Efficient Nearest Neighbor Queries on Non-point Data

Poster Id: 02

Achilleas Michalopoulos[1] Dimitrios Tsitsigkos[1]

Panagiotis Bouros[2]

Nikos Mamoulis[1] Manolis Terrovitis[3]

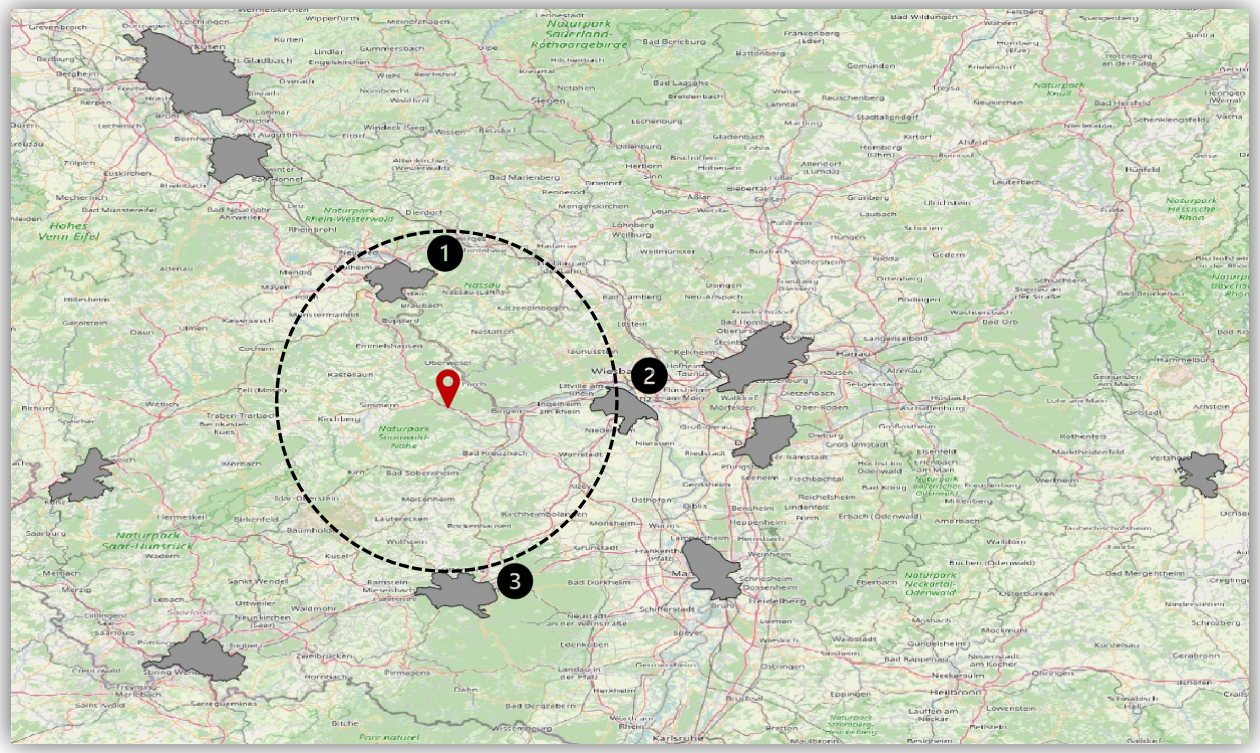
[1] University of Ioannina, Greece

[2] Johannes Gutenberg University Mainz, Germany

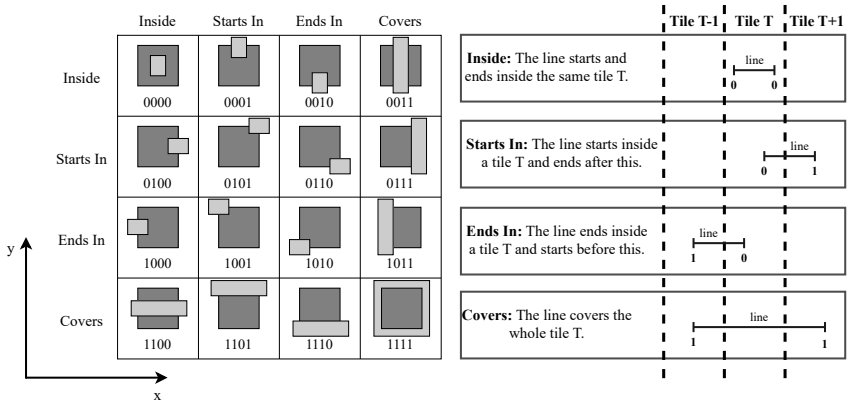
[3] Athena RC, Greece

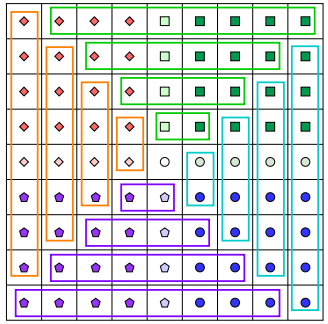
Nearest Neighbor Search

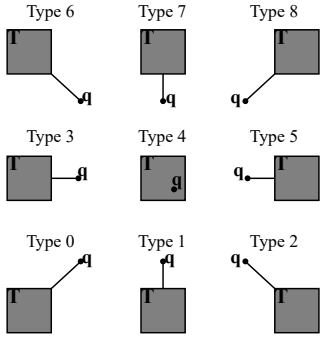
- Fundamental data operation
  - GIS, data analysis tasks, scientific applications etc.
  - Find the 3 nearest cities to a query point *q*



Using Two-Layer Partitioning







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}

- Two-Layer Partitioning [Tsitsigkos et al. 2021]
  - Suitable for SOP indices
- Decompose tiles into classes
  - On the projections of the objects
  - 4 cases per axis
  - Overall, 16 classes
- Incremental and *k*-NN search
  - Examine tiles around the one containing query *q* [Mouratidis et al. 2005]
  - Tile categorization
    - Duplicate avoidance
    - Reducing comparisons