Finding All Nearest Neighbors with a Single Graph Traversal

Yixin Xu, Jianzhong Qi, Renata Borovica-Gajic, and Lars Kulik

Motivation

Parking undersupply?

65% oversupply

SMART PARKING

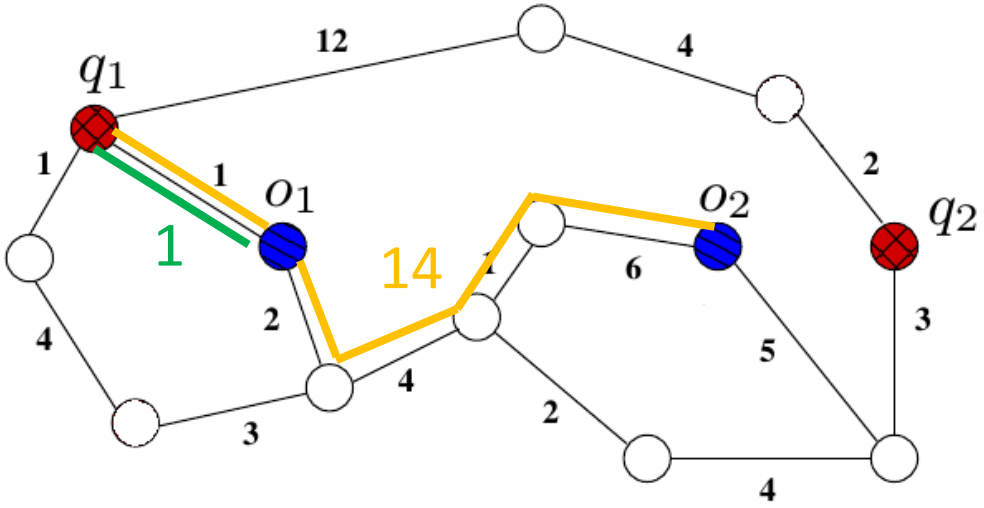
[1] https://www.eveningtelegraph.co.uk/fp/park-ride-scheme-considered-ease-parking-dundees-ninewells-hospital/
[2] http://nelsonnygaard.com/publication/parking-in-mixed-use-districts/
[3] http://www.global.datafest.net/projects/smart-parking-imt

Problem definition

- Find the nearest parking space for every driver

Efficient & scalable *All Nearest Neighbour* (ANN) algorithm

- Example:



Query objects

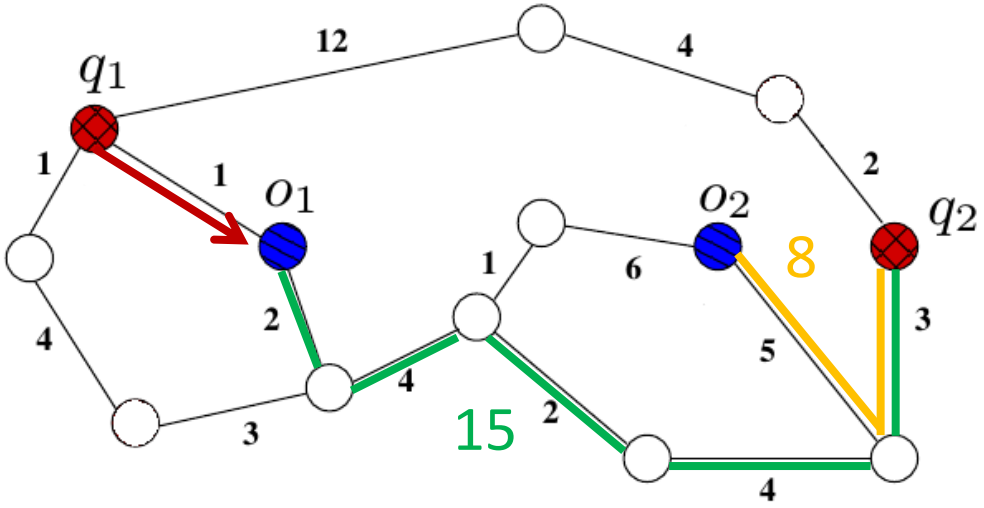
Data objects

Problem definition

- Find the nearest parking space for every driver

Efficient & scalable *All Nearest Neighbour* (ANN) algorithm

- Example:



Query objects

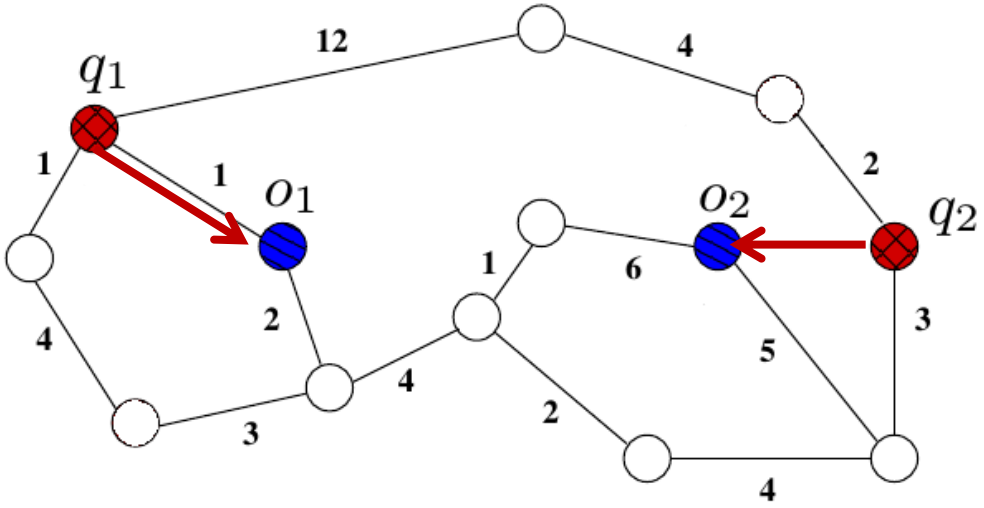
Data objects

Problem definition

- Find the nearest parking space for every driver

Efficient & scalable *All Nearest Neighbour* (ANN) algorithm

- Example:

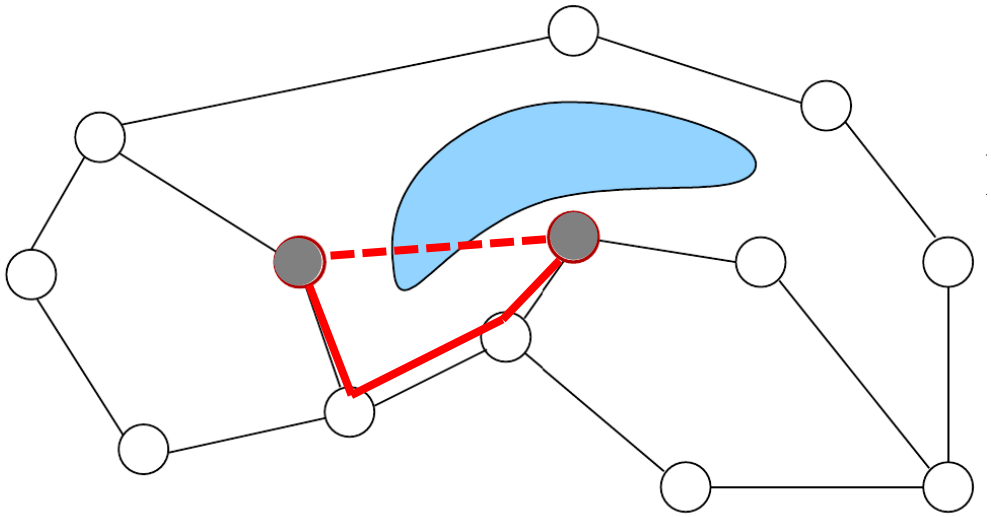


Query objects

Data objects

Literature review

- ANN algorithms in Euclidean space cannot be applied



VIVET the first study on ANN problem in spatial networks

THE UNIVERSITY OF MELBOURNE

Existing NN algorithms comparison

	INE	G-tree	ROAD	IER-PHL	DisBrw
Query time	5th	2nd	3rd	1st	3rd
Precomputation time	1st	3rd	2nd	4th	5th
Precomputation memory	1st	2nd	3rd	4th	5th

State-of-the-arts: IER-PHL, G-tree, INE

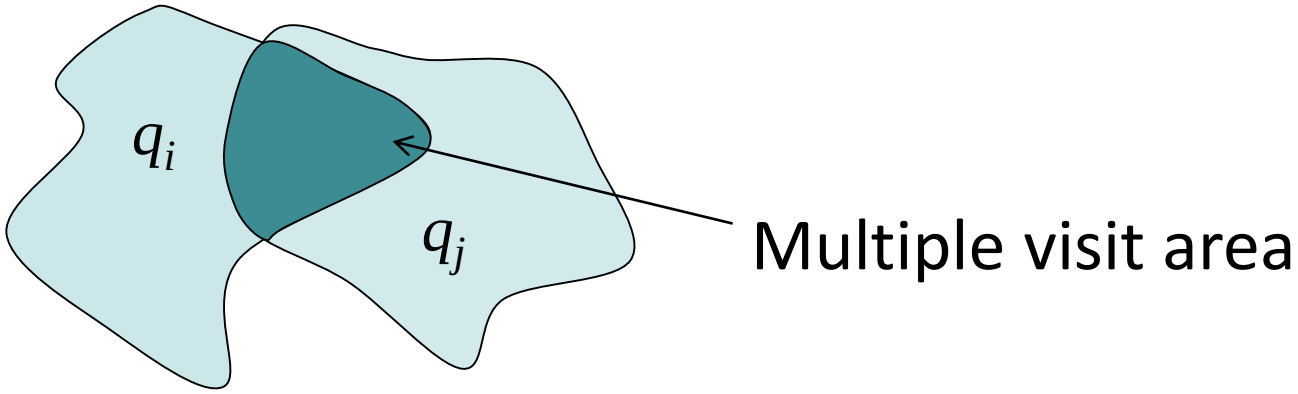
[4] Abeywickrama, T., Cheema, M.A., Taniar, D.: K-nearest neighbors on road networks: a journey in experimentation and in-memory implementation. PVLDB 9(6), 492–503 (2016)

Limitation of NN algorithms

- Large memory cost, not scalable to large networks

US road network (23.9 million vertices)			
Memory	IER-PHL	G-tree	VIVET
	>64 GB	2.4 GB	182.7MB

- Multiple visit to the same areas, not efficient for large query sets

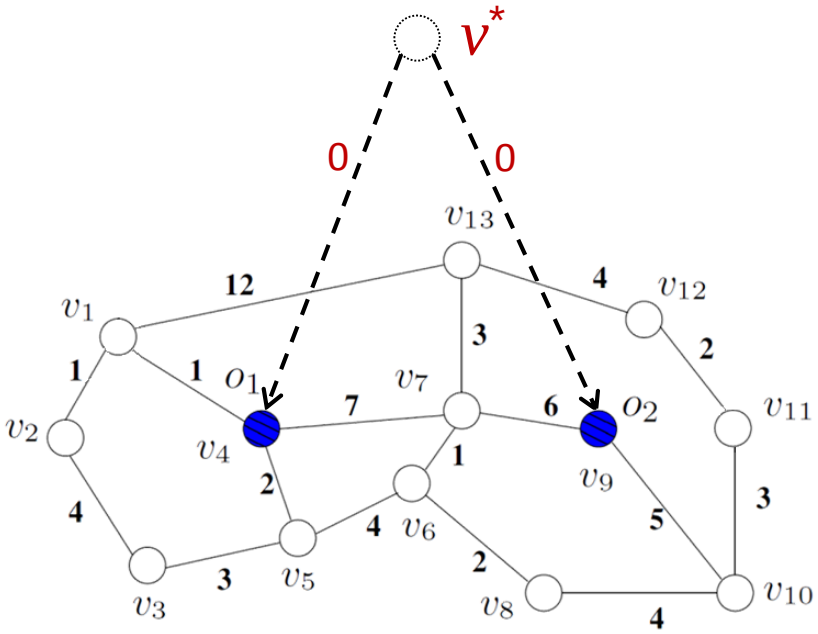


VIVET overview

- Precomputation phase
  - Traverse the graph only once
  - Short precomputation time
  - Low memory size
- Query phase
  - Answer a NN query in constant time
  - Answer an ANN query in linear time

Precomputation phase

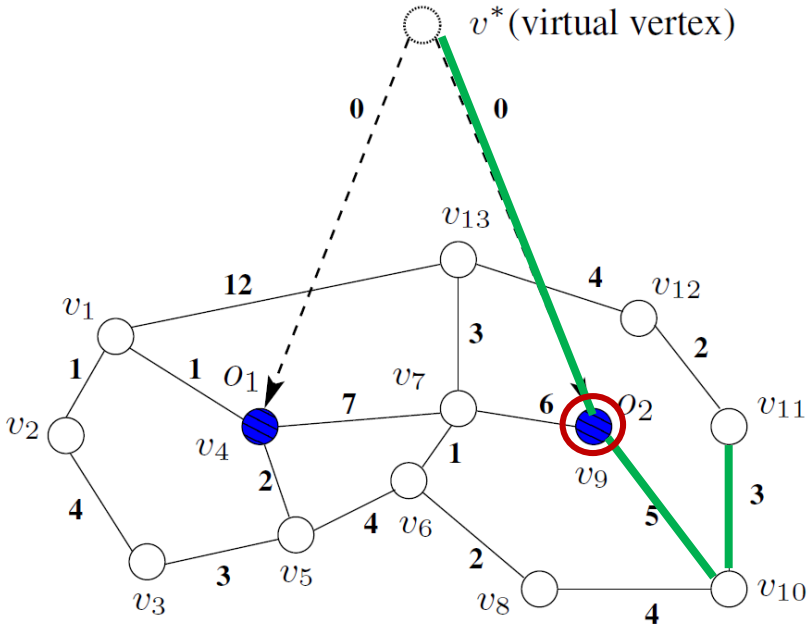
- Precomputation algorithm
  - Step 1, add a virtual vertex v^*
  - Step 2, connect v^* with all data objects with weight zero
  - Step 3, traverse the road network from the virtual vertex (Dijkstra's algorithm)
- Example



Precomputation phase

- Get NN(v_i) from SP(v^*, v_i)
  - SP(v^*, v_i) must traverse exactly one data object
  - The traversed object is the nearest neighbour (NN) of v_i

- Example

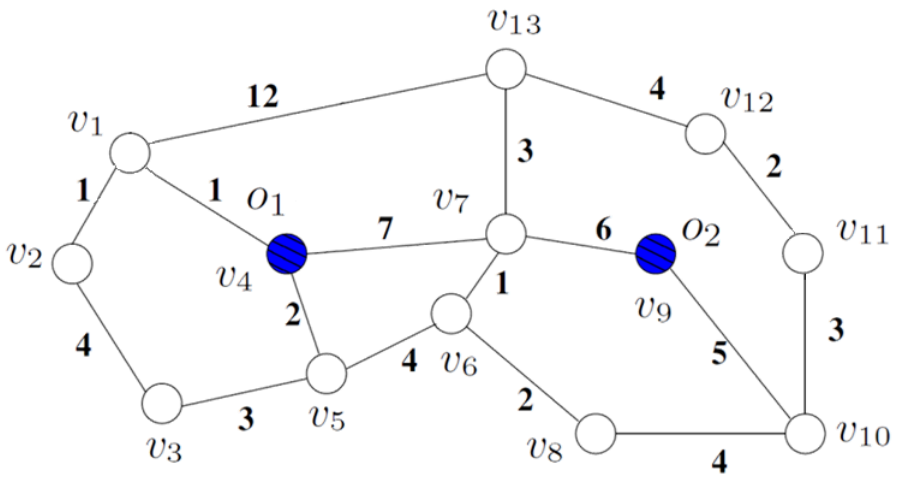


SP(v^*, v_{11}) = {v^*, o_2, v_{10}, v_{11}}

NN(v_{11})=o_2

The Index of VIVET

- VIVET index



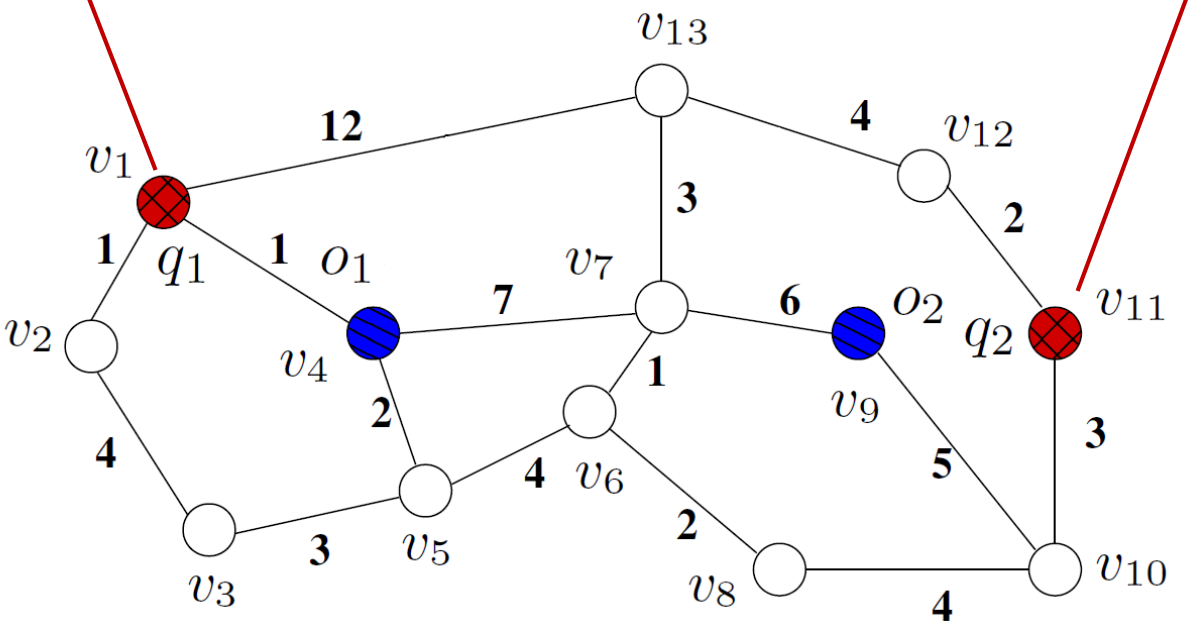
	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}
NN	o_1	o_1	o_1	o_1	o_1	o_1	o_2	o_1	o_2	o_2	o_2	o_2	o_2
distance	1	2	5	0	2	6	6	8	0	5	8	10	9

Memory: linear to the number of vertices

Query phase

- Query algorithm

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}
NN	o_1	o_1	o_1	o_1	o_1	o_1	o_2	o_1	o_2	o_2	o_2	o_2	o_2
distance	1	2	5	0	2	6	6	8	0	5	8	10	9



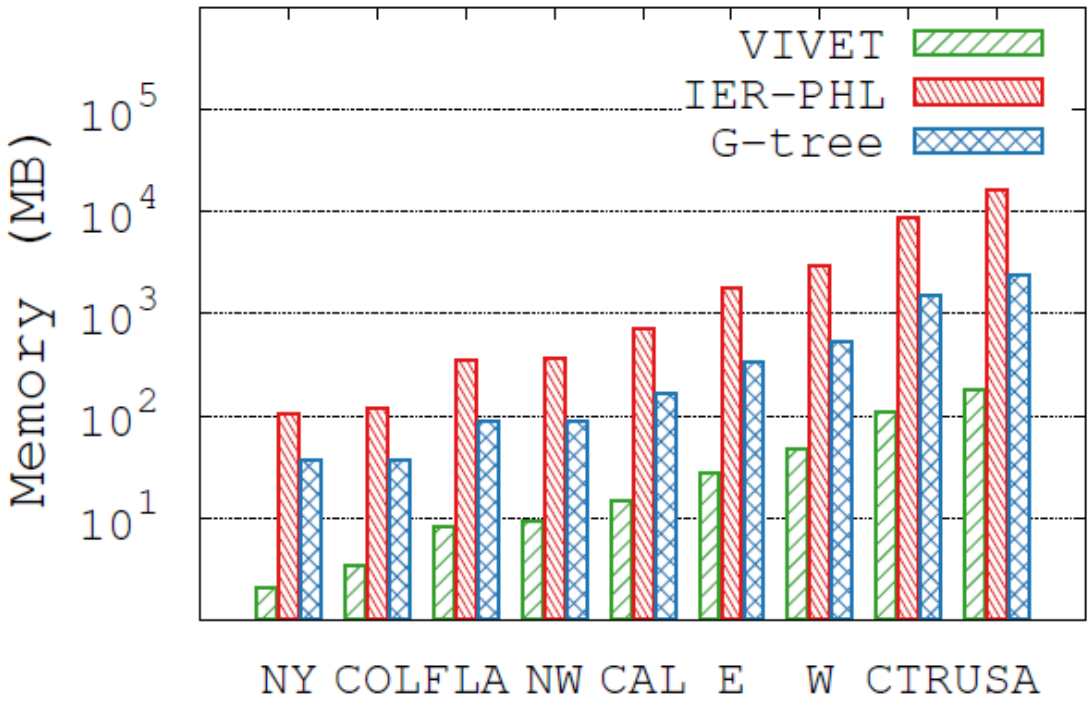
Experiments

- Datasets
  - Road network: 9th DIMACS Implementation Challenge[5]
  - Real-world data objects from OpenStreetMap[4]
  - Synthetic objects
- Implementation
  - C++
  - 64-bit virtual node with 1.8GHz GPU and 64GB RAM from Nectar[6]

[4] Abeywickrama, T., Cheema, M.A., Taniar, D.: K-nearest neighbors on road networks: a journey in experimentation and in-memory implementation. PVLDB 9(6), 492–503 (2016)
[5] http://www.dis.uniroma1.it/challenge9/download.shtml
[6] https://nectar.org.au

Precomputation performance

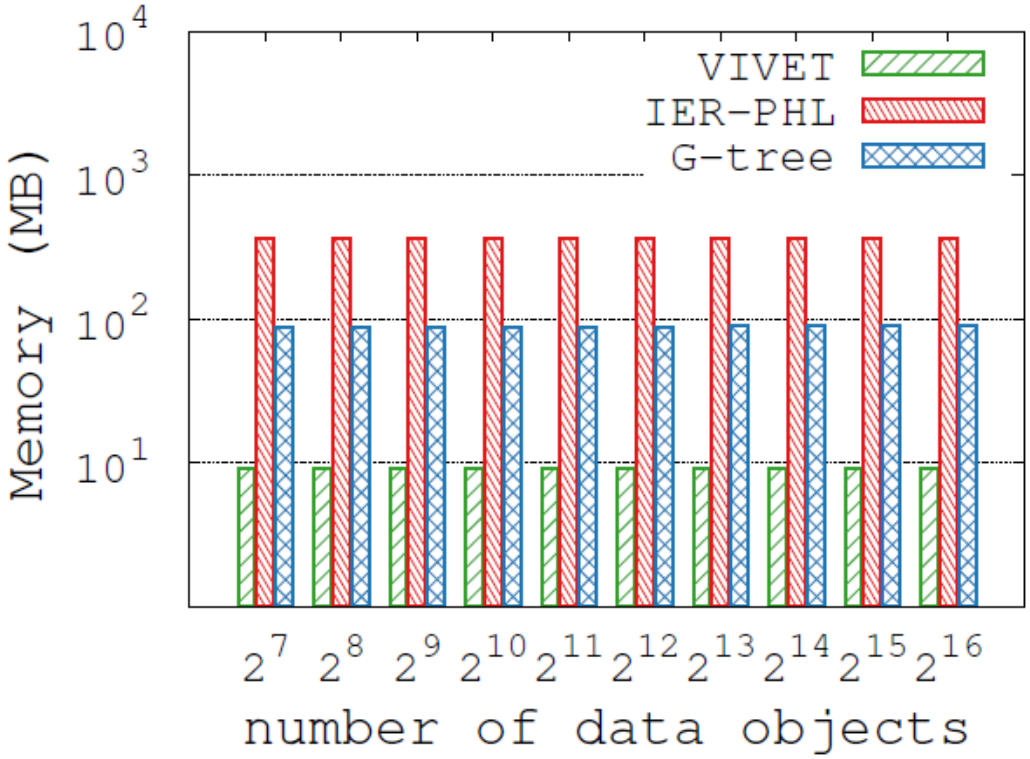
- Precomputation memory
  - Vary the road network size



VIVET reduces the memory consumption by one order of magnitude

Precomputation performance

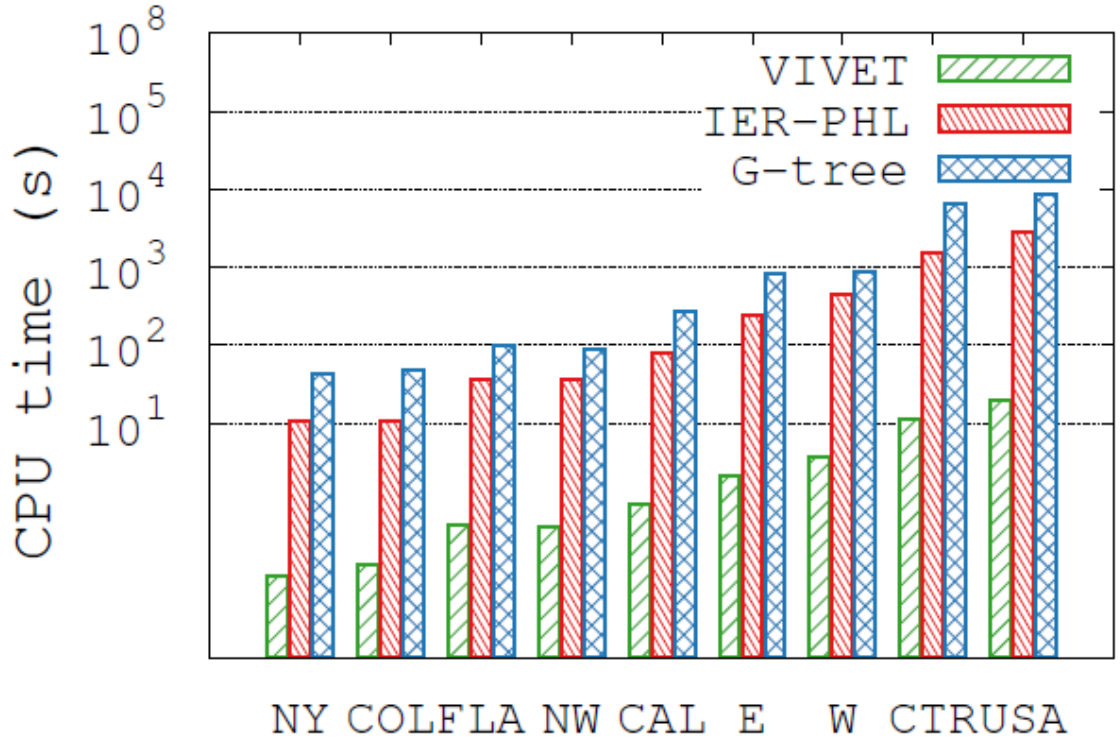
- Precomputation memory
  - Vary the number of data objects



VIVET reduces the memory consumption by one order of magnitude

Query performance

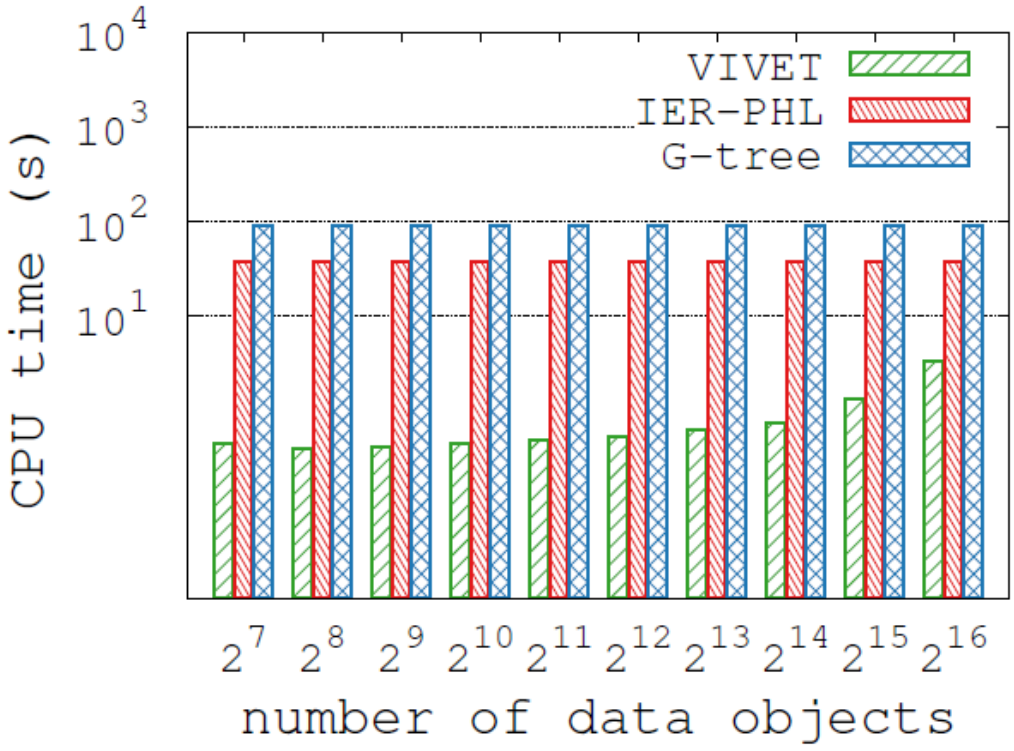
- Precomputation time
  - Vary the road network size



VIVET reduces the precomputation time by one order of magnitude

Precomputation performance

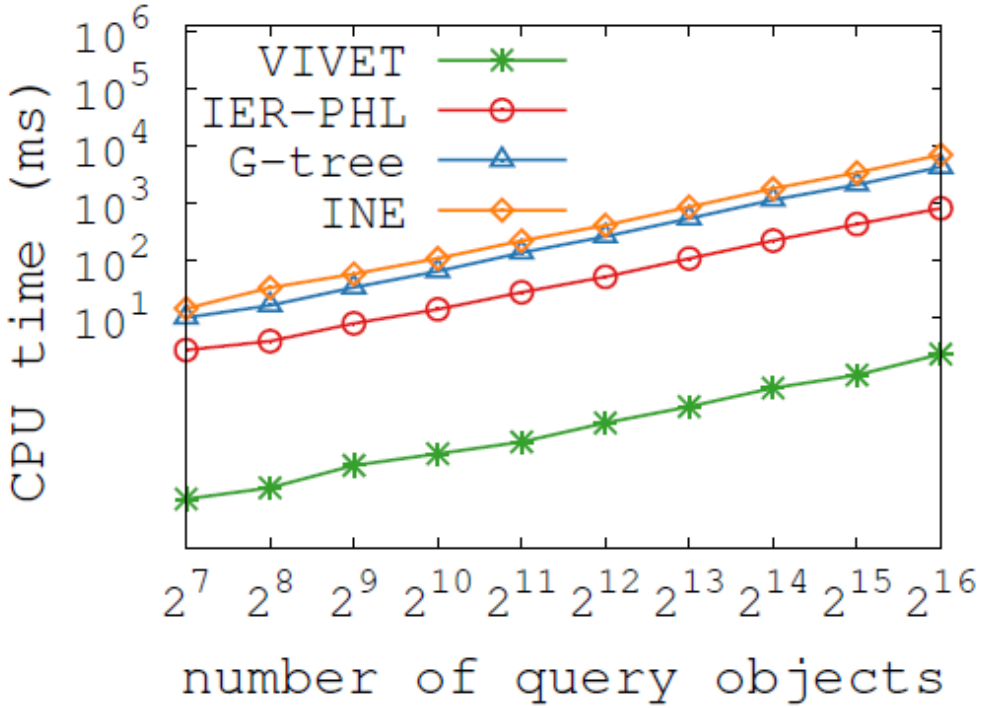
- Precomputation time
  - Vary the number of data objects



VIVET reduces the precomputation time by one order of magnitude

Precomputation performance

- Query time
  - Vary the number of query objects



VIVET outperforms state-of-the-art by more than two orders of magnitude

Extensions

- VIVET in directed graphs
  - Reverse the road network edges
  - Apply VIVET on the reversed graph
- VIVET without index
  - Run the precomputation phase online

Conclusion and future work

- Conclusion
  - ANN is a fundamental query in spatial database
  - The size of VIVET index is linear to the number of vertices
  - VIVET answers an ANN query in linear time

- Future work
  - All k nearest neighbor
  - Other nearest neighbor problems, i.e., continuous nearest neighbor, reverse nearest neighbor

Thank you