Geodetic Point-In-Polygon Query Processing in Oracle Spatial

Ying Hu, Siva Ravada, Richard Anderson

Oracle New England Development Center

ORACLE

Talk Outline

- The Geodetic PIP Problem
- Related Work
- Our Approach
- Experiments
- Conclusions

Talk Outline

- The Geodetic PIP Problem
- Related Work
- Our Approach
- Experiments
- Conclusions

The Problem (I)

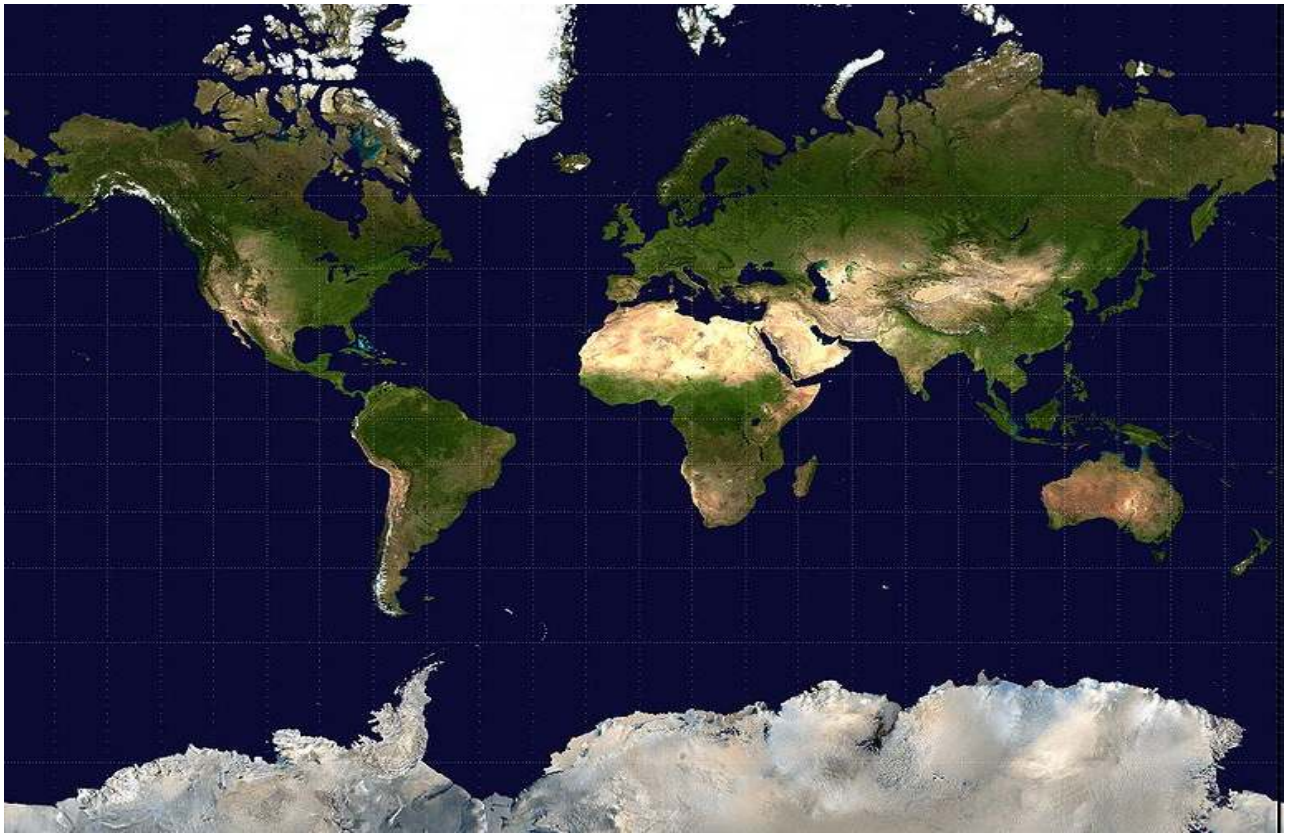
- PIP is very important
  - fundamental problem in Computational Geometry
  - used in many geospatial applications
- Location databases are growing
  - GPS is everywhere $\Rightarrow$ lots of points w/ (long, lat)
- Complex query polygons
  - e.g. Norway's extensive coastline, including many long fjords, small islands
- The focus: given any arbitrary query polygon, return all points inside this polygon from a big point database

The Problem (II)

- Additional Challenges for geodetic or geography data
  - Earth is not flat, and is very close to sphere
  - The computations on geodetic data are more expensive than those on Cartesian or projected data
- Many users use geodetic data natively
  - Although some users can transform geodetic data to projected data, projection details can be difficult.

The Problem (III)

- How to deal with poles, international date line in the Mercator projection

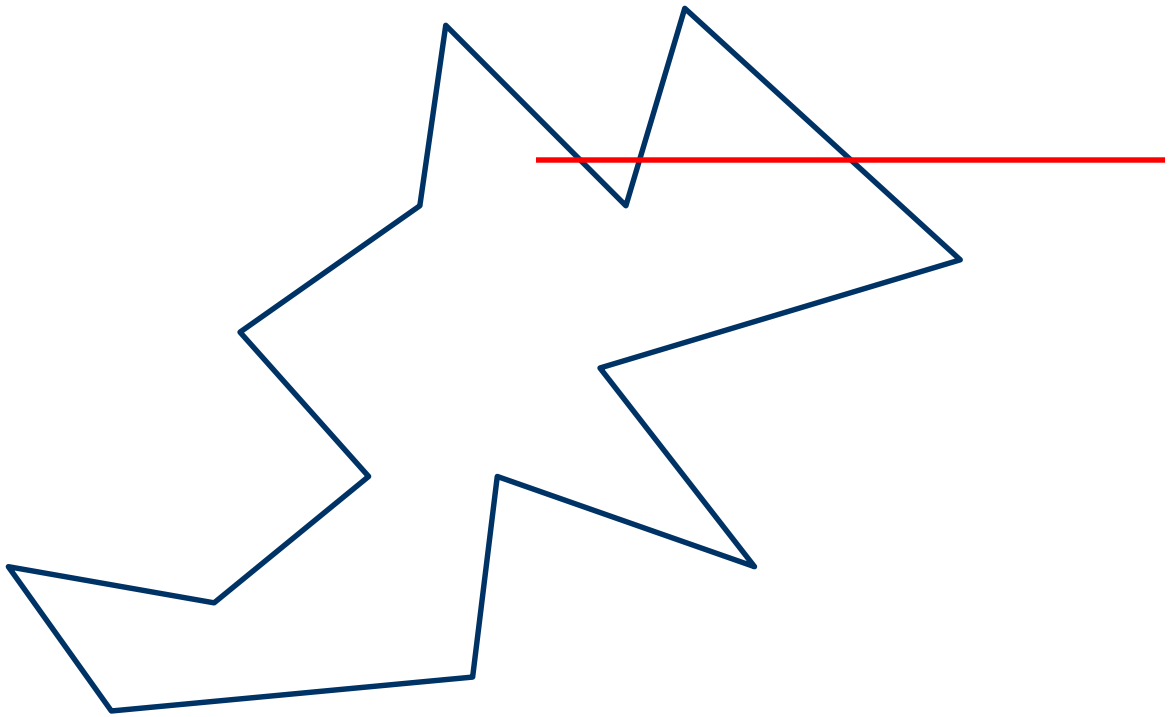


Talk Outline

- The Geodetic PIP Problem
- Related Work
- Our Approach
- Experiments
- Conclusions

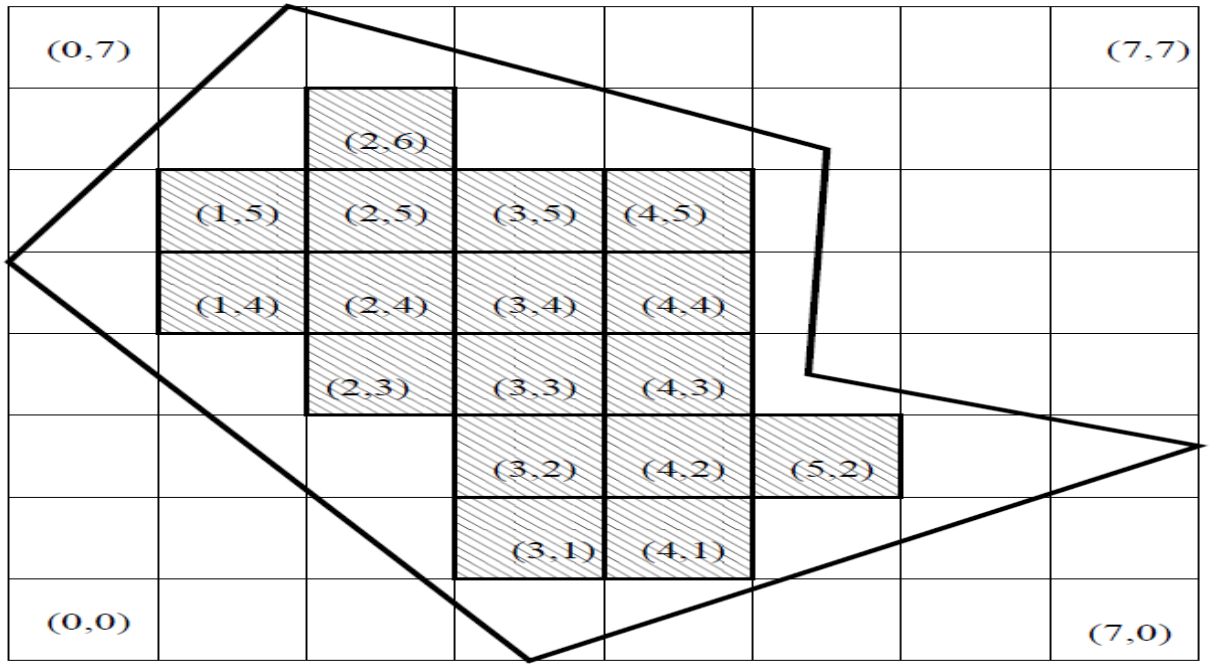
Related Work (I)

- Spatial indexes: Quadtree and R-tree indexes
- The ray-crossing PIP algorithm is well-known: O(N) algorithm, odd – inside; even - outside



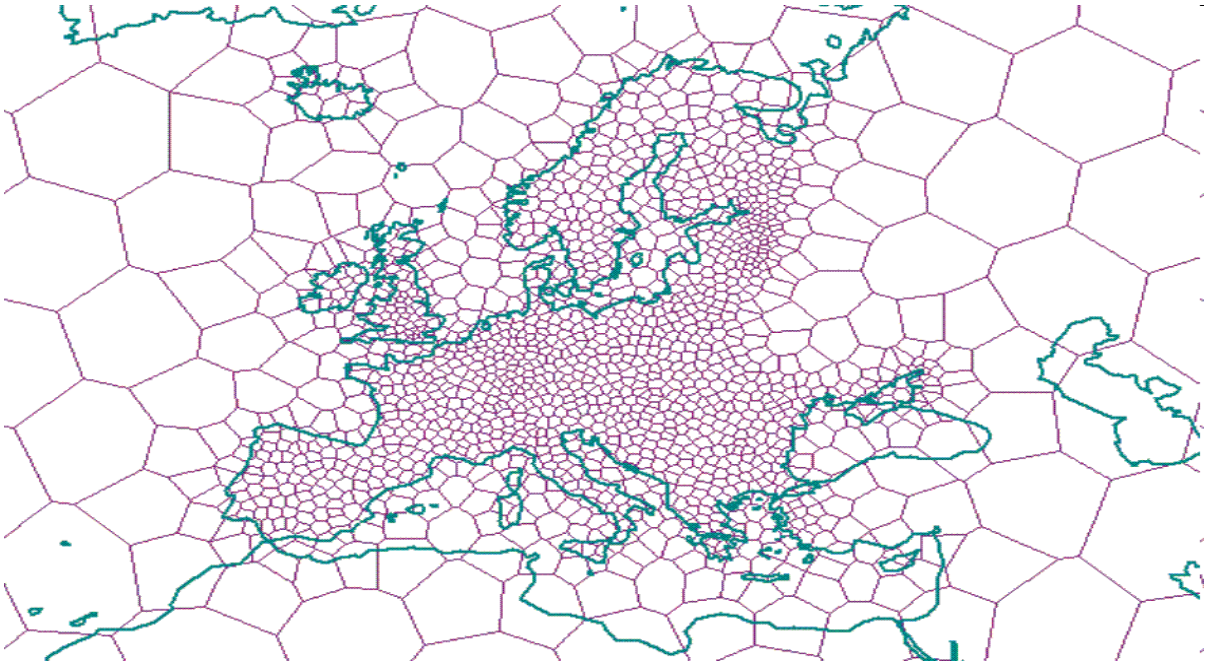
Related Work (II)

- Interior approximations or tessellations for non-geodetic polygons:
  - After Quadtree tessellation, if a point or an MBR is inside an interior tile, it must be in the polygon



Related Work (III)

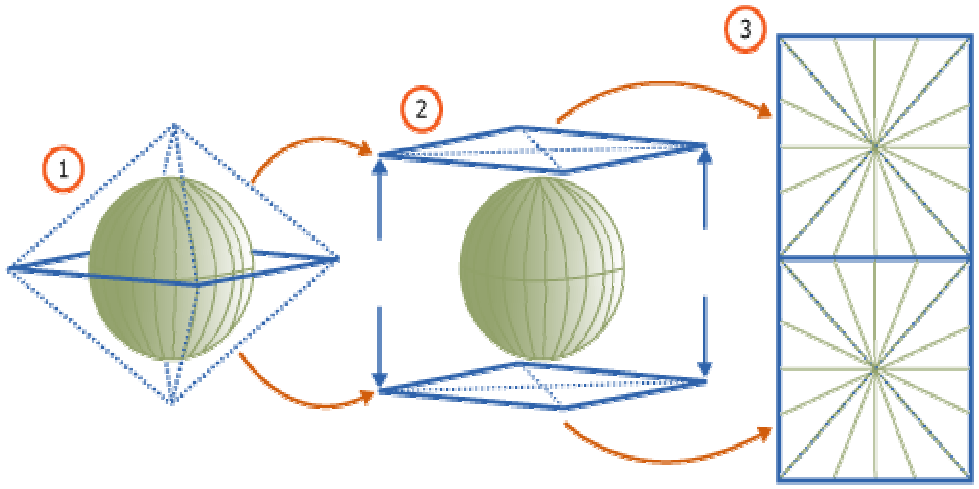
- Voronoi tesselllations: used in IBM Informix Geodetic DataBlade, appropriate tessellations are required.
  - Some tessellations are good for some data sets, but bad for other data sets

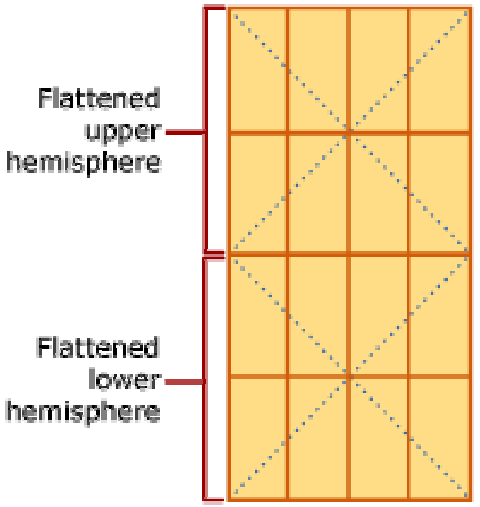


Voronoi Tessellation of Europe (from IBM Informix)

Related Work (IV)

- Microsoft SQL Server uses the geography grid tessellation scheme.
  - users need to understand tessellation rules





Spatial index on geography data type (from Microsoft SQL Server)

Talk Outline

- The Geodetic PIP Problem
- Related Work
- Our Approach
- Experiments
- Conclusions

Our Approach - Objectives

- Four objectives:
  - High performance
  - No user tuning
  - Small memory footprint $\Rightarrow$ high scalability for enterprise users
  - Handle Earth's curved surface, poles, equator, international date line

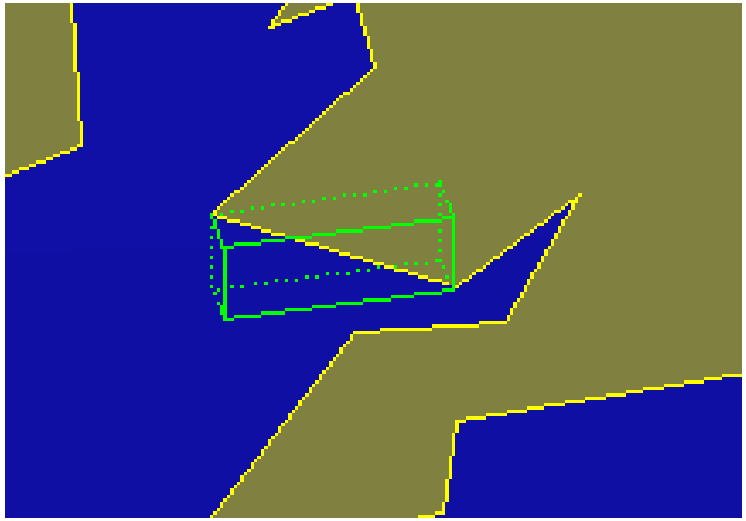
Our Approach – R-tree Index on 3D MBB

- 3D R-tree index on 3D Minimal Bounding Box
  - Geodetic 2D ⇒ Geocentric 3D

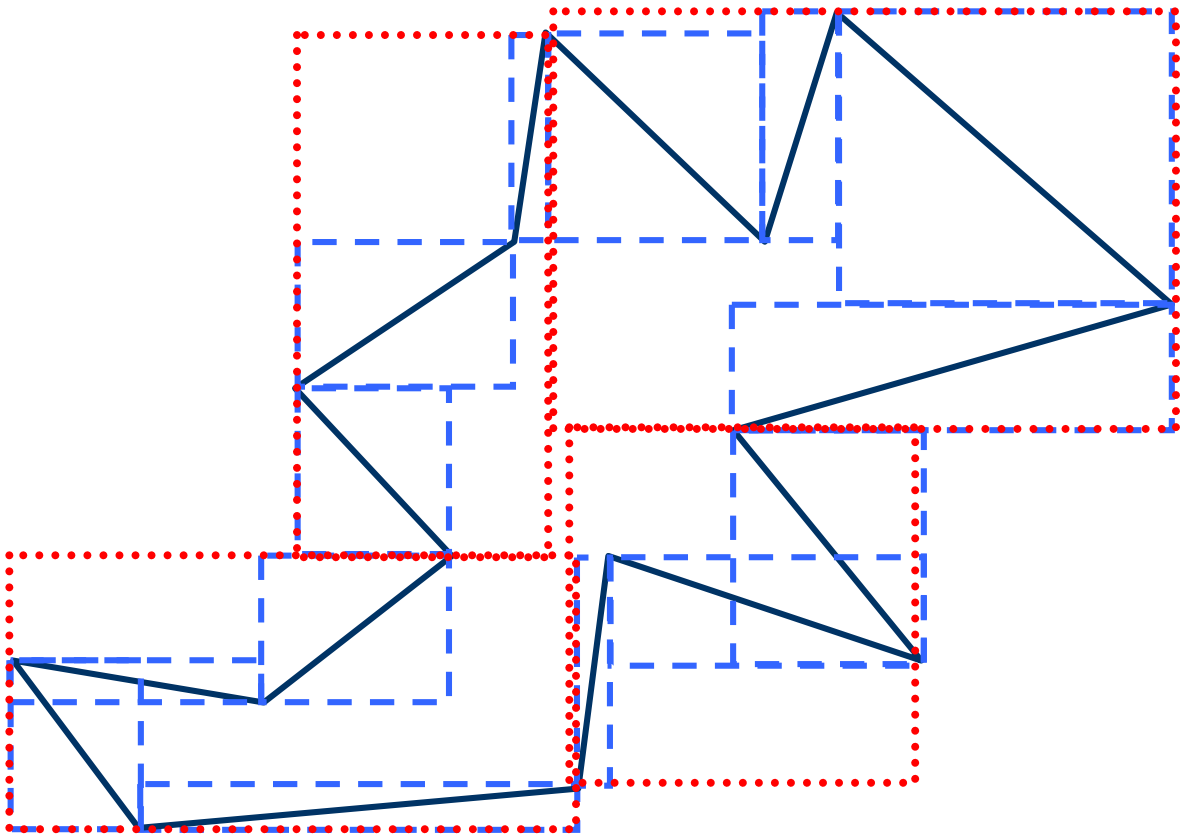
source: 123rf.com

Our Approach – Query Polygon Window (I)

- A hierarchy structure on a query polygon window: choose **in-memory 3D R-tree structure** because it is versatile



Our Approach – Query Polygon Window (II)



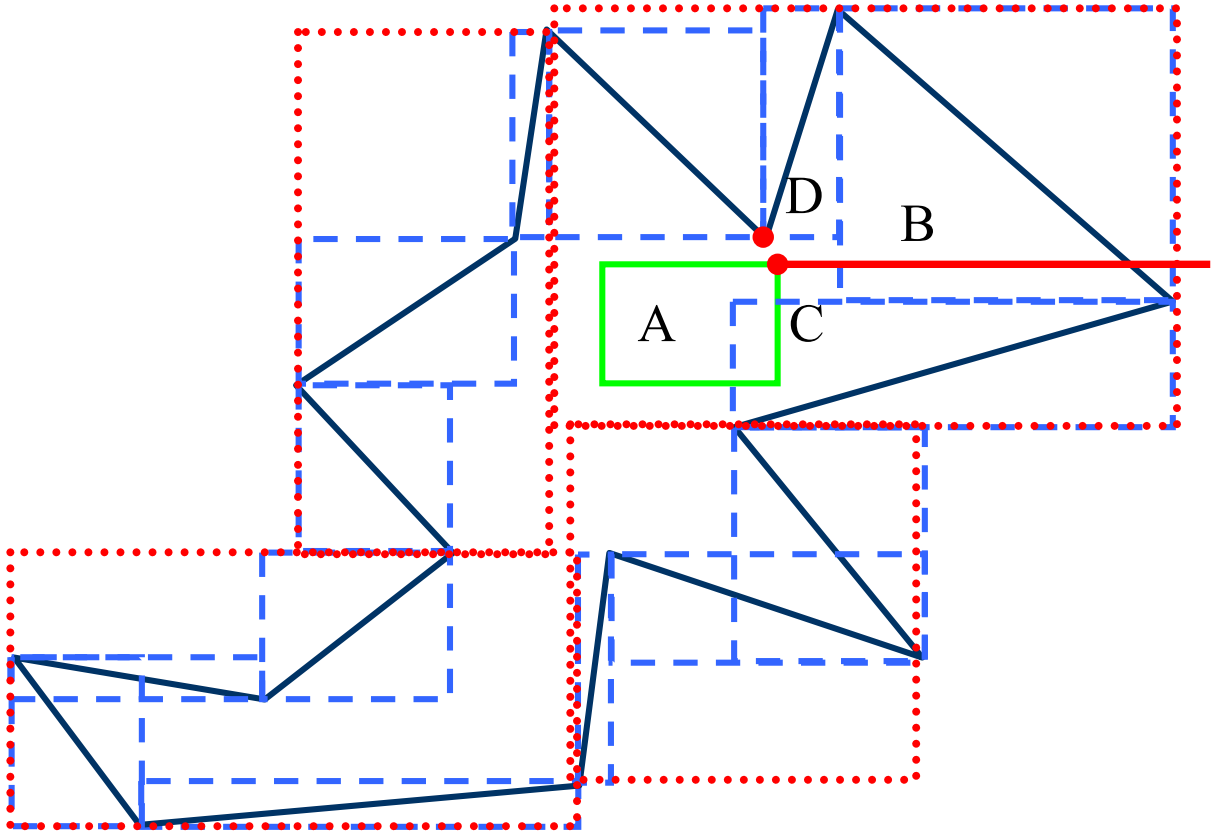
Our Approach – Use In-Memory R-tree in 2 Steps (I)

- Traditionally two-step spatial query processing:
  - (1) the filter step uses a spatial index only, e.g. uses only MBRs or MBBs in R-tree index
  - (2) the refinement step is a test on exact geometries

- Now our in-memory R-tree structure can be used not only in the refinement step, but also in the filter step

Our Approach – Use In-Memory R-tree in 2 Steps (II)

MBR (A) is from a R-tree index, either leaf node, or non-leaf node:
(1) MBR (A) intersects polygon? (2) MBR (A) is inside polygon?



Our Approach – The Algorithm (I)

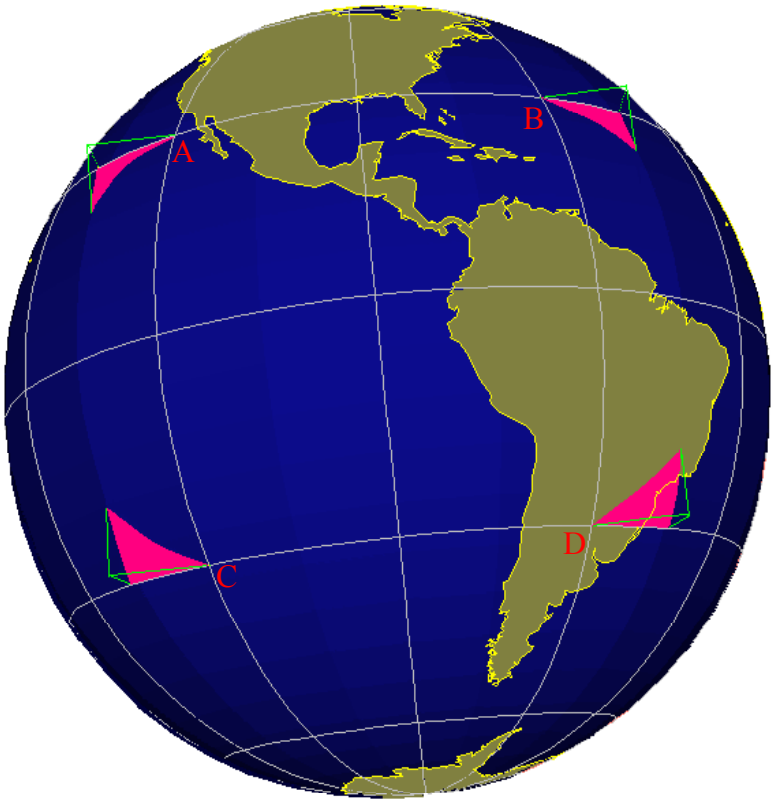
```
build an In-memory R-tree structure (IR) of the query polygon (Q)
push the R-tree index root into stack
while (stack is not empty) {
  pop an R-tree index entry (RE) from a stack;
  if (RE is not from an R-tree index leaf node) {
    if (RE's inclusion flag is set) {
      for each child entry of RE, set its inclusion flag; push it into stack; }
    else {
      determine the topological relationship between RE and Q using IR;
      if (RE is inside Q) {
        for each child entry of RE, set its inclusion flag; push it into stack; }
```

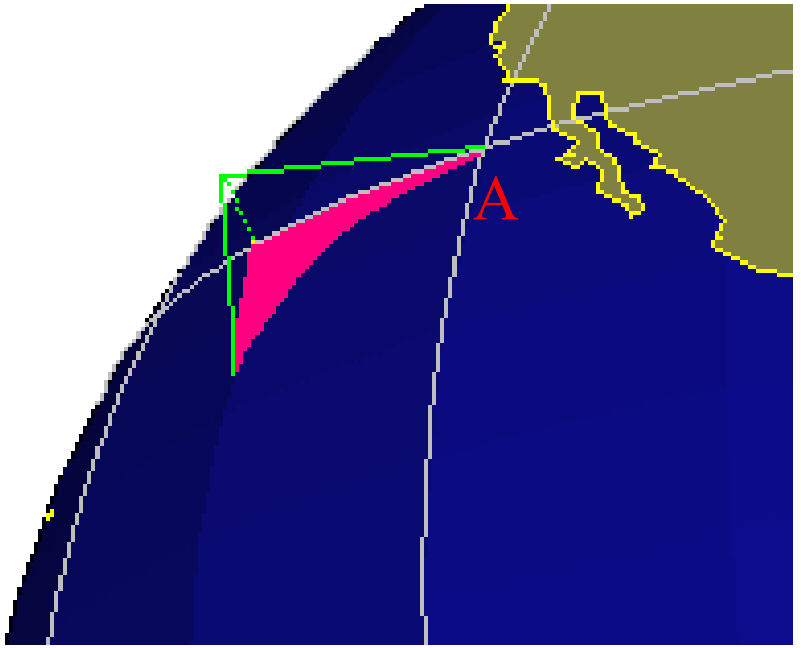
Our Approach – The Algorithm (II)

```
      else if (RE intersects with Q)
        for each child entry of RE, push it into stack;
      else /* RE is outside Q */
        continue; }}
    else  /* an entry in a leaf node */ {
     if (RE's inclusion flag is set) put this data point into the result set;
     else  {
      determine the topological relationship between RE and Q using IR;
      if (RE is inside Q)  put this data point into the result set;
      else continue;
} } }
```

Our Approach - At Most 8 Disconnected Regions Covered by A 3D MBB

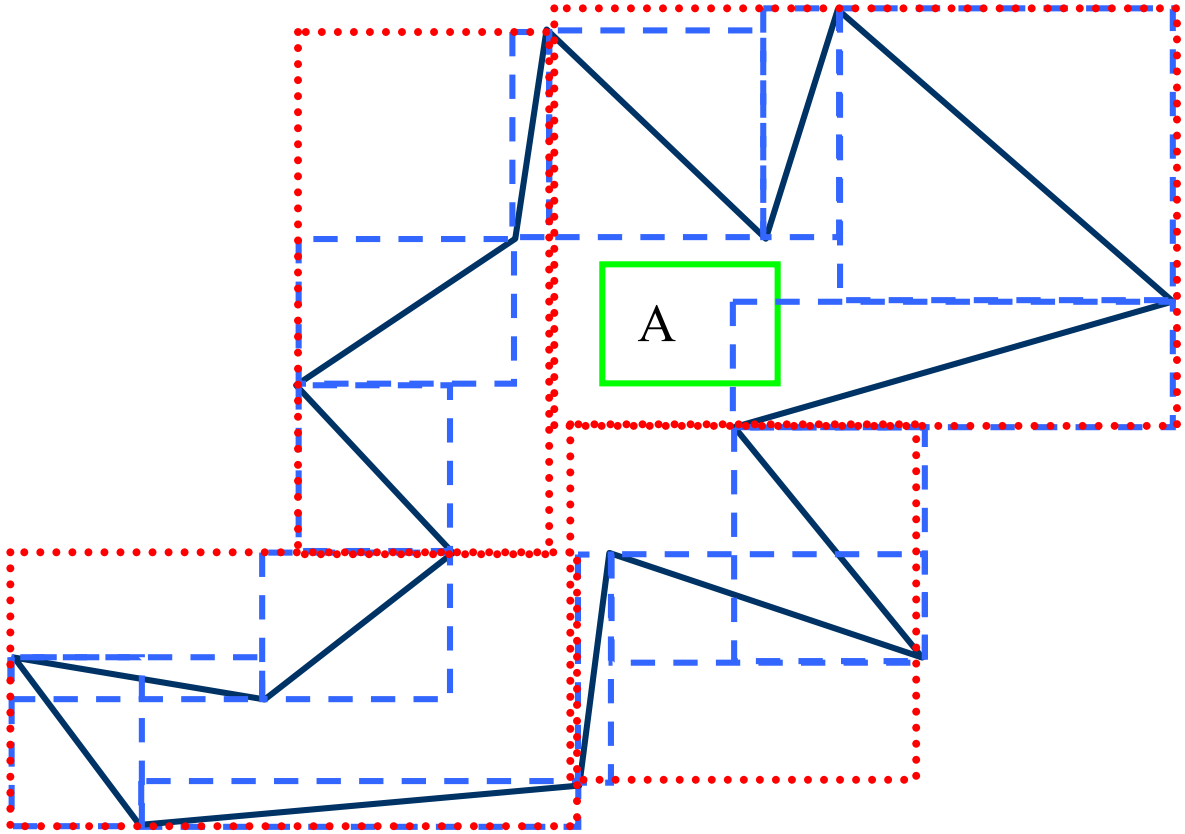
An MBB (in green) covers 4 regions (in pink) on the Western Hemisphere, it does not intersect the boundary of Northern Hemisphere, but intersects the Northern Hemisphere





Our Approach – Difference between Geodetic and Non-Geodetic

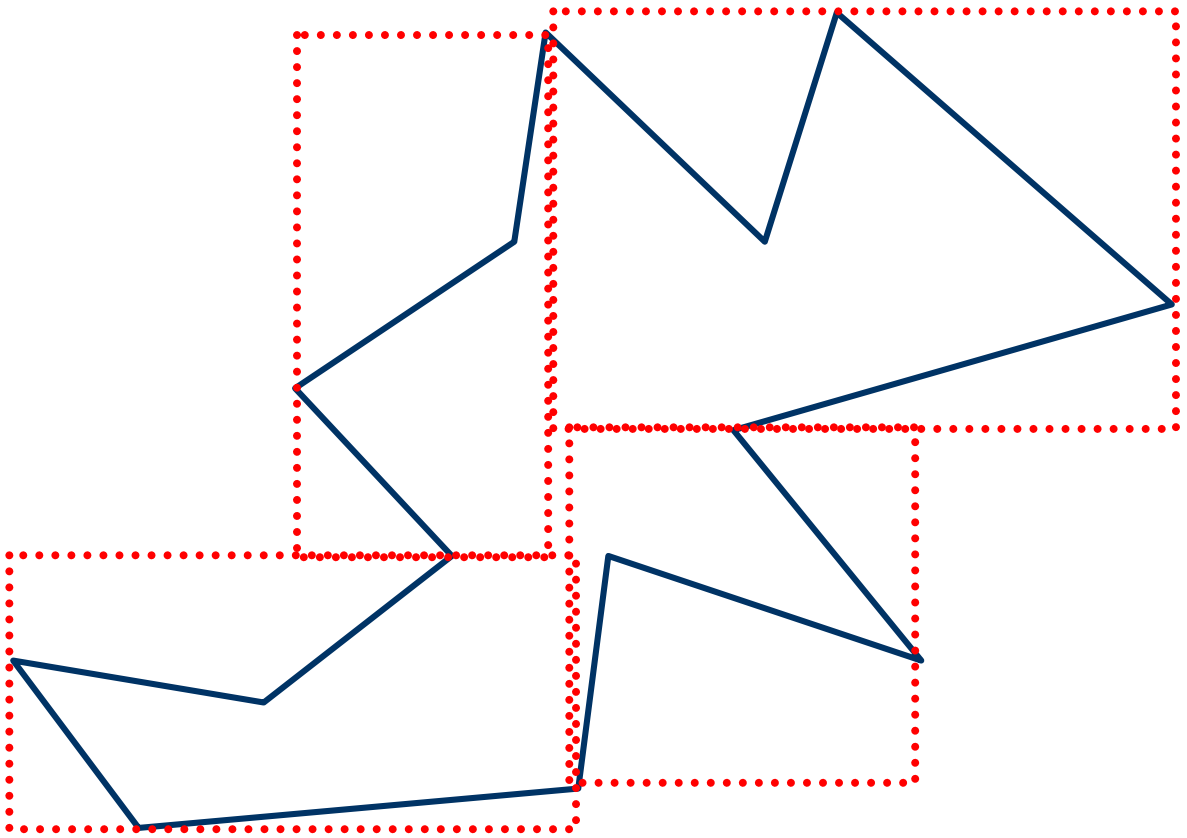
If MBR does not intersect with the boundary of polygon, it will be either completely outside polygon, or completely inside polygon



Our Approach – Memory Usage (I)

- To limit the memory usage, we group several line segments into a single leaf entry to construct the in-memory R-tree structure.
  - E.g. “Norway” has 343,395 line segments
  - 48M bytes, if no limit for leaf entry, 1 line segment ⇒ 1 entry
  - ~1.2M bytes, if the leaf entry limit is 8192, ~42 line segments ⇒ 1 entry
  - ~0.6M bytes, if the leaf entry limit is 4096 (default), ~84 line segments ⇒ 1 entry

Our Approach – Memory Usage (II)



Talk Outline

- The Geodetic PIP Problem
- Related Work
- Our Approach
- Experiments
- Conclusions

Experiments (I)

- Three real-world data sets
  - (1) **An energy company:** 39,000 locations in USA, a query polygon 4 times as big as Texas, 197,146 line segments
  - (2) **A transportation system:** 3 million locations and 30 regions around world: e.g. “Canada/Mountain”, “American/Honolulu”, “Germany”. average: 59,000 line segments, “Norway” 343,395 line segments
  - (3) **US Business Area data set:** 10 million locations
    - ➢ 50 US states, average: 1,755 line segments, “Alaska” 18,603 line segments;
    - ➢ 1061 local regions, average: 520 line segments, “Long Island”: 6,915 line segments

Experiments (II)

Three configurations

	Filter Step	Refinement Step
Conf. (A)	Use MBBs only	Use ray-crossing algorithm
Conf. (B)	Use MBBs only	Use in-memory R-tree
Conf. (C)	Use in-memory R-tree	Use in-memory R-tree

Experiments (III)

	Data Set (1)	Data Set (2)	Data Set (3) with 50 states	Data Set (3) with local reg.
Config. (A)	204s	20883s	2559s	11685s
Config. (B)	10.8s (20)	349s (60)	277s (9.2)	282s (41)
Config. (C)	1.4s (7.7)	88s (4.0)	52s (5.3)	81s (3.5)

Experiments (IV)

Limit of entries	Data Set (1)	Data Set (2)	Data Set (3) with 50 states	Data Set (3) with local reg.
NONE	1.4s	88s	52s	81s
8192	1.22s	88s	52s	81s
4096	1.4s	88s	53s	82s
2048	1.81s	90s	55s	82s
1024	2.26s	95s	56s	82s
512	3.85s	106s	61s	83s
256	5.7s	126s	70s	89s
128	10s	154s	90s	99s
64	18s	237s	124s	127s
32	28s	543s	170s	162s

Talk Outline

- The Geodetic PIP Problem
- Related Work
- Our Approach
- Experiments
- Conclusions

Conclusions

- Let's revisit four objectives:
  - High performance ✓
  - No user tuning ✓
  - Small memory footprint ⇒ high scalability for enterprise users ✓
  - Handle Earth's curved surface, poles, equator, international date line ✓

Thank you


QUESTIONS
ANSWERS

Backup / Non-geodetic Results (I)

	Data Set (1)	Data Set (2)	Data Set (3) with 50 states	Data Set (3) with local reg.
Config. (A')	71s	7778s	935s	2158s
Config. (B)	2.98s (23.8)	226s (34.4)	122s (7.66)	130s (16.6)
Config. (C)	2.44s (1.22)	190s (1.19)	58s (2.1)	64s (2.03)

Backup /Non-geodetic vs. Geodetic

	Data Set (1)	Data Set (2)	Data Set (3) with 50 states	Data Set (3) with local reg.
Config. (A' or A)	71s / 204s	7778s / 20883s	935s / 2559s	2158s / 11685s
Config. (B)	2.98s / 10.8s	226s / 349s	122s / 277s	130s / 282s
Config. (C)	2.44s / 1.4s	190s / 88s	58s / 52s	64s / 81s

Backup / Non-geodetic Results (II)

Limit of entries	Data Set (1)	Data Set (2)	Data Set (3) with 50 states	Data Set (3) with local reg.
NONE	2.44s	190s	58s	64s
8192	2.02s	190s	58s	64s
4096	2.03s	190s	58s	64s
2048	2.08s	191s	59s	64s
1024	2.09s	191s	59s	64s
512	2.16s	191s	60s	64s
256	2.5s	195s	62s	64s
128	3.12s	201s	68s	68s
64	5.05s	213s	76s	69s
32	8.43s	240s	92s	80s

Backup / NN-Based PIP vs. Ray-Based PIP