Indexing Land Surface for Efficient kNN Query

Cyrus Shahabi, Lu-An Tang and Songhua Xing
InfoLab
University of Southern California
Los Angeles, CA 90089-0781
http://infolab.usc.edu

Outline

- Motivation
- Related Work
- Background
- Indexing Land Surface
- Query Processing
- Performance Evaluation
- Conclusion and Future Work

Motivation

Yosemite National Park

Which is the NEAREST campsite???

Problem

To find k Nearest Neighbor based on the Surface Distance.

Challenges

- Huge size of surface model
- Millions of terrain data for a region of 10km x 10km
- Costly Surface distance computation
- Tens of minutes on a modern PC for a terrain of 10,000
- No efficient surface index structure
- R-tree, Voronoi Diagram cannot apply directly.

Applications

- Tourist Applications
- Scientific Adventures
- Military Operations
- Geo-realistic Games
- Space Explorations
Outline

- Motivation
- Related Work
- Background
- Indexing Land Surface
- Query Processing
- Performance Evaluation
- Conclusion and Future Work

Related Work

- Spatial Database
- kNN Query Processing
- Euclidean Space
- Road Networks
- Surface
- Conventional kNN
- Reverse kNN
- Time-aware kNN
- Visible kNN

Related Work

- Spatial Database
- kNN Query Processing
- Euclidean Space
- Road Networks
- Surface
- Conventional kNN
- Reverse kNN
- Time-aware kNN
- Visible kNN

Related Work

- Spatial Database
- kNN Query Processing
- Euclidean Space
- Road Networks
- Surface
- Conventional kNN
- Reverse kNN
- Time-aware kNN
- Visible kNN

Related Work

- Spatial Database
- kNN Query Processing
- Euclidean Space
- Road Networks
- Surface
- Conventional kNN
- Reverse kNN
- Time-aware kNN
- Visible kNN

- Conventional kNN
- Reverse kNN
- Time-aware kNN
- Visible kNN

NN Query: Roussopoulos et al., SIMGOD 1995
Influences Set: Korn et al., SIMGOD 2000
FINCH Algorithm: Wu et al., VLDB 2008
Time-parameterized queries: Tao et al., SIMGOD 2002
Continuous NN Search: Tao et al., VLDB 2002
VkNN Query: Nutanong et al., DASFAA 2007
Related Work

- Spatial Database
- kNN Query Processing

- Euclidean Space
- Road Networks
- Surface

- Conventional kNN
- Reverse kNN
- Time-aware kNN
- Visible kNN

- Query Processing in SNDB: Papadias et al., VLDB 2003
- V-based kNN in SNDB: Shahabi et al., VLDB 2004
- KNN in Large Graphs: Yu et al., TKDE 2006
- CNN Monitoring in RN: Mouratidis et al., VLDB 2006

Outline

- Motivation
- Related Work
- Background
- Indexing Land Surface
- Query Processing
- Performance Evaluation
- Conclusion and Future Work

Background

- Triangular Irregular Network (TIN) Model
  - Digital Elevation Model (DEM)

- Computational Geometry: AGGREGATION, DENG, M., ARXUZ, M., ETAMAE, M., SCHMIDT, C. J.
**Background**

- Shortest Surface Path Computation
  - Chen-Han (CH) Algorithm: unfold all the faces of a polyhedron to one plane
  - Time Complexity: $O(n^{1.5})$, $n$ is the total number of the vertices on the surface.

---

**Outline**

- Motivation
- Related Work
- Background
- Indexing Land Surface
- Query Processing
- Performance Evaluation
- Conclusion and Future Work

---

**Indexing Land Surface**

- Intuition — Surface Voronoi Diagram
- Too Complex to Build

---

**Indexing Land Surface**

- Tight Surface Index
- $TC(p_1): \{ q : q \in \mathcal{F} \text{ and } D_v(p_1, q) < D_v(p_2, q) \}$
- For any query point $q \in TC(p_1)$, the nearest neighbor of $q$ in surface distance is $p_1$.
- $D_v(p_1, q) \leq D_v(p_2, q) \leq D_v(p_3, q)$ (where $p_3 = p_1 + p_2$)
Indexing Land Surface

Loose Surface Index

\[ LC(p) = \{q : q \in T \text{ and } D_T(p, q) < D_L(p, q) \} \]

Site \( p \) is guaranteed not to be the nearest neighbor of \( q \) if \( q \) is outside \( LC(p) \).

\[ \exists (p, q) \text{ such that } D_T(p, q) > D_L(p, q) \geq D_L(q, q) \]

Storage Scheme

R-Tree

Unlike the Voronoi diagram, tight/loose cell are concave polygons in most cases and much more irregular.

All cells are adjacent to each other, causing too much overlapping in R-Tree.

Index both on TC/LC

Solution: SIR-tree

An R-tree that is generated on site set \( P \)

Leaf node stores: sites inside the corresponding MBR, the pointer to the vertices list of the tight/loose cell and its neighbor list

SIR-Tree

An R-tree that is generated on site set \( P \)

Leaf node stores: sites inside the corresponding MBR, the pointer to the vertices list of the tight/loose cell and its neighbor list

SIR-Tree Insertion

Algorithm

1. locate \( p \) in \( I \), find out the loose cell \( LC(p) \) containing \( p \);
2. \( p \) neighbor \& \( LC(p) \)'s neighbor;
3. compute \( TC(p) \) and \( LC(p) \);
4. for each site \( p_i \) in \( p \) neighbor
5. update \( LC(p_i) \)'s edges according to \( TC(p) \);
6. update \( TC(p_i) \)'s edges according to \( LC(p) \);
7. insert \( p_i \) into \( I \);
8. return \( I \);

Outline

Motivation
Related Work
Background
Indexing Land Surface
Query Processing
Performance Evaluation
Conclusion and Future Work
Query Processing

Nearest Neighbor Query

If the query point falls into one tight cell, its nearest neighbor could be identified immediately without any surface distance computation.

Our experiment shows about 75% queries fall into one of these tight cells.

If the query point falls out of all tight cells, we need to unfold all loose cells that contain the query point to compute its surface distance to the candidates.

Search (i.e., number of candidates we need to compute distance to) is localized in loose cells.

Algorithm: Depth First Search

Current Node Stack: Nodelist

Please refer to Section 4.2 Property 4 of the paper for proof.
Query Processing

- Nearest Neighbor Query
- Algorithm: Depth First Search

Current Node: N4
Stack: Nodelist

Does TC(P1) or LC(P1) contain q?
YES, TC(P1)
Return p1 as NN

Does TC(P2) or LC(P2) contain q?
NO

Does TC(P3) or LC(P3) contain q?
NO

Does TC(P4) or LC(P4) contain q?
NO

Does TC(P5) or LC(P5) contain q?
NO
Query Processing

Nearest Neighbor Query
Algorithm: Depth First Search

Current Node Stack: Nodelist

p9

Current Node Stack: Nodelist

query

Yes, LC(p3)

Current Node Stack: Nodelist

Does TC(p3) or LC(p3) contain q?

Current Node Stack: Nodelist

p1

p2

Candidate Set C

Does the LC of any p3's neighbor contain q?
Query Processing

- Nearest Neighbor Query
  - Algorithm: Depth First Search

- k Nearest Neighbor Query
  - Property 4
    The next nearest site is the generator of one of the neighbors of the NNs found so far.

Therefore, the shortest surface path from \( q \) to the \( k \)-th NN \( p_k \) will lie in the area of
\[ LC(G) \cup LC(p_a) = LC(p_1) \cup LC(p_2) \cup ... \cup LC(p_k). \]

Query Processing

- More about Query Processing
  - Surface Index R-Tree (SIR-tree)
  - How an R-tree is built on TSI and LSI?
  - SIR-tree insertion
  - Please refer to Section 4.4 in the paper.

- NN Query Algorithm
  - Please refer to Section 5.1 Algorithm 1 in the paper.

- kNN Query Processing
  - Property of next nearest neighbor
  - Incremental algorithm for kNN Query
  - Please refer to Section 5.2 in the paper.

Outline

- Motivation
- Related Work
- Background
- Indexing Land Surface
- Query Processing
  - Performance Evaluation
- Conclusion and Future Work

Performance Evaluation

- Dataset
  - Eagle Peak (EP) at Wyoming State, USA
    - 10.7km × 14km, 1.4M sampled points.
  - Bearhead (BH) at Washington State, USA
    - Similar size as above, 1.3M sampled points.
  - Uniformly distributed Point of Interest

http://data.geocomm.com/
Performance Evaluation

Competing Approaches

- Surface Index (SI)
  - Exact and quick answer
- Range Ranking (RR)
  - Approximate and quick answer
- Chen Han Algorithm (CH)
  - Exact and slow answer

Query Efficiency, I/O cost vs. Value of k

The difference in improvement of SI over CH increases for larger k.

Accuracy vs. Value of k

- The accuracy of RR drops dramatically when the value of k increases.
- The accuracy of SI stays at 100%.

Outline

- Motivation
- Related Work
- Background
- Indexing Land Surface
- Query Processing
- Performance Evaluation
- Conclusion and Future Work

Conclusion and Future Work

- Conclusion
  - We extend the traditional kNN Query to the space constrained with the third dimension.
  - We construct two complementary indexing schemes, namely Tight Surface Index (TSI) and Loose Surface Index (LSI) to reduce the invocation of the costly surface distance computation.
  - SI significantly outperforms its competitors in accuracy and efficiency.

- Future Work
  - Further evaluate its performance with synthetic datasets.
  - Study variations of skNN such as the continuous skNN query, dynamic skNN query and visible skNN query.

Thanks!

Email: Songhua Xing
sxing@usc.edu