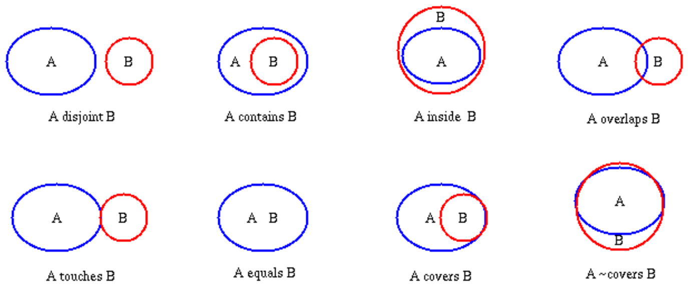
Example



3.1.4 – Topological queries

- Query about position and adjacency of spatial objects
- Allen and Egenhofer relations
- « touch », « intersect » etc.
- Object A :
  - inside: A°
  - outside: ¬A
  - border: δA

Egenhofer Relations



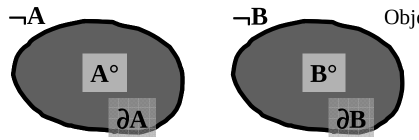
9 intersection Egenhofer Model

Object A:

inside : A°

outside : ¬A

border : δA



Object B:

inside : B°

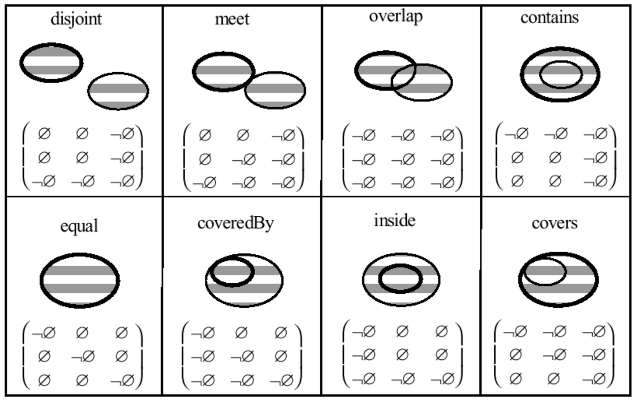
outside : ¬B

border : δB

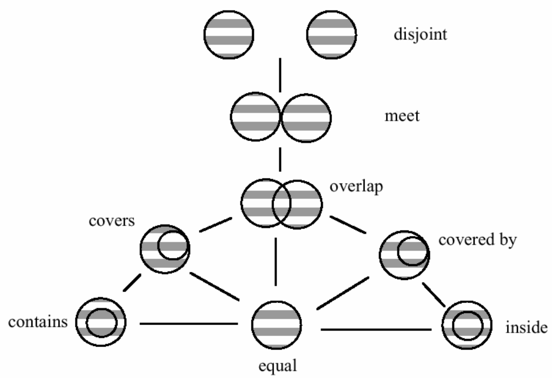
R(A,B) =

	B°	∂B	¬ B
A°	A° ∩B°	A°∩∂B	A°∩¬B
∂A	∂A∩B°	∂A∩∂B	∂A∩¬B
¬ A	¬A∩B°	¬A∩∂B	¬A∩¬B

$$R(A,B)=\begin{pmatrix} A^{\circ}\cap B^{\circ} & A^{\circ}\cap \partial B & A^{\circ}\cap B^{-} \\ \partial A\cap B^{\circ} & \partial A\cap \partial B & \partial A\cap B^{-} \\ A^{-}\cap B^{\circ} & A^{-}\cap \partial B & A^{-}\cap B^{-} \end{pmatrix}$$



Neighbouring



3.1.5. Conclusion about spatial queries

- Importance of spatial queries
- Topology
- Operation research
- Importance of response time
- Necessity of indexing (spatial indexing)

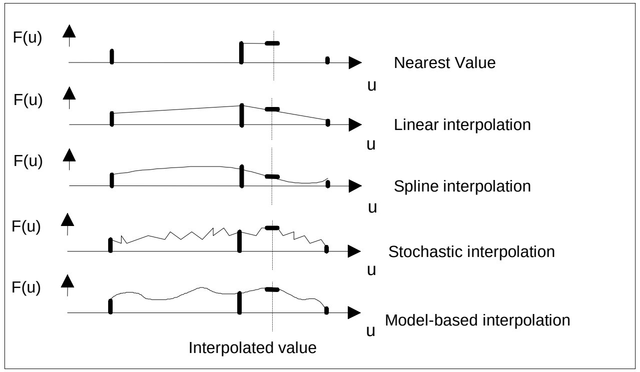
3.2 – Introduction to spatial analysis

- 1. Interpolation and extrapolation
- 2. Operation research
- 3. Spatial analysis by map overlay
- 4. Simulation methods
- 5. Multicriteria analysis
- 3. Examples
- 7. Conclusion

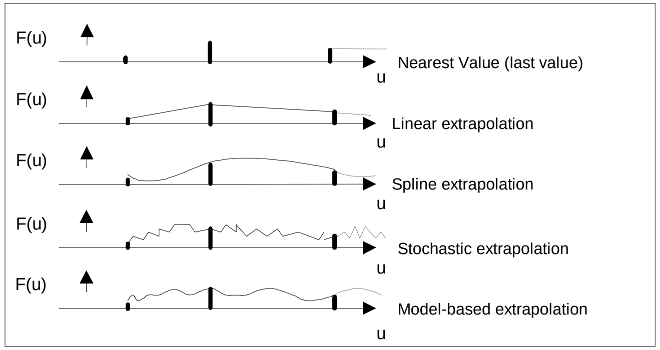
3.2.1 Interpolation and extrapolation

- 1. Data Interpolation
- 2. Data Extrapolation
- 3. Geometric Inference

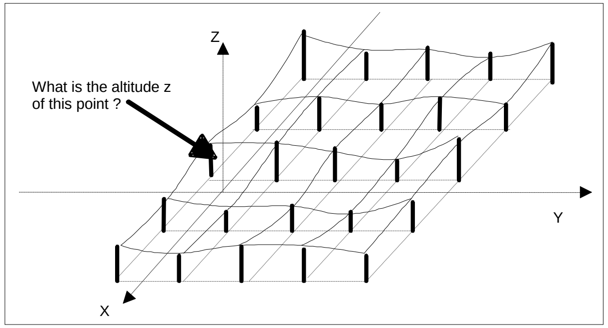
Various possibilities of interpolation



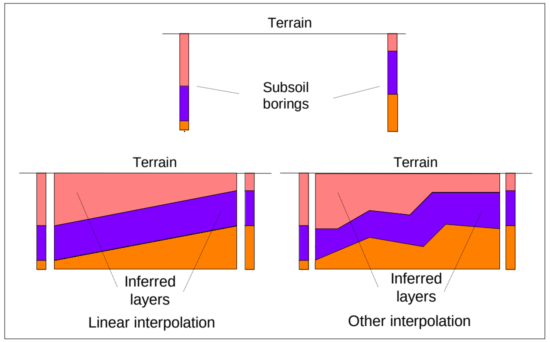
Various possibilities of extrapolation



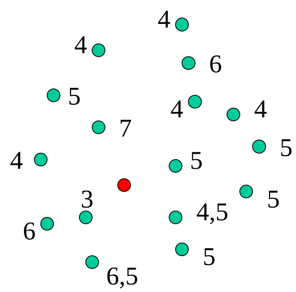
Geometric inference: estimation of altitude of a point



Geometric inference: geologic layers from boring



Calcul of influence: Newtonian interpolation



$$z_r = \frac{\sum_{i=1}^{n} \frac{z_i}{d_i^2}}{\sum_{i=1}^{n} \frac{1}{d_i^2}}$$

In which

$$d_i = \sqrt{(x_r - x_i) - (y_r + y_i)}$$

If we set

$$p_i = \frac{1}{d_i^2}$$

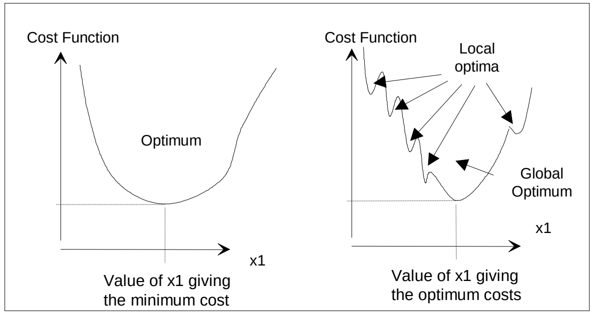
We get

$$z_r = \frac{\sum_{i=1}^{n} z_i * p_i^2}{\sum_{i=1}^{n} p_i^2}$$

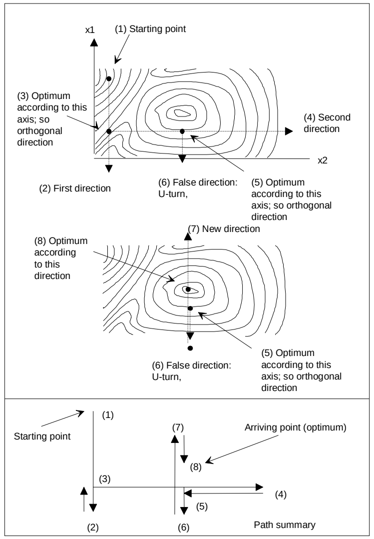
3.2.2 – Operation Research

- Simplex method
- Gradient method
- Optimal path

Optimizing a monovariable function



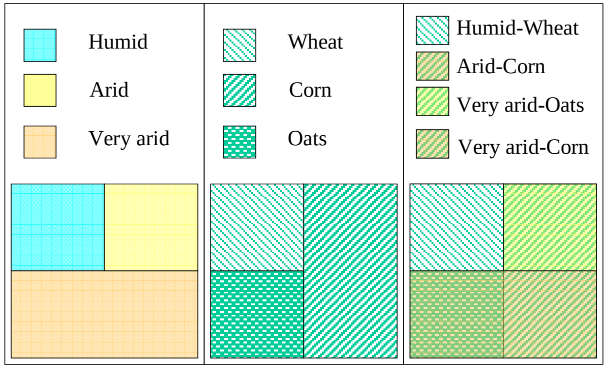
Searching the optimum of a function



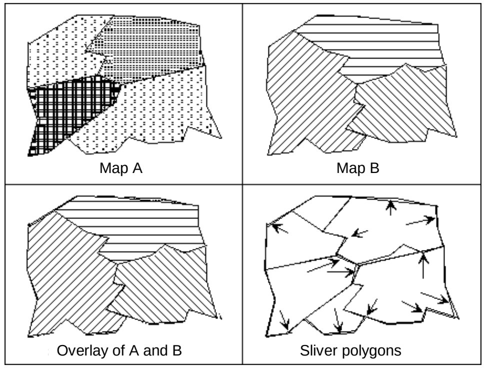
3.2.3 – Spatial Analysis by map overlay

- « overlay »
- Metaphor of the light table

Exact overlay



Map overlay with sliver polygons



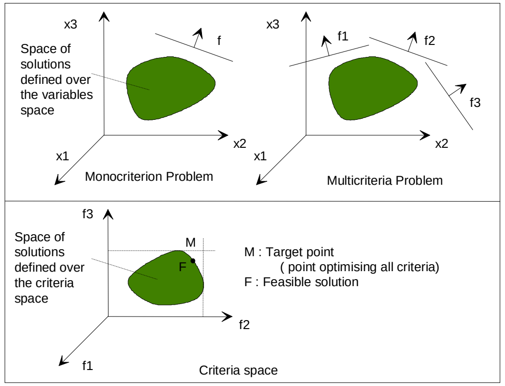
3.2.4 – Simulation methods

- Monte Carlo Simulation
- Statistical distribution a priori known
- Using random numbers
- Numerous simulations
- Computation of parameters (mean, variance, etc.)

3.2.5 – Multicriteria Analysis

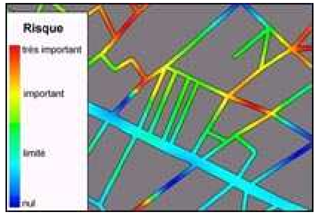
Min $f_1(x1, x2, x3,...., x_n)$
Max $f_2(x1, x2, x3,...., x_n)$
Min $f_3(x1, x2, x3,...., x_n)$
....
Min $f_i(x1, x2, x3,...., x_n)$
....
Max $f_k(x1, x2, x3,..., x_n)$
etc.

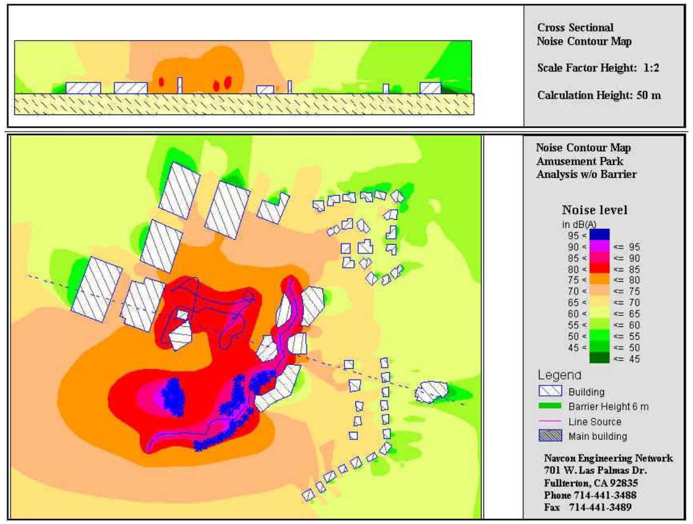
Multicriteria Optimization



3.2.6 – Examples

Road risk





3.2.7 – Conclusion about Spatial Analysis

- Importance of spatial analysis
  - points
  - lines
  - zones
  - graphs

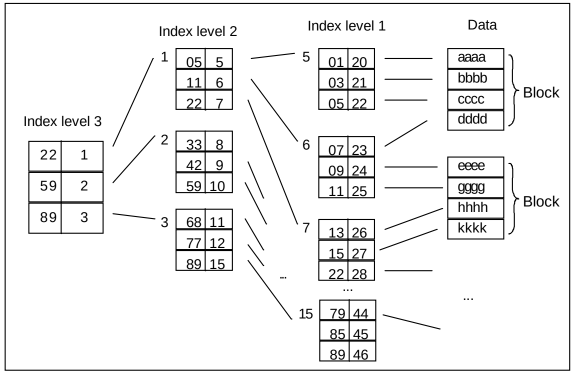
3.3 – Spatial indexing

3.3.1 – Importance of spatial indexing

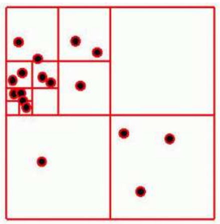
- Accelerating system
- Without index:
  - Browsing the whole DB (all objects)
  - Very time-consuming (expensive)
- Necessity of creating adapted data structures

Indexing in relational DB

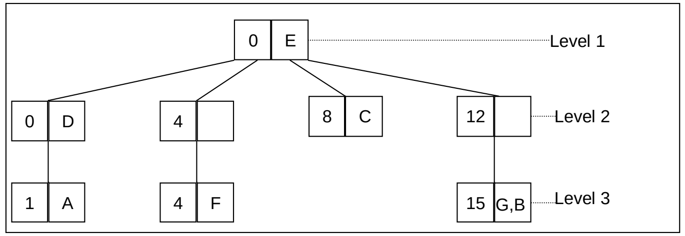
Hierarchy of indices



3.3.2 – Using quadtree



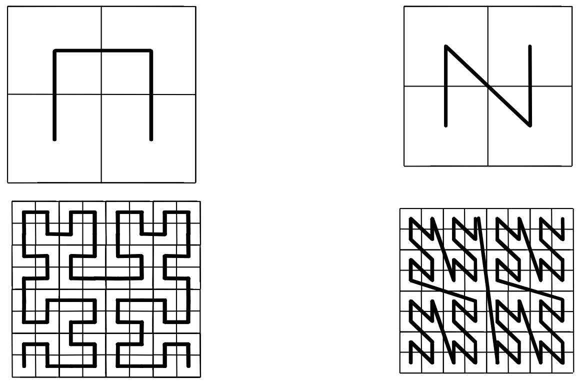
Quadtree



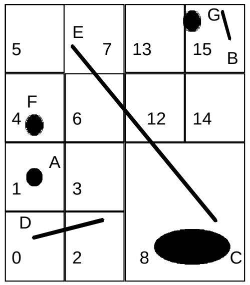
3.3.3 – Using Peano curves

- Space-filling curves
- Total coverage of the space
- Impossible with Euclidian geometry
- Possible within Peanian vision

Hilbert and Peano Curves

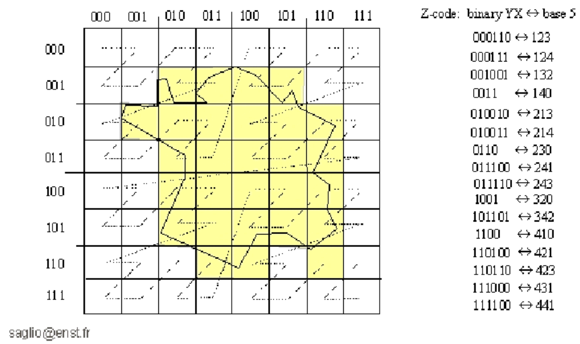


Indexing a small territory



Peano keys	Side	Objects
0	4	E
0	2	D
1	1	A
4	1	F
8	2	C
15	1	B,G

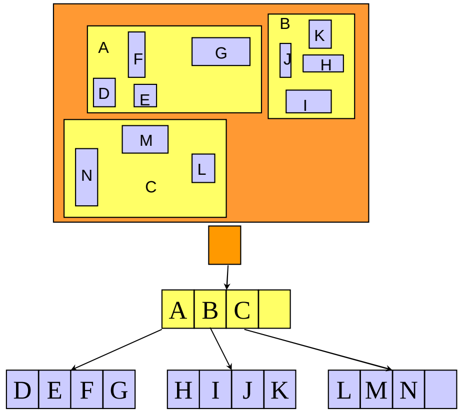
Z-order (Morton codes)



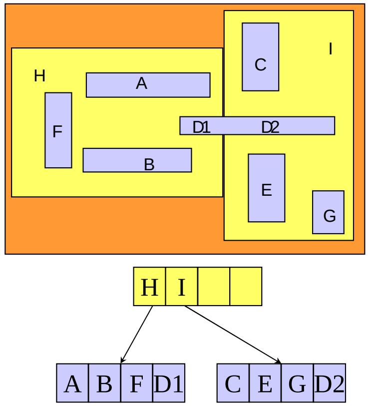
3.3.4 – Using R-tree

- Tree of rectangles (r-tree)
- Ameliorated trees (r*-tree)

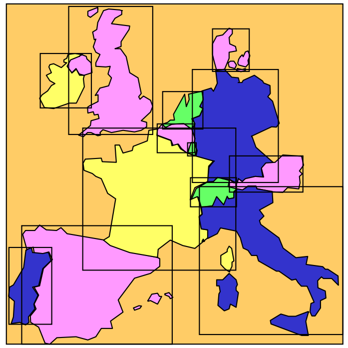
Example of an R-tree



Example of an R*-tree



Indexing European countries with Rectangles

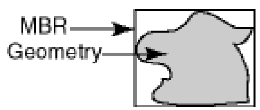


3.3.5 – Indexing in Oracle

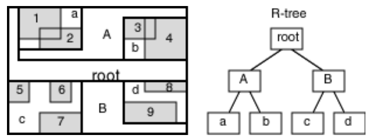
- Quadtree / R-tree

```
SQL>  create table <layername>_SDOINDEX
   2  (
   3    SDO_GID integer,
   4    SDO_CODE raw(255)
   5    );
```

R-tree

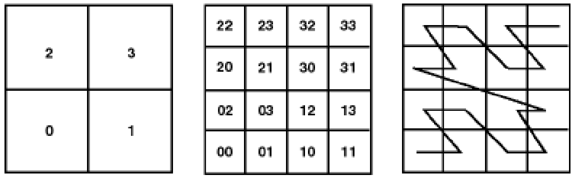


Minimum Bounding Rectangles



Indexing principle

Quadtree



Peano-key based Quadtree (Morton code)

HH codes

- HHCODEs (Helical Hyperspatial Codes)
- Peano key
- Longitude/latitude/altitude/time

Spatial Index Creation

```
--------------------------------------------------------------------
-- CREATE THE SPATIAL INDEX --
--------------------------------------------------------------------
CREATE INDEX cola_spatial_idx
ON cola_markets(shape)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
-- Preceding created an R-tree index.
-- Following line was for an earlier quadtree index:
--    PARAMETERS('SDO_LEVEL = 8');
```

Selecting one index

R-tree Indexing	Quadtree Indexing
The approximation of geometries cannot be fine-tuned. (Spatial uses the minimum bounding rectangles.	The approximation of geometries can be fine-tuned by setting the tiling level and number of tiles.
Index creation and tuning are easier.	Tuning is more complex, and setting the appropriate tuning parameter values can affect performance significantly.
Less storage is required.	More storage is required.
If your application workload includes nearest-neighbor queries (SDO_NN operator), R-tree indexes are faster.	If your application workload includes nearest-neighbor queries (SDO_NN operator), quadtree indexes are slower.
If there is heavy update activity to the spatial column, an R-tree index may not be a good choice.	Heavy update activity does not affect the performance of a quadtree index.
You can index up to four dimensions.	You can index only two dimensions.
An R-tree index is recommended for indexing geodetic data if SDO_WITHIN_DISTANCE queries will be used on it.	
An R-tree index is required for a whole-earth index.	

3.3.6 – Conclusions about spatial indexing

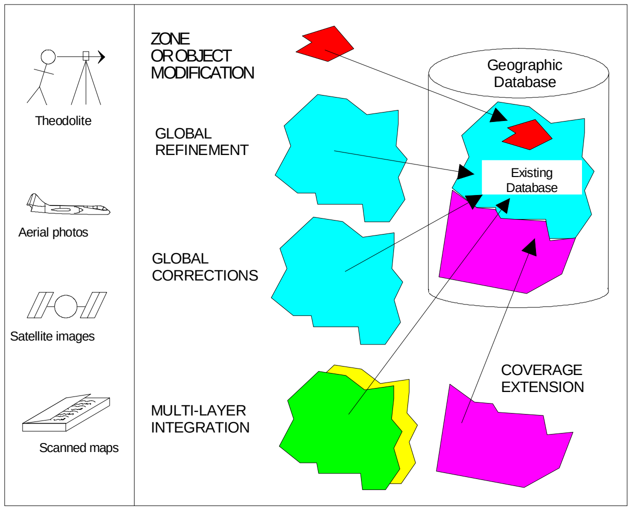
- Importance of access methods
- Data Organization
- Evolution to spatio-temporal
- Evolution to 3D
- Evolution to continuous phenomena
- Using in Oracle 11g (Locator or Spatial)

3.4 – Updating

- 3.4.1. Introduction
- 3.4.2. Alphanumeric Updating
- 3.4.3. Zonal Updating and Refinement
- 3.4.4. Global Updating
- 3.4.5. Mixing two layers
- 3.4.6. Coverage Extension
- 3.4.7. Conclusions

Importance of sources

- newly made measures with more accurate devices (theodolites, ..)
- vector and raster format for instance aerial photos or satellite images
- various data producers, using different bases or standards
- etc.



Toy Database

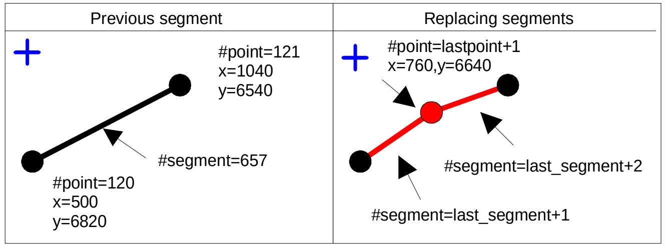
- PARCEL (#parcel,(#segment)*)
- SEGMENT (#segment, (#point)²)
- POINT (#point, x, y)
- LAST-KEYS (#parcel, #segment, #point)

3.4.2. Alphanumeric Updating

- Using languages such as SQL

```
• UPDATE POINT
  SET   x = 4567,    y = 7890
  WHERE #point = 2537;
```

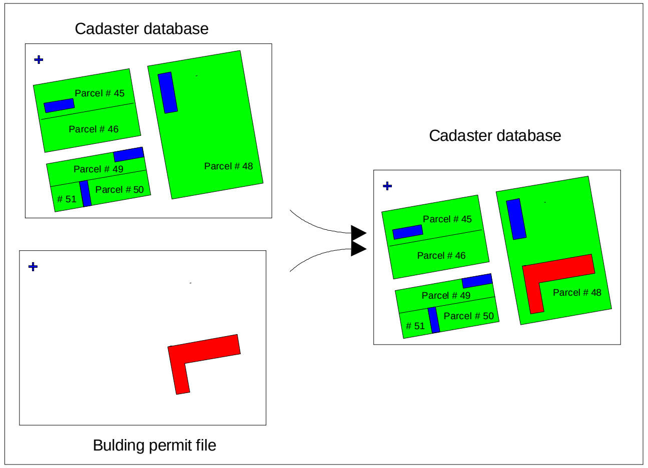
Introducing a new point into a segment



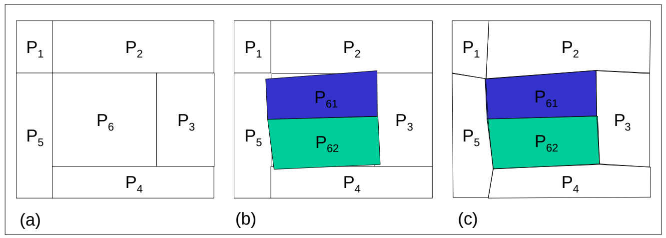

```
• DELETE FROM SEGMENT
WHERE #segment=657;
• INSERT INTO POINT VALUES
(LAST-KEYS.#point+1,760,6640);
• INSERT INTO SEGMENT    VALUES
(LAST-KEYS.#segment+1,120,
LAST-KEYS.#point+1);
• INSERT INTO SEGMENT VALUES
(LAST-KEYS.#segment+2,121,
LAST-KEYS.#point+1);
• UPDATE LAST-KEYS
   SET #segment=OLD.#segment+2,
   SET #point=OLD.#point+1;
• COMMIT;
```

3.4.3. Zonal Updating and Refinement

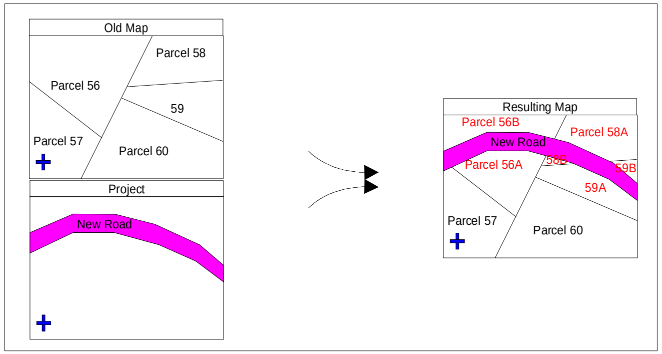
- Object integration without by-effects
- Updating with local modification without elastic transformation
- Updating with elastic transformation



Inserting new information



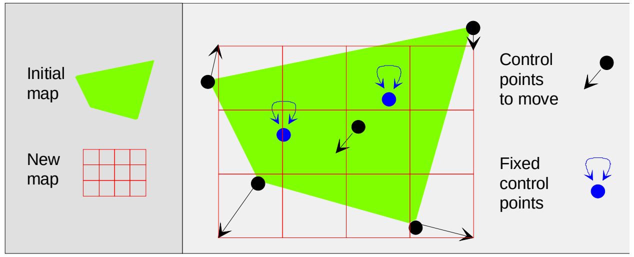
Updating with topological modifications



3.4.4. Global Updating

- conventional rubber-sheeting when a few number of control points are provided
- more sophisticated rubber-sheeting based on several points with constraints
- global updating based on aerial photos.

Rubber-sheeting



Formulae for rubber-sheeting

- Linear Rubber-sheeting

$$X = A \times x + B \times y + C$$

$$Y = D \times x + E \times y + F$$

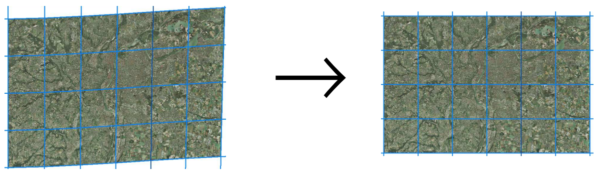
- Bilinear Rubber-sheeting

$$X = A \times xy + B \times x + C \times y + D$$

$$Y = E \times xy + F \times x + G \times y + H$$

- Old map coordinates: x, y
- New map coordinates : X, Y

Example with aerial photos



Before

After

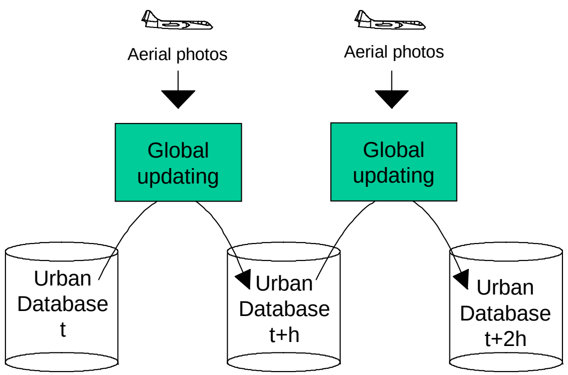
Force-fitting

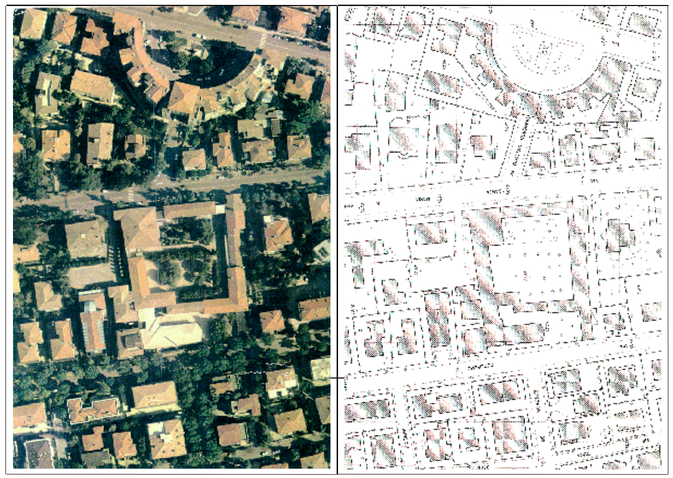
- force-fit of a point (coordinates),
- force-fit of the length of a segment,
- force-fit of an angle (especially right angles),
- force-replacing of a segment by a new polyline,
- etc..

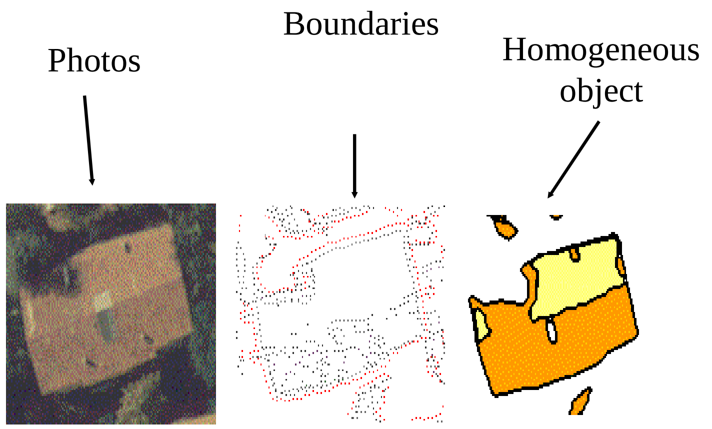
Global updating with aerial photos

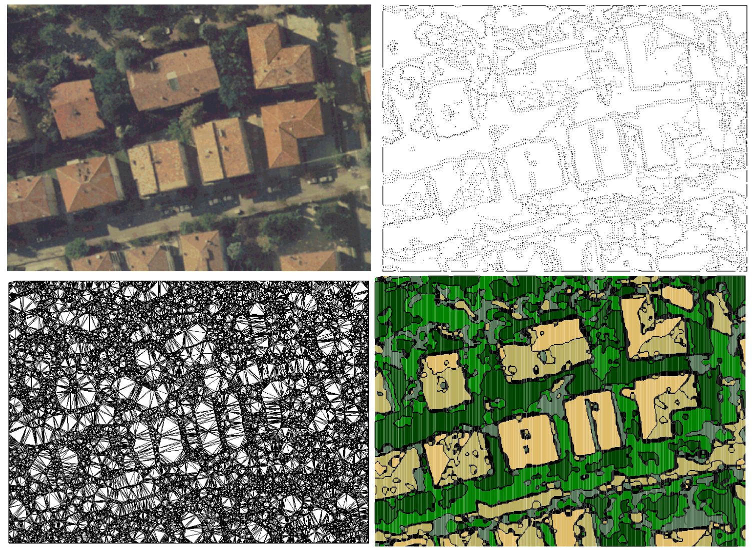
- Aerial photos taken every two years
- Updating the whole urban database
- Pixel = 8 cm × 8 cm
- A priori model-based reasoning

General principle



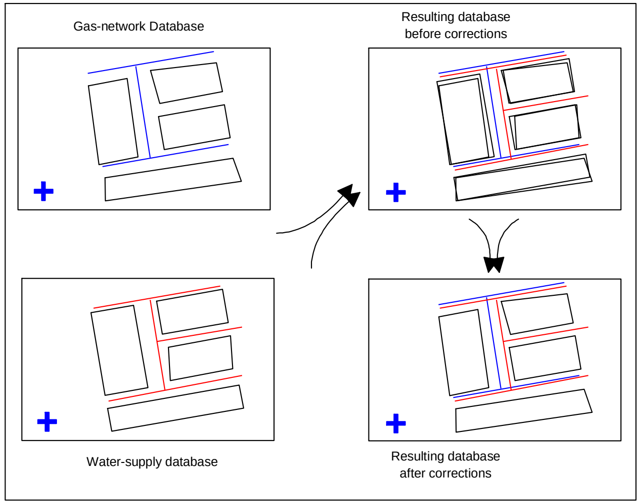






3.4.5. Mixing Several Layers

- For instances, streets, gas and water
- Alignment of coordinates
- Topological problems



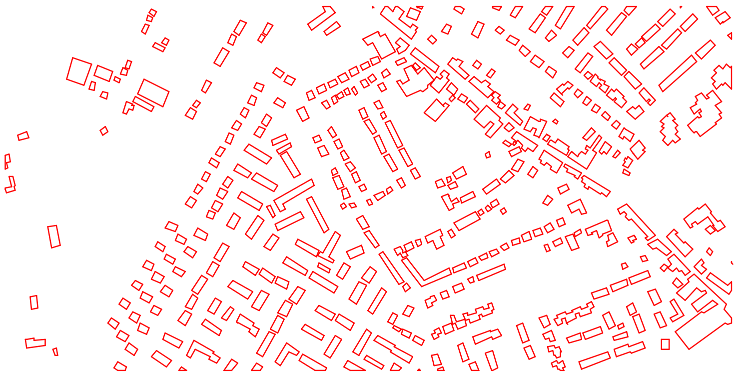
Mixing same information from different sources

- Examples: two buildings databases
- Dutch cadaster
- Problem: to reconcile the databases

Two Topographic Maps (1)

- Mid-scale (TOP10vector)
  - Scale 1 : 10,000
  - Photogrammetric mono-plotting
  - 4 year update cycle
  - ‘snap shot’ database

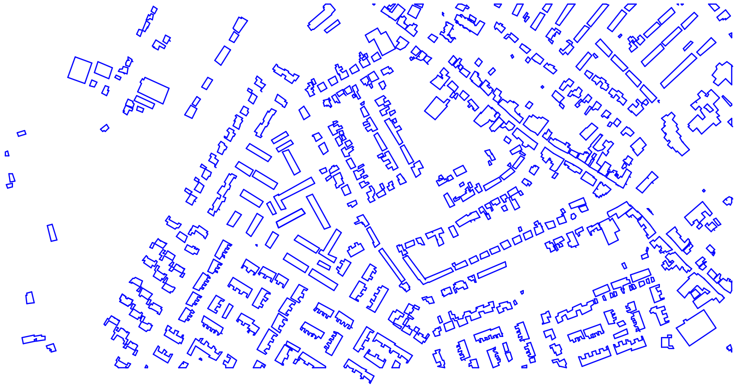
Mid-scale (Buildings)

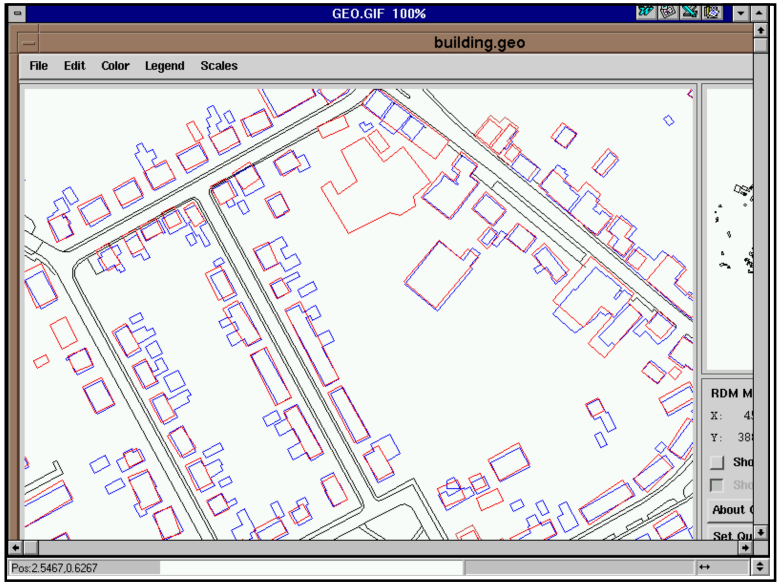


Two Topographic Maps (2)

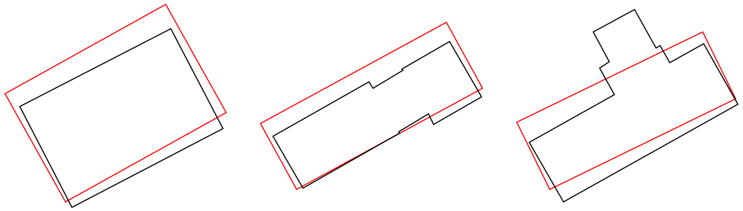
- Large-scale (GBKN)
  - Scale 1 : 1,000
  - Photogrammetric stereo-plotting
  - Updated continously
  - Contains history

Large-scale (Buildings)

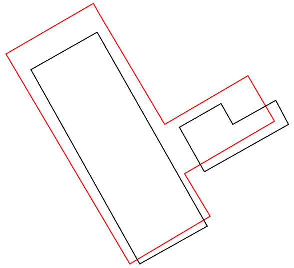




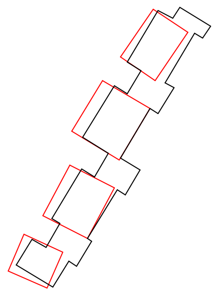
Correspondences: 1-to-1



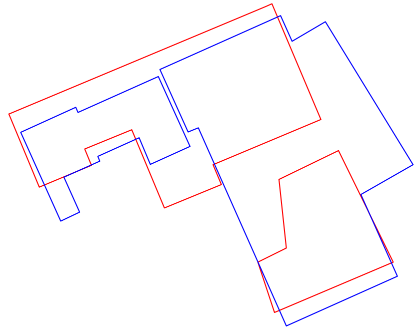
Correspondences: n-to-1



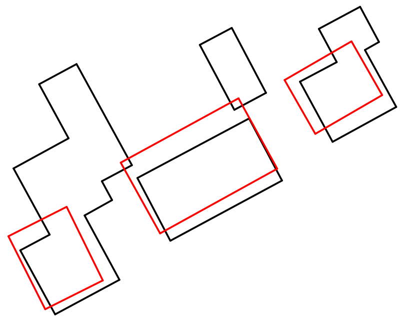
Correspondences: 1-to-n



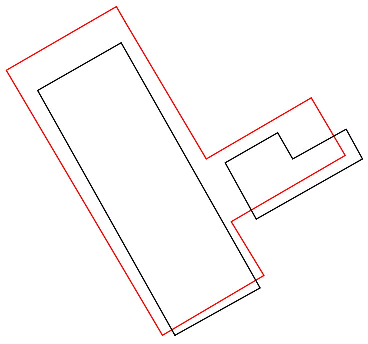
Correspondences: n-to-m



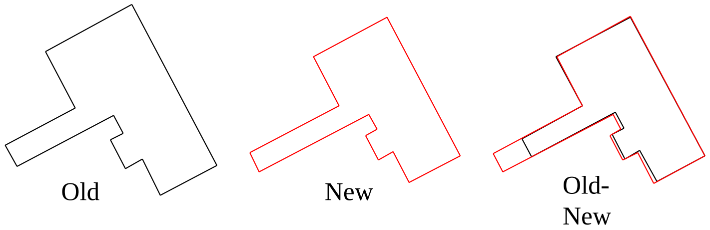
Finding Corresponding Objects



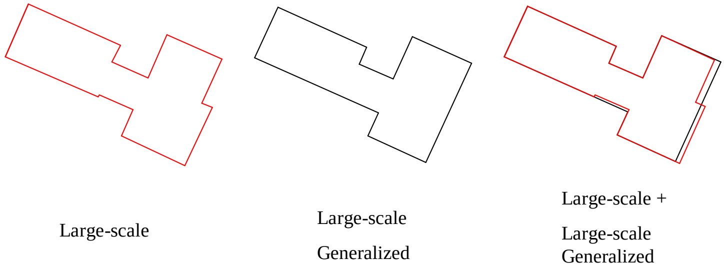
Aggregation & Filtering (1)



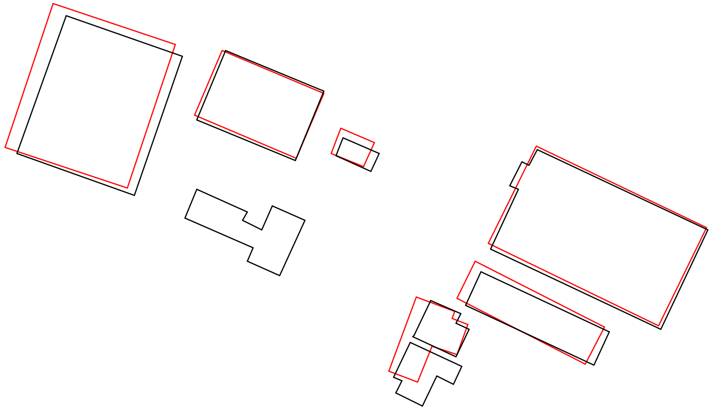
Aggregation & Filtering (2)



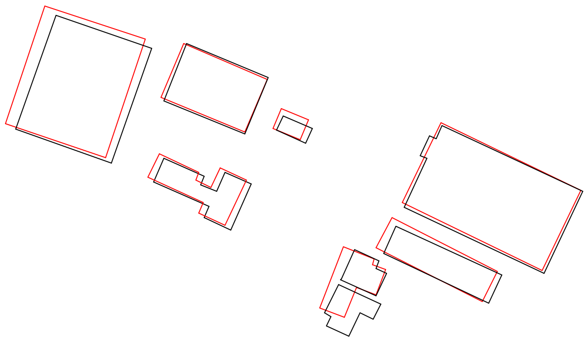
Cartographic Generalization



Adjustment (1)



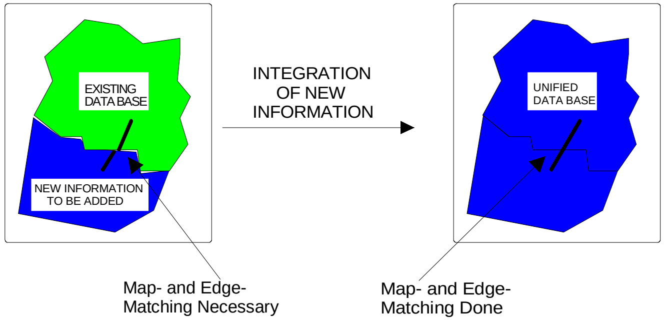
Adjustment (2)



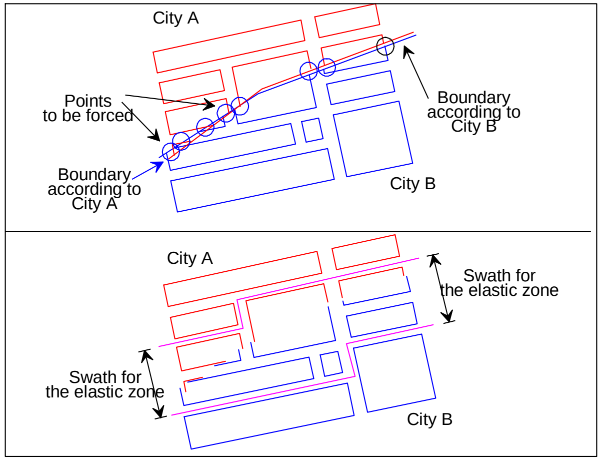
3.4.6. Coverage Extension

- Same layers or classes of objects
- Removing of overlaps
- Problems at the boundary

Integration of new information



Reorganization at the boundary



Rules

Rule 1: If the boundary segments of A are considered more accurate than those of B, then keep them (A) and force-fit the boundary segment of B;

Rule 2: If boundaries are different and both inaccurate then take a sort of mid-line and distort objects neighboring the boundary accordingly.

Examples of Constraints

- alignment of streets
- parallelism of kerbs or parcel limits
- rectangularity of some buildings.

3.4.7. Conclusions

- Importance of updating
- Importance of sources
- Importance of quality control
- Geometric accuracy
- Topological checking
- Necessity of nice visual interfaces
- Legislative aspects

3.5 – Conclusions

- Cartography
- Updating
- Querying