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

- Why surface?
- The availability of the data (imagery, elevation) of Earth surface at very high resolution
- The quick geo-realistic rendering of terrain surface on the computer display (e.g., Google Earth™)
- The prevalence of GPS equipped sensors
- Various Applications

Various Applications:
- June 2009: the most complete terrain map of the Earth’s surface has been published through the collaboration between NASA (USA) and the ASTER (Japan).
Motivation

- Why is this problem challenging?
  - Huge size of surface model
  - Millions of *triangles* within a region of 100 km

Triangular Meshes
Motivation

- Why is this problem challenging?
  - Huge size of surface model
    - Millions of *triangles* within a region of 10km×10km
  - Costly surface distance computation
    - Tens of minutes on a modern PC for a terrain of 10,000 triangular meshes

*Chen-Han Algorithm: \(O(n^2)\)*

*Shortest paths on a polyhedron:* CHEN, J., HAN, Y., Computational Geometry 1990
Motivation

Why is this problem challenging?

- Huge size of surface model
  - Millions of triangles within a region of 100 km²
- Costly surface distance computation
  - Tens of minutes on a modern PC for a terrain of 10,000 triangular meshes

Chen-Han Algorithm: $O(n^2)$

* Shortest paths on a polyhedron: CHEN, J., HAN, Y., Computational Geometry 1990
Motivation

Why is this problem challenging?

- Huge size of surface model
  - Millions of triangles within a region of 100 km²
- Costly surface distance computation
  - Tens of minutes on a modern PC for a terrain of 10,000 triangular meshes

Chen-Han Algorithm: $O(n^2)$

* Shortest paths on a polyhedron: CHEN, J., HAN, Y., Computational Geometry 1990
Motivation

Distance Metrics

- Euclidean Distance $D_E(p,q)$
- Network Distance $D_N(p,q)$
- Surface Distance $D_S(p,q)$

$D_E(p,q) \leq D_S(p,q) \leq D_N(p,q)$

Other Approximations?

Restricted on Edges

NOT Restricted on Edges
Projective Surface Paths are not necessarily better than the network paths to approximate the actual surface paths.
Outline

- Motivation

- Related Work

- 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

✓ NN Query: Roussopoulos et al., SIMGOD 1995
Related Work

- 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
Related Work

- 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
Related Work

**Spatial Database**

- **kNN Query Processing**
  - **Euclidean Space**
  - **Road Networks**
  - **Surface**

- **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
☑ RNN in Large Graphs: Yiu et al., TKDE 2006
☑ CNN Monitoring in RN: Mouratidis et al., VLDB 2006
Related Work

Spatial Database

kNN Query Processing

- Euclidean Space
- Road Networks
- Surface

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

✓ SkNN Query: Deng et al., ICDE 2006, VLDB J. 2008
Related Work

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

SkNN Query: Deng et al., ICDE 2006, VLDB J. 2008
- Not an incremental approach
- Not an exact approach
Outline

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

- Intuition – Surface Voronoi Diagram

Too Complex to Build

Voronoi Diagram
For any query point $q \in TC(pi)$, the nearest neighbor of $q$ in surface distance is $pi$.

For any query point $q \in TC(pi)$, the nearest neighbor of $q$ in surface distance is $pi$.

$$TC(pi)=\{q: q \in T \text{ and } DN(pi, q) < DE(p_j, q) \ (\forall p_j \in P, p_j \neq pi)\}$$

$$DS(pi, q) \leq DN(pi, q) < DE(p_j, q) \leq DS(p_j, q) \ (\forall p_j \in P, p_j \neq pi)$$
Indexing Land Surface

- **Loose Surface Index**

\[ LC(p_i) = \{ q : q \in T \text{ and } DE(p_i, q) < DN(p_j, q) \ (\forall p_j \in P, p_j \neq p_i) \} \]

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

- \( \exists p_j \in P \ (p_j \neq p_i) \) such that \( DS(p_i, q) \geq DE(p_i, q) > DN(p_j, q) \geq DS(p_j, q) \)
Indexing Land Surface

- **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**
    - Same as VOR-Tree

* For the purpose of clarity, textures on terrain are removed.
Outline

- Motivation
- Related Work
- 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.
Query Processing

- **Nearest Neighbor Query**

- 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.
Query Processing

Nearest Neighbor Query

- If $p_i$ is the nearest neighbor of $q$, then the shortest surface path from $q$ to $p_i$ is inside the loose cell $LC(p_i)$.

- Computation (i.e., unfolding: invocation of CH algorithm) is **localized** in loose cells.

* Please refer to Section 4.2 Property 4 of the paper for proof.
Outline

- Motivation
- Related Work
- 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


![Eagle Peak (EP) and Bearhead (BH)](image-url)
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
Performance Evaluation

- **Query Efficiency, I/O cost vs. Value of k**
- The difference in improvement of SI over CH increases for larger k.

![Graphs showing query efficiency and I/O cost vs. k](image)
Performance Evaluation

**Accuracy vs. Value of k**

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

![Accuracy vs. Value of k](image-url)

(a) Query Accuracy on EP

(b) Query Accuracy on BH
Outline

- Motivation
- Related Work
- 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.
References

Continuous Monitoring of Nearest Neighbors on Land Surface

Instructor: Cyrus Shahabi
Preliminaries

- Continuous nearest neighbor monitoring in road networks (Mouratidis et al, VLDB 06)
- Shortest Path Spanning Tree (Dijkstra Tree)
Observation 1: SE Tree is fat and short.
Observation 2: These shortest surface paths rarely share common edges.
Observation 3: These shortest surface paths do not cross each other.
Preliminaries

- Continuous Monitoring ($k = 3$)
  - More incoming objects
    - Original Result Boundary contains more than $k$ objects
    - Identify the $k$th object and shrink the result boundary
Preliminaries

- Continuous Monitoring \((k = 3)\)

**DRAWBACKS**
- Expansion is slow;
- Search Area may be large;
- SE Tree is fat and short.

- More outgoing objects
  - Original Result Boundary contains less than \(k\) objects
  - Compute the Expansion Boundary
  - Expand SE tree to Expansion boundary and identify the \(k\)th object inside the search area
Surface Index based Approach

Intuition

Motivated by *Partitioning* Shortest Paths on road networks

- Scalable Network Distance Browsing in Spatial Databases: Samet et al., SIGMOD 2008 (best paper)

Is there a way to partition paths on surface?

The shortest path from X to any vertex of one region must be inside that region.
Surface Index based Approach

- **Intuition**
  - Motivated by *Partitioning* Shortest Paths on road networks
    - Scalable Network Distance Browsing in Spatial Databases: Samet et al., SIGMOD 2008 (best paper award)

- Too many partitions

V: a vertex on the surface
Surface Index based Approach

- Surface Shortest Path Container
  - Built based on the Observations that shortest paths do not cross each other and hardly share common edges.

![Diagram showing edges opposite to the source point](image)
Surface Index based Approach

- Angular Surface Index (ASI)
- Hierarchically index these containers

(a) ASI

(b) an abstraction of ASI

ADVANTAGES
- Size: Much Smaller
- Regularity: Well Balanced
- Efficiency: Tall and Thin
Surface Index based Approach

- **Query Processing**
  - The core is the same as using SE Tree
  - Search is localized in Containers only

![Diagram showing search regions and points](image-url)
Surface Index based Approach

- **More Details**
  - Definitions/Properties of Containers, Equidistance Line and ASI-tree
    - Please refer to Section 5.1, 5.2 and 5.3 in the paper.
  - Constructions of Containers
    - Please refer to Section 5.1.2 in the paper.
- **Algorithms for Query Processing**
  - Please refer to Section 5.4 in the paper.
Conclusion and Future Work

- **Contribution**
  - First study the Continuous $k$NN query on Surface;
  - Show the concept of expansion tree for land surface does not work as SE-tree suffers from intrinsic defects: it is fat and short;
  - Propose a superior approach that partitions SE-Tree into hierarchical chunks of pre-computed surface distances which overcome the above deficiency.
  - Experimentally verify the applicability of proposed methods.

- **Future work**
  - Study the Continuous $k$NN query problem for arbitrary moving query point.
References

• Songhua Xing, Cyrus Shahabi, Bei Pan, Continuous Monitoring of Nearest Neighbors on Land Surface, VLDB 2009.