南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

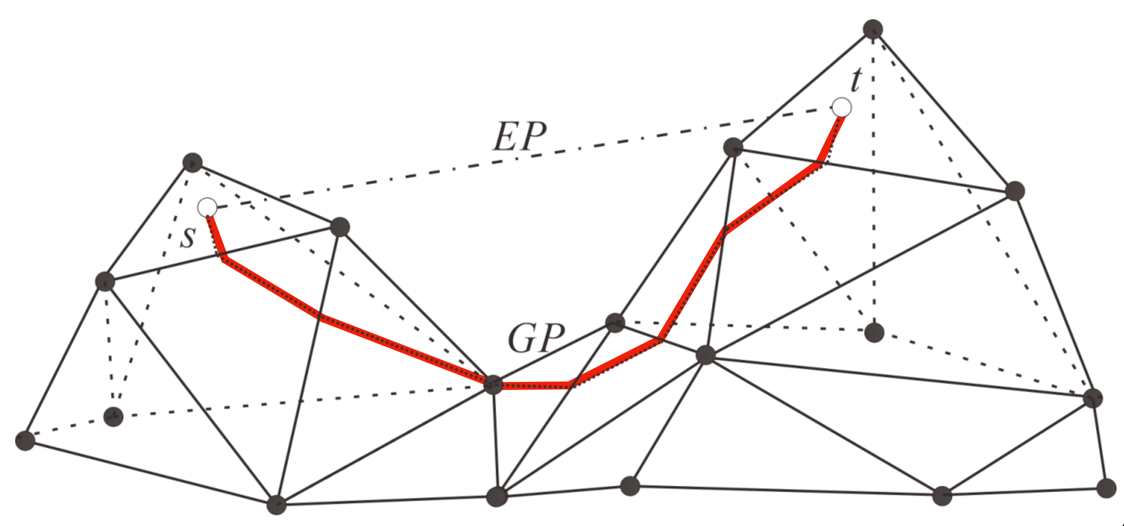
香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

On Efficient Indexing for Distance Queries between Arbitrary Points on Terrain Surface

Bo Huang (SUSTech), Victor Junqiu Wei (PolyU), Raymond Chi-Wing Wong (HKUST), and Bo Tang (SUSTech)

Overview

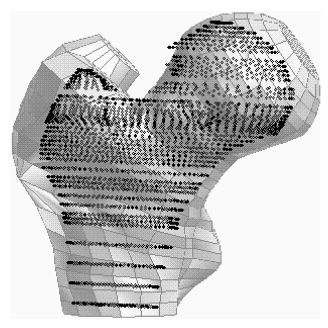
- The distance computation on the terrain surface is a fundamental and important problem that is widely applied in geographical information systems and 3D modeling.
- We propose an *indexing structure*, namely *Efficient Arbitrary Point-to-Arbitrary Point Distance Oracle* (*EAR-Oracle*), with theoretical guarantee on the accuracy, oracle building time, oracle size and query time. Our experiments demonstrate that our oracle outperforms the state-of-the-art algorithms *by orders of magnitude*.

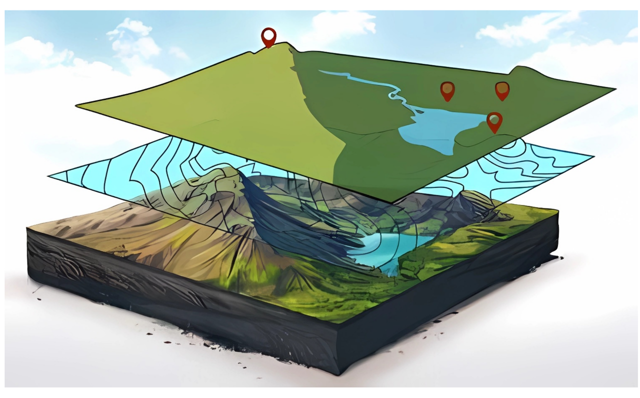


Preliminary

- *Terrain surface*: Consists of *Vertices*, *Edges* and *Faces* in 3D.
- *Geodesic distance*: *Shortest distance* on the terrain surface.
- Arbitrary point-to-arbitrary point (*A2A*) *queries*: Geodesic *distance queries* between *arbitrary* surface points.

Application





- Geographic Information Systems (GIS):
  - Compute travel cost;
  - Study animal travel patterns.
- Spatial Data Mining:
  - Check spatial co-location patterns;
  - Clustering objects on terrain surfaces.
- Scientific 3D Modeling:
  - Analyze key features.

Research Problem

- Given two *arbitrary* surface points s and t, find the *approximate* geodesic distance between s and t with *theoretical guarantee*.

Existing Study

- *On-the-fly* algorithms:
  - Fixed Scheme (*FS*) [*Algorithmica' 2001*]
  - Unfixed Scheme (*US*) [*J. ACM' 2005*]
  - K-Algorithm (*K-Algo*) [*VLDB' 2015*]

❖*Deficiency*: Queries are *processed online* (large query latency).

- *Index-based* algorithms:
  - Steiner-Point Oracle (*SP-Oracle*) [*ESA' 2011*]
  - Space-Efficient Oracle (*SE-Oracle*) [*SIGMOD' 2017*]

❖*Deficiency*: *Index* too many *auxiliary points* (large index cost).

Contribution

- We propose an *index-based* algorithm for *A2A queries*, Efficient Arbitrary point-to-arbitrary point distance oracle (*EAR-Oracle*):
  - *No assumptions* on query points;
  - Outperforms state-of-the-art index-based algorithm by *2 orders of magnitude* in terms of *building time* and *space consumption*;
  - Outperforms the fastest on-the-fly algorithm by *more than 1 order of magnitude* in terms of *query time*;
- We provide thorough *theoretical analysis*:
  - Building time, space consumption, query time and relative error.
- We conduct extensive *experimental studies*:
  - On several *real datasets* with different scales;
  - On *factors* influencing the performance of *EAR-Oracle*.

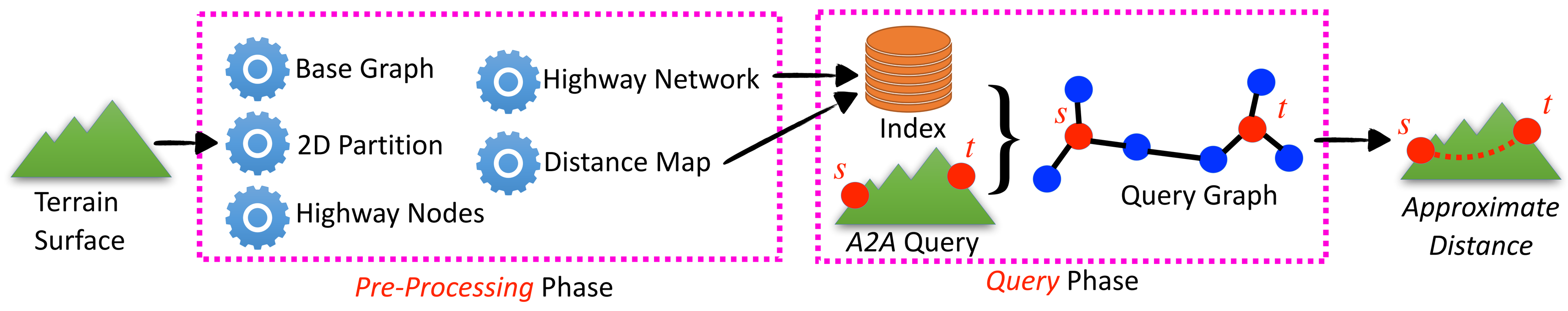
Theoretical Analysis

- Let N be the number of terrain faces and ϵ be the user-defined error bound:
  - Building time: *linearithmic* to *N*;
  - Space consumption: *linear* to *N*;
  - Query time: *linearithmic* to the amount of highway nodes;
  - Distance relative error: *close to* ϵ in practice.

A *subset* of terrain vertices whose cardinality is much less than N.

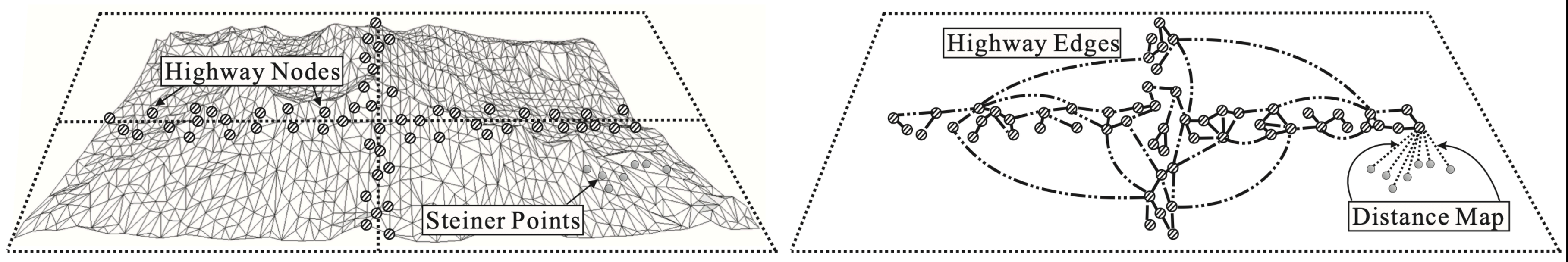
[*Algorithmica' 2001*]. M. Lanthier, A. Maheshwari, and J-R Sack. Approximating shortest paths on weighted polyhedral surfaces.
[*J. ACM' 2005*]. L. Aleksandrov, A. Maheshwari, and J-R Sack. Determining approximate shortest paths on weighted polyhedral surfaces.
[*VLDB' 2015*]. M. Kaul, R. C.-W. Wong, and C. S. Jensen. New lower and upper bounds for shortest distance queries on terrains.
[*ESA' 2011*]. H. N. Djidjev and C. Sommer. Approximate distance queries for weighted polyhedral surfaces.
[*SIGMOD' 2017*]. V. J. Wei, R. C.-W. Wong, C. Long, and D. M. Mount. Distance oracle on terrain surfaces.

EAR-Oracle Flow Chart



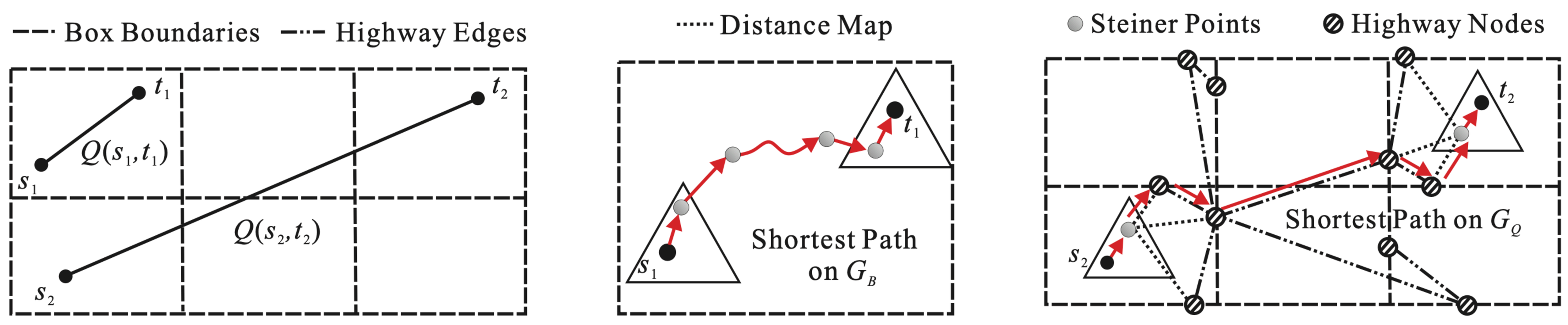
Proposed Method: *EAR-Oracle*

- *Pre-processing* Phase:
  - Construct *base graph* (G_B) for distance metric approximation:
    - Place m Steiner points *uniformly* on each angle bisector.
  - *Partition* terrain surface into *boxes* on x-y plane:
    - When the query points are *close*, they have *spatial locality*;
    - When the query points are *distant*, their *geodesic path* will go through *boundaries of some boxes*.
  - Select *highway nodes* based on terrain vertices:
    - A *subset* of the terrain vertices *near* box boundaries.
  - Construct *highway network* based on highway nodes:
    - Generate edges between highway nodes according to *geometric property* (avoid all-pair distances computation).
  - Build *distance map* based on highway nodes and Steiner points:
    - For *each box*, index the distance between each *highway node* on its boundaries and *Steiner points* on the faces inside it.



(a) Highway Nodes and Steiner Points (b) Highway Edges and Distance Map

- *Query* Phase:
  - *Inner-box* queries (Query points in the *same* box):
    - Adopt Dijkstra's algorithm on *base graph* G_B.
  - *Inter-box* queries (Query points in *different* boxes):
    - Construct a *query graph* G_Q by adding edges (from the *distance map*) to the highway network;
    - Perform Dijkstra's algorithm on *query graph* G_Q.



(a) Shortest Distance Queries (b) Inner-Box Query (c) Inter-Box Query

Experimental Study

- We test both *on-the-fly* and *index-based* algorithms.
- *EAR-Oracle* has the *best* overall performance.

Dataset	No. of Faces	Region Covered
HorseMountain (HM)	1,488	15 km^2
BigMountain (BM)	2,772	29 km^2
HeadLightMountain (HL)	4,771	49 km^2
RobinsonMountain (RM)	7,200	71 km^2
GunnisonForest (GF)	199,998	10,038 km^2
LaramieMountain (LM)	199,996	12,400 km^2
BearHead (BH)	292,914	140 km^2
EaglePeak (EP)	325,713	150 km^2