Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases

InfoLab

USC

Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases

Citation 138

Ugur Demiryurek

Mohammad R. Kolahdouzan & Cyrus Shahabi
University of Southern California
Dept. of Computer Science
Los Angeles, CA 90089-0781

shahabi@usc.edu
http://infolab.usc.edu

Agenda

- Problem Definition
- Related Work
- Voronoi Diagrams
- Voronoi Based Network Nearest Neighbor (VN[3])
- Experiments
- Discussion

Problem Definition

kNN Problem:
Given a set of objects *P* and query point *q*, find the ***k*** objects in *P* that are closest to *q*

Related Work

Spatial Database

↓

kNN Query Processing

↓

Euclidean Space | Road Networks

- ✓ NN Query: Roussopoulos et al., SIMGOD 1995
- ✓ K-NN for moving query point: Zong et al., SSTD 2001
- ✓ Time-parameterized queries : Tao et al., SIMGOD 2002
- ✓ Continuous NN Search: Tao et al,. VLDB 2002

Object Based

Related Work

Spatial Database

↓

kNN Query Processing

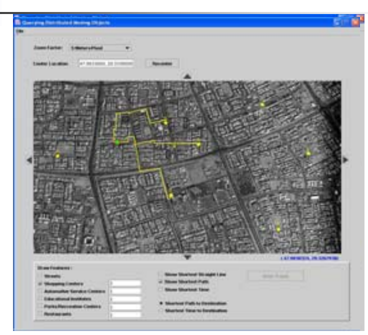
↓

Euclidean Space | Road Networks

- ✓ Sina-Continuous querying: Mokbel et al., SIGMOD 2004
- ✓ Sea-CNN queries: Xiong et al., ICDE 2005
- ✓ Monitoring kNN on moving objects: Yu et al., ICDE 2005
- ✓ Conceptual partitioning: Mouratidis et al., SIGMOD 2005

Space Based

Example 1:
Finding the 3 closest shopping centers

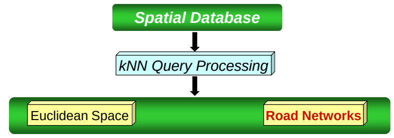


Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases

Example 2:
Finding the 3 closest restaurants to USC with Yahoo

7

Related Work

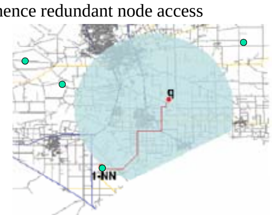


✓ Query processing in SNDB : Papadias et al., VLDB 2003
✓ Voronoi-based kNN in SNDB: Shahabi et al., VLDB 2004

8

Related Work

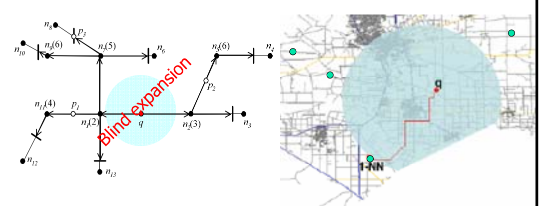
Query processing in SNDB : Papadias et al., VLDB 2003
Incremental Network Expansion (INE)
- Blind expansion hence redundant node access



9

Related Work

Query processing in SNDB : Papadias et al., VLDB 2003
- Incremental Network Expansion

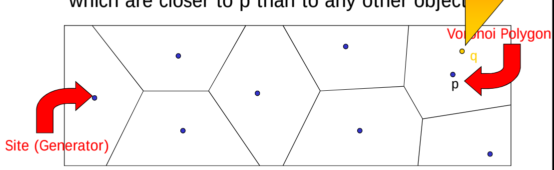


10

Preliminaries: Voronoi Diagram

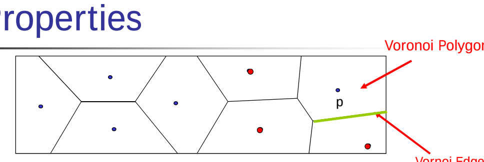
- Given set of sites (POI), a Voronoi diagram partitions the plane into disjoint Voronoi pol for each site
- The region including a site p inclu which are closer to p than to any other object

q is closer to the generator of the Voronoi Polygon than any other generator



11

Preliminaries: Voronoi Properties

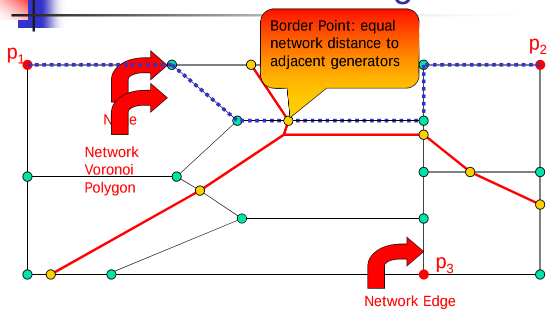


- Property 1: Voronoi diagram is unique
- Property 2: Voronoi edges are
  - shared by two generators
  - in equal distance to neighboring generators
- Property 3: Nearest generator of p is among the generators whose VP shares similar edges with p
- Property 4: Average number of Voronoi edges per VP is at most 6

12

Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases

Network Voronoi Diagram



13

VN^3 Approach

- Offline Index Generation
  1. Network Voronoi Construction
  2. Index Generation (R-tree)
  3. Distance Precomputation
- Online Query Processing
  1. Find 1st NN
  2. Find k NN-> Filter & Refine

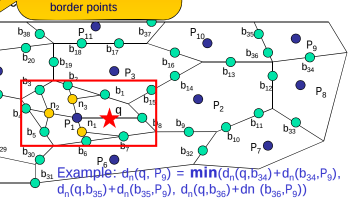
14

Offline Step

- Compute Netwo…)
- Index NVPs wit…
- Precompute th… and inter (within polygons) network dista…

1. Precompute the Shortest Path distance from each generator to its border point
2. Precompute the Shortest Path between border points

b_{34}	P_9	$d_n(b_{34}, P_9)$
b_{35}	P_9	$d_n(b_{35}, P_9)$
b_{36}	P_9	$d_n(b_{36}, P_9)$
…	…	……
b_1	b_2	$d_n(b_1, b_2)$
b_1	b_{15}	$d_n(b_1, b_{15})$
b_1	b_{13}	$d_n(b_1, b_{13})$
…	…	……



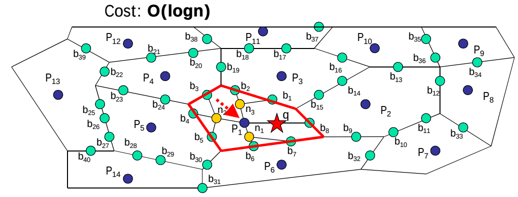
Example: $d_n(q, P_9) = \min(d_n(q,b_{34})+d_n(b_{34},P_9), d_n(q,b_{35})+d_n(b_{35},P_9), d_n(q,b_{36})+dn(b_{36},P_9))$

$d_n(q,b_{34}) = \min(d_n(q,b_1)+d_n(b_1,b_{34}), d_n(q,b_2)+d_n(b_2,b_{34}), .., d_n(q,b_8)+d_n(b_8,b_{34}))$

15

Online Step

- Find 1st NN
  - Search R-Tree to find the NVP that overlaps q
  - Report the generator of the NVP as the 1st NN;
    Cost: **O(logn)**

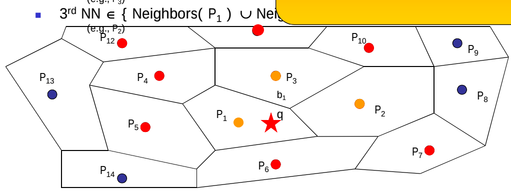


16

Online Step

- Find k NN → Filter &
- 1st NN = P_1
- 2nd NN ∈ { Neighbors(P_1) }
  (e.g., P_3)
- 3rd NN ∈ { Neighbors(P_1) ∪ Nei…
  (e.g., P_2)

NOOOO! it is to much computation if you have many candidates
GOOD NEWS! In theory average # of neighbors= O(5k+1)=O(k)



17

Performance of VN^3

- Data set:
  - Road network in Los Angeles (from NavTeq)
    - 110,000 streets, 79,800 intersections
    - Different points of interest:
      - restaurants, auto services, schools, parks, shopping centers, hospitals
- Measured:
  - Query response time and CPU
    - Comparison with INE [Papadias et al. (vldb03)]
  - Size of candidate set of VN^3 filter
    - Comparison with R-tree-based KNN [Seidl et al. (sigmod98) and Hjaltason et al. (tods99)]

18

VN^3 vs INE

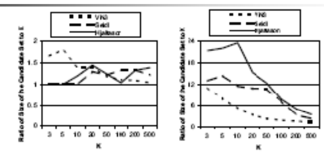
VN^3 finds the 1st NN using R-Tree
INE needs to expand the network around q

CPU Time:
INE uses a queue which is incrementally updated

		Query Processing Time (Sec)											
		K=1		K=5		K=10		K=25		K=50		K=100	
		VN^3	INE	VN^3	INE	VN^3	INE	VN^3	INE	VN^3	INE	VN^3	INE
Entities	Qty (density)	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk	(cpu) disk
Hospital	46 (0.0004)	(0) 0.018	(0.3) 12.4	(1.5) 6.5	(1.7) 78.3	(4.5) 14.0	(3.8) 165.1	(15.3) 35.1	(10.1) 430.2	- -	- -	- -	- -
Shopping Centers	173 (0.0016)	(0) 0.020	(0.09) 3.6	(0.45) 3.3	(0.5) 21.1	(1.3) 6.9	(1.1) 44.0	(3.4) 18.1	(3.1) 118.0	- -	- -	- -	- -
Parks	561 (0.0053)	(0) 0.021	(0.03) 1.4	(0.15) 1.5	(0.2) 8.2	(0.37) 2.8	(0.3) 15.3	(1.4) 6.4	(0.8) 36.4	(2.5) 13.3	(1.6) 71.1	- -	- -
Schools	1230 (0.0115)	(0) 0.027	(0.015) 0.6	(0.06) 0.75	(0.07) 3.5	(0.18) 1.46	(0.14) 6.6	(0.7) 3.9	(0.36) 15.6	(1.9) 7.5	(0.7) 32.2	- -	- -
Auto Services	2093 (0.0326)	(0) 0.030	(0.013) 0.57	(0.01) 0.65	(0.05) 2.43	(0.09) 1.4	(0.09) 4.3	(0.58) 2.95	(0.23) 10.0	(1.65) 6.68	(0.44) 19.4	(2.78) 13.1	(0.87) 38.00
Restaurants	2944 (0.0580)	(0) 0.032	(0.01) 0.49	(0.01) 0.57	(0.03) 1.34	(0.04) 1.48	(0.06) 2.7	(0.26) 2.8	(0.15) 6.8	(0.8) 6.1	(0.3) 13.3	(1.85) 12.8	(0.6) 26.0

19

VN^3: Performance of Filter Step



a. Minimum size of the candidate set

b. Maximum size of the candidate set

- VN^3's filter behaves independently from the density/distribution of points of interest
- As k increases ratio of number of candidate decreases
  - Some of the new neighbors are already explored!

20

Conclusion

- A novel approach for KNN queries in SNDB
- Based on:
  - Pre-calculating the solution space (first order Voronoi diagrams)
  - Pre-computing some distances (from borders to points inside each polygon)
- Outperforms the only other solution for SNDB
- Independent from object distribution
- Outperforms the solutions for Euclidean space in "filtering"

21

Discussion

- What if the edge weights are changing?
- What if the both query and data objects are moving?
- What if you like to find the nearest hotel and gas station at the same time?

22