P02

EFFICIENT NEAREST NEIGHBOR QUERIES ON NON-POINT DATA

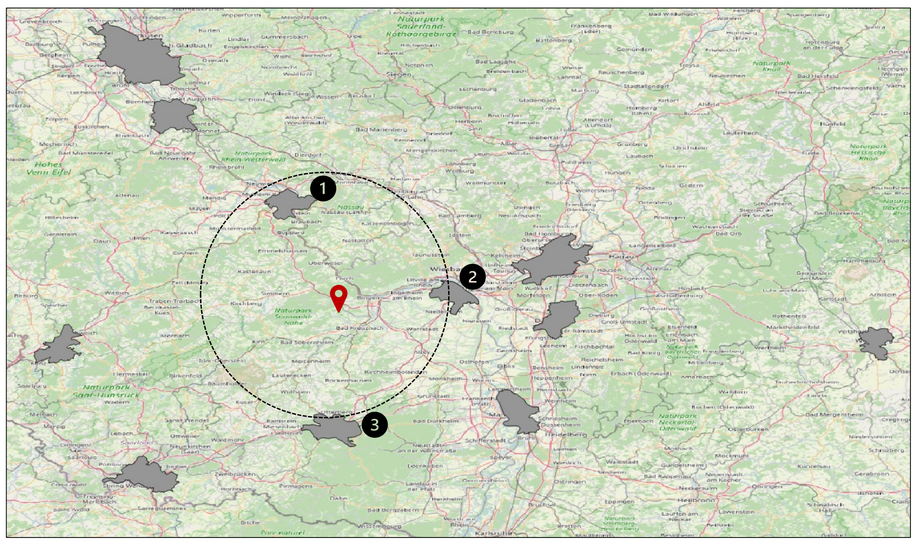
Achilleas Michalopoulos[1], Dimitrios Tsitsigkos[1], Panagiotis Bouros[2], Nikos Mamoulis[1] and Manolis Terrovitis[3]

[1]University of Ioannina, Greece [2]Johannes Gutenberg University Mainz, Germany [3]Athena RC, Greece

{amichalopoulos,dtsitsigkos,nikos}@cs.uoi.gr, bouros@uni-mainz.de, mter@imis.athena-innovation.gr

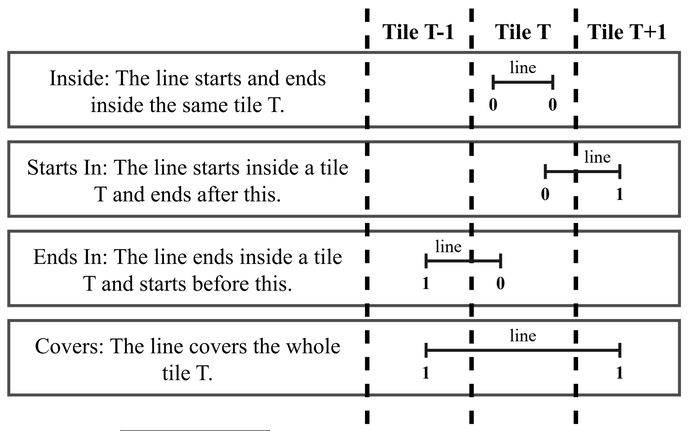
Motivation

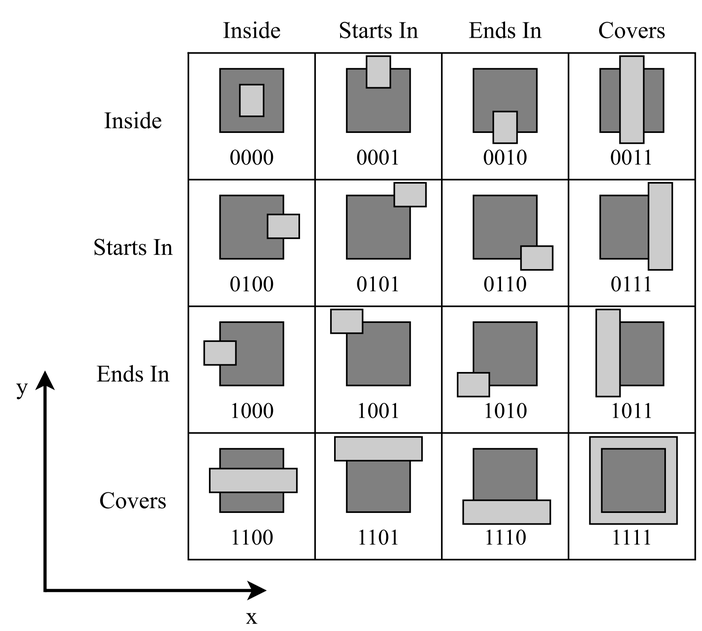
- *Nearest Neighbor* search -> *fundamental* operation
- GIS, data analytics, scientific applications etc.
- Find the 3 nearest cities to a query point *q*



Two-Layer Partitioning

- An SOP first-layer, e.g., a grid
- Decompose every first-layer partition into classes [1]
- Use the object *projections* to the axes
- *4 cases* per axis
- Overall, *16 classes* represented by a *4-bit* encoding

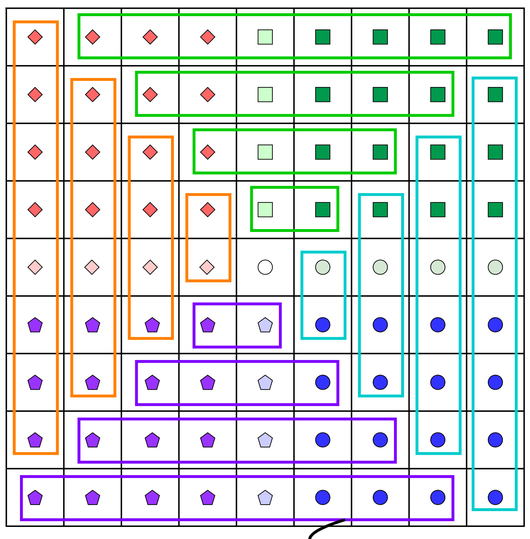




Query Processing

Tile examination [2]

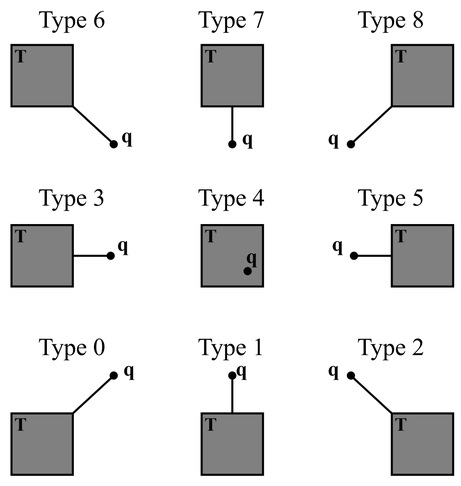
- *Splitting* the grid into conceptual *levels*
- *Level 0*: the tile *containing* query *q*
- ∀ $Level\ n \in [1, \mathrm{N}]$: $8n$ tiles (*4* tile-groups)
- Examining groups by distance to *q*



Tile categorization

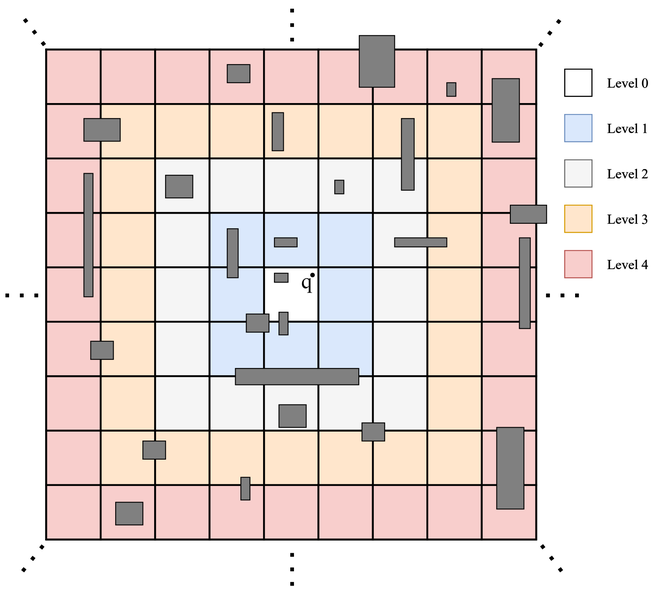
- *9 types* representing a *set of classes* each
- Process types based on their *direction* to q
- *Duplicate* avoidance, instead of elimination [3]
- Reducing comparisons

Tile Types	Sets of Classes
Type 0	{0000,0010,1000,1010}
Type 1	{0000,0010,0100,0110,1000,1010,1100, 1110}
Type 2	{0000,0010,0100,0110}
Type 3	{0000,0001,0010,0011,1000,1001,1010,1011}
Type 4	ALL CLASSES
Type 5	{0000,0001,0010,0011,0100,0101,0110,0111}
Type 6	{0000,0001,1000,1001}
Type 7	{0000,0001,0100,0101,1000,1001,1100,1101}
Type 8	{0000,0001,0100,0101}



Incremental search [4]

- *Min-heap Q*
- Storing tile-groups, tiles and objects
- *Max-heap H* (objects)



k-NN search

- *k* is given
- *Min-heap Q* (tile-groups and tiles)

Experiments

Setup

- All data in main memory
- *2-layer16* using a regular grid as SOP
- Focus on filtering phase
- 2 Tiger 2015 datasets

ROADS

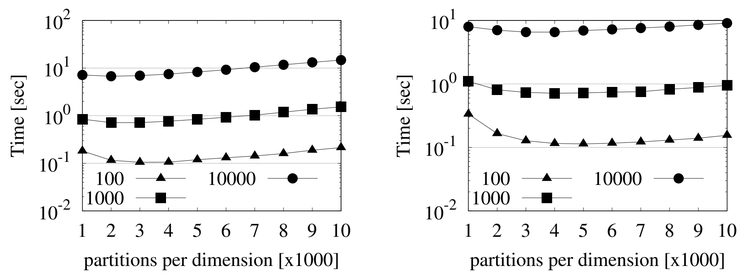
- Containing *19* millions *linestrings*
- Occupying *538* MB

EDGES

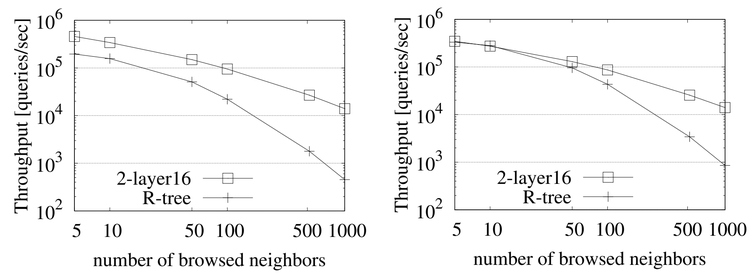
- Containing *69* millions *polygons*
- Occupying *1.6* GB

ROADS **EDGES** **ROADS** **EDGES**

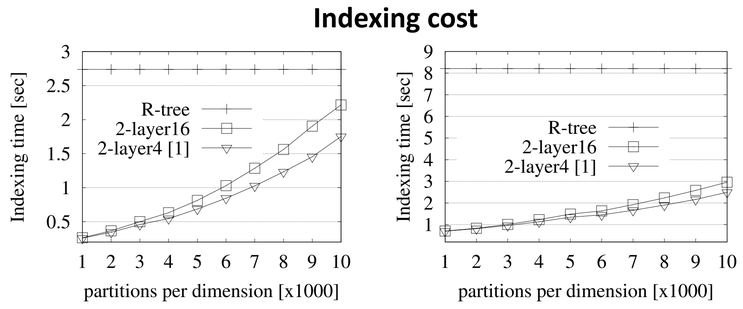
Tuning grid granularity



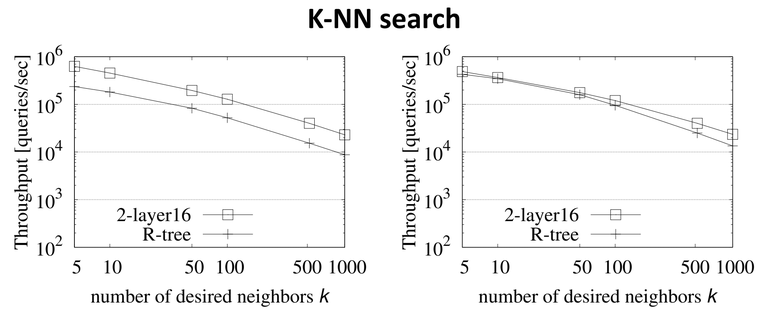
Incremental search



Indexing cost



K-NN search



References

[1]D. Tsitsigkos, K. Lampropoulos, P. Bouros, N. Mamoulis and M. Terrovitis. A Two-Layer Partitioning for Non-point Spatial Data. In *IEEE ICDE*, 2021.

[2] K. Mouratidis, M. Hadjieleftheriou and D. Papadias. Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring. In *ACM SIGMOD*, 2005.

[3] J.-P. Dittrich and B. Seeger. *Data Redundancy and Duplicate Detection in Spatial Join Processing*. In *IEEE ICDE*, 2000.

[4] G.R. Hjaltason and H.Samet. Distance Browsing in Spatial Databases. *ACM Trans. Database Syst.*, 1999.

31st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL), Hamburg, Germany, November 13-16, 2023