Accelerating Nearest Neighbor Search on CUDA for Learning Point Clouds

Lixin Xue

Supervised by

Yifan Wang, Prof. Cengiz Oztireli, Prof. Olga Sorkine-Hornung

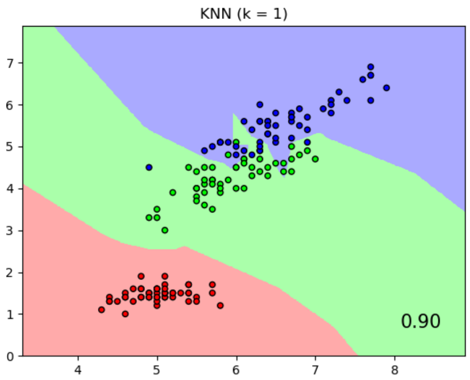
INTERACTIVE GEOMETRY LAB

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

October 20, 2020

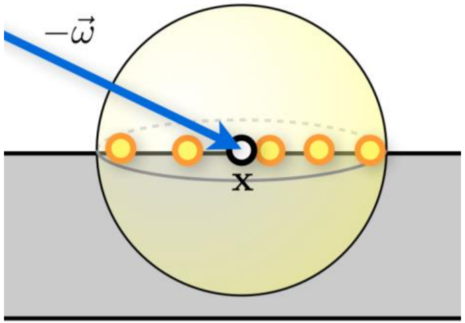
Motivation

- Nearest Neighbor Search is everywhere



KNN Classifier

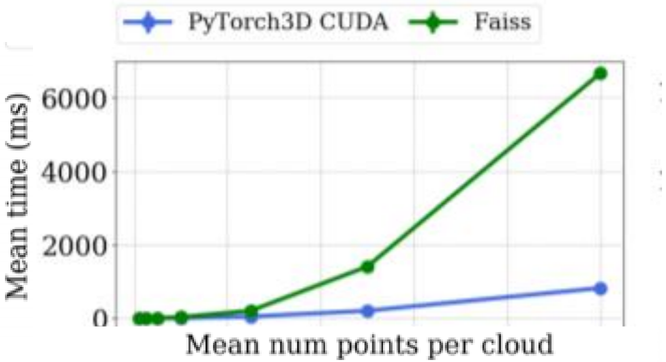
Image Retrieval



Photon Mapping

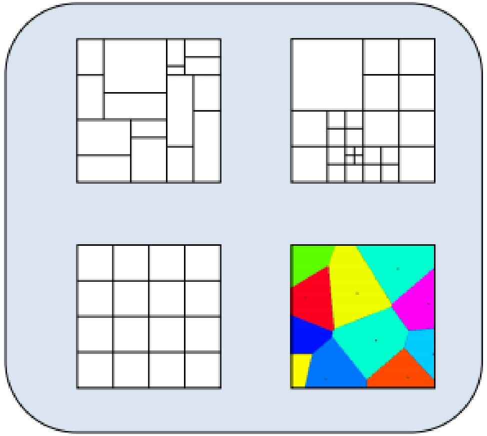
Existing Solutions

- ANN Benchmark
- Facebook FAISS
- Facebook Pytorch3D's KNN
- Pytorch3D is Faster
  - Still too slow
  - Brute force



Possible Solutions

- Spatial Partitioning to speed up?
  - KD-Tree, Octree, Uniform Grid etc.
- GPU Characteristics
  - Thread Divergence
  - Coalesced Read
  - Limited Shared Memory



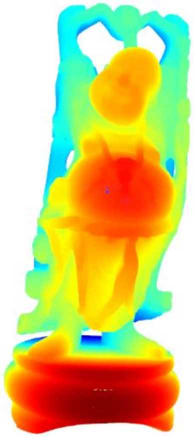
Fixed Radius NN vs KNN

- Fixed Radius NN
  - + Lower thread divergence
  - + More invariant to point density change
  - + Easier to implement
  - - Need to specify the search radius
  - - Need to deal with the case of few neighbors

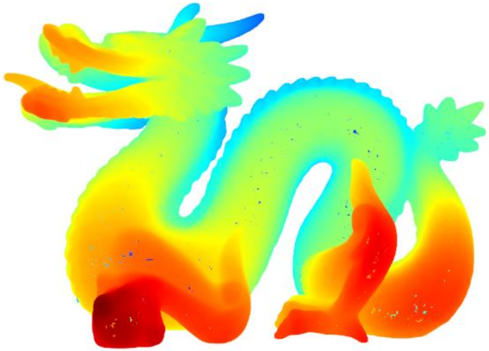
Achievements

- Fixed Radius NN Search vs Pytorch3D’s KNN

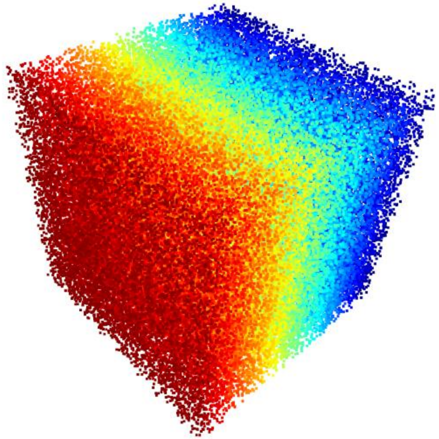
172974 points
~16x speed up



543652 points
~12x speed up



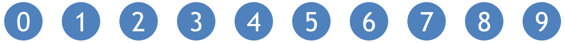
437645 points
~15x speed up

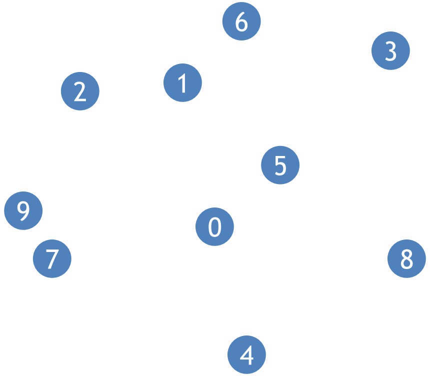


100000 points
~30x speed up

Algorithm Walkthrough

- Reference Points





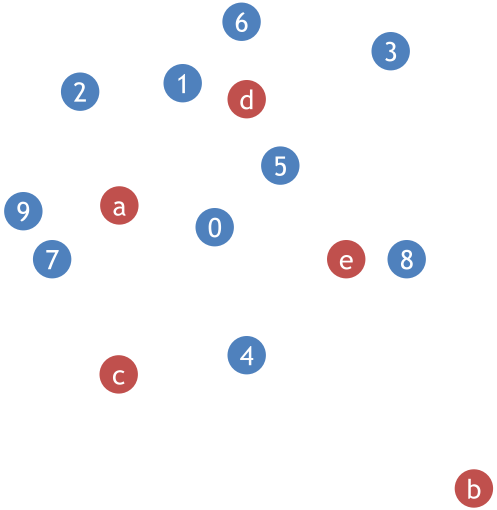
Algorithm Walkthrough

- Reference Points

0 1 2 3 4 5 6 7 8 9

- Query Points

a b c d e



Algorithm Walkthrough

- Reference Points

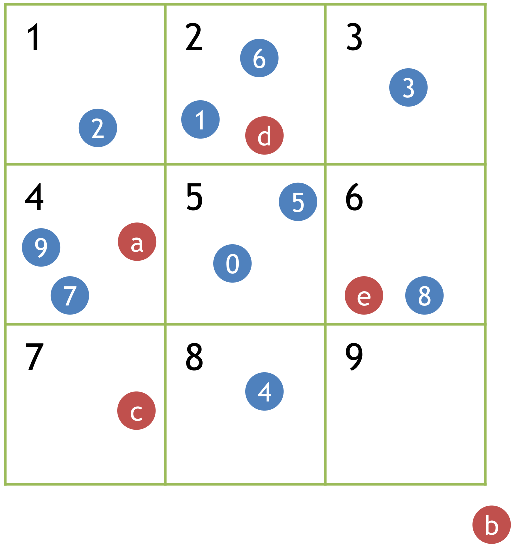
0 1 2 3 4 5 6 7 8 9

- Query Points

a b c d e

- Uniform Grids

1 2 3 4 5 6 7 8 9

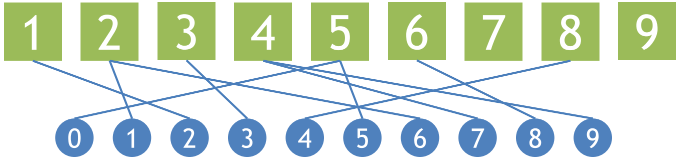


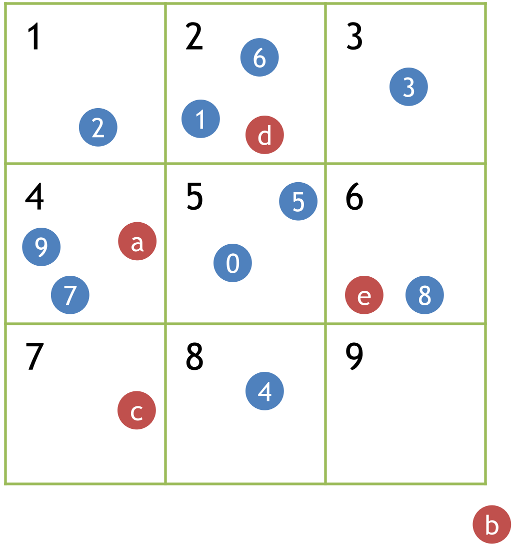
Algorithm Walkthrough

- Reference Points

- Query Points

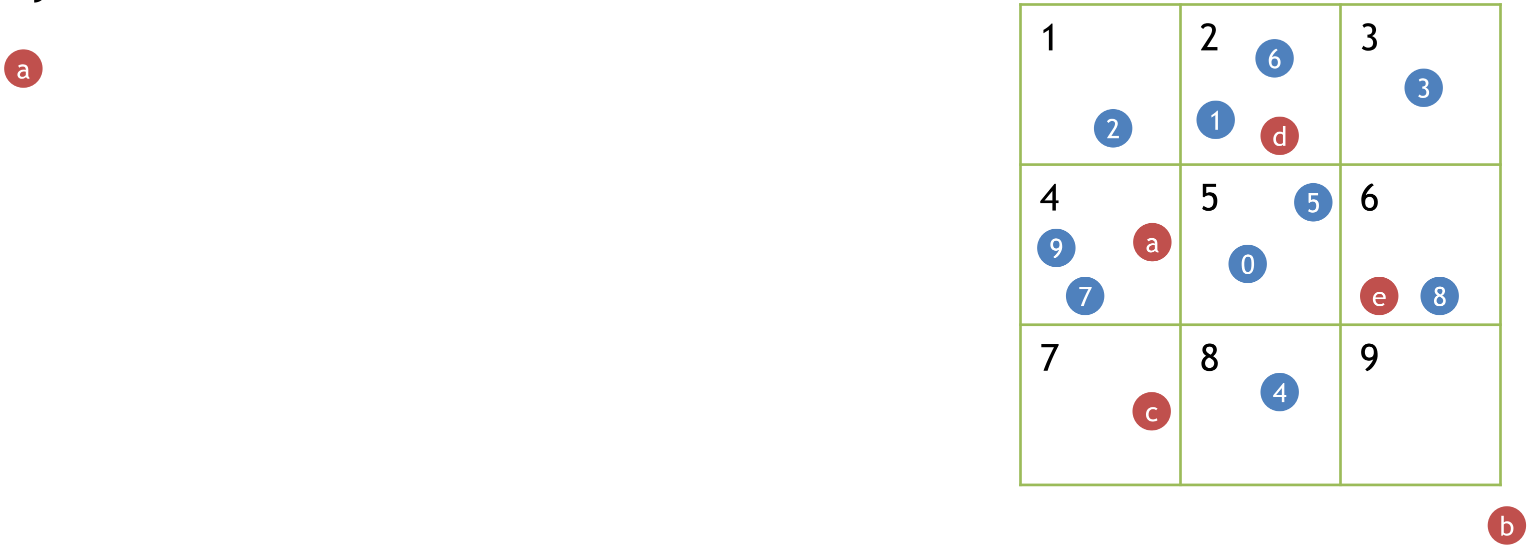
- Uniform Grids





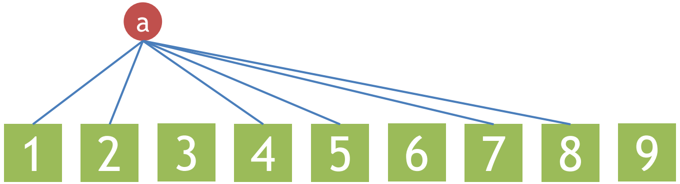
Algorithm Walkthrough

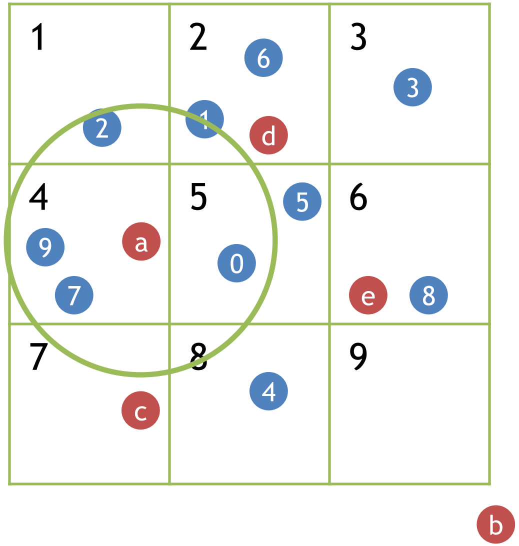
- Query Points



Algorithm Walkthrough

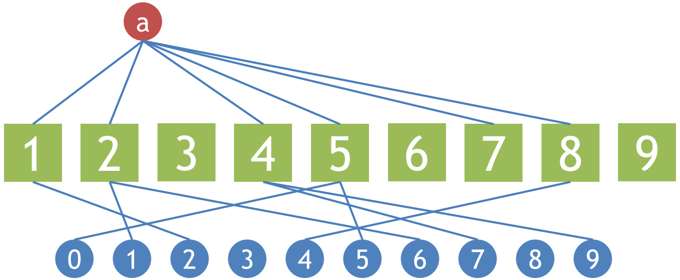
- Query Points

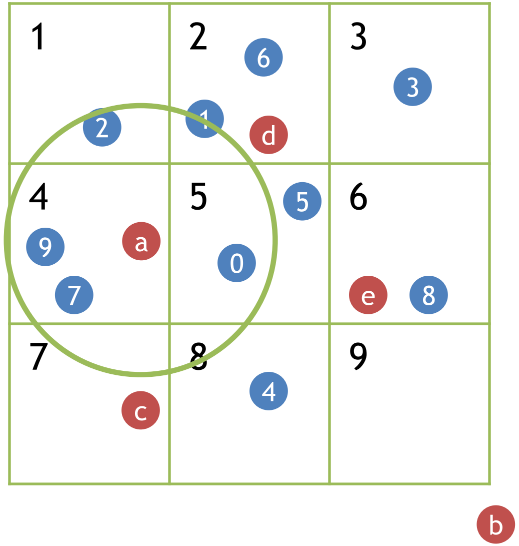




Algorithm Walkthrough

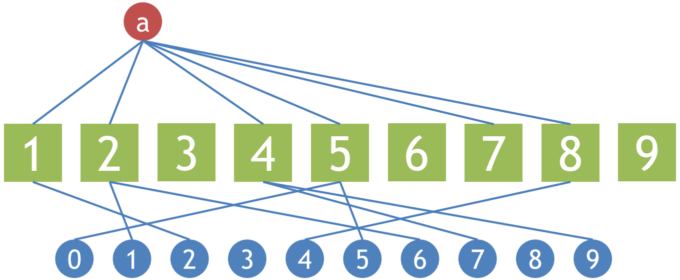
- Query Points



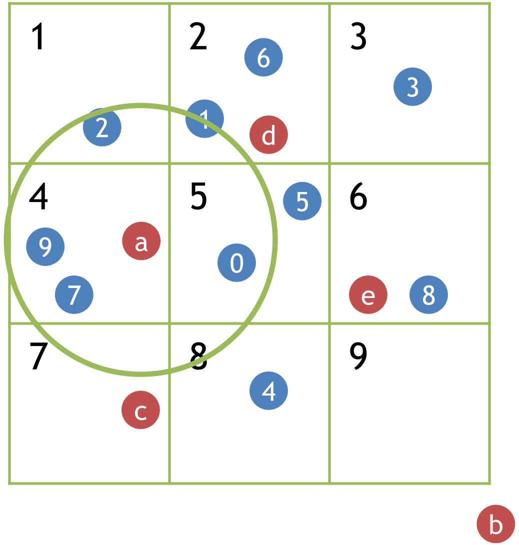


Algorithm Walkthrough

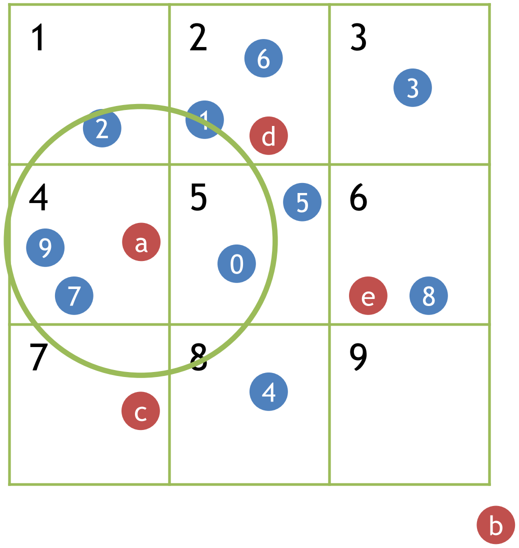
- Query Points



⚠ Scatter Reads

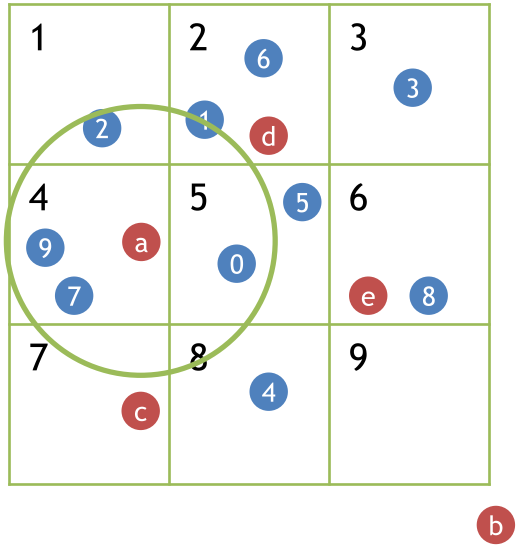


Algorithm Walkthrough



Algorithm Walkthrough

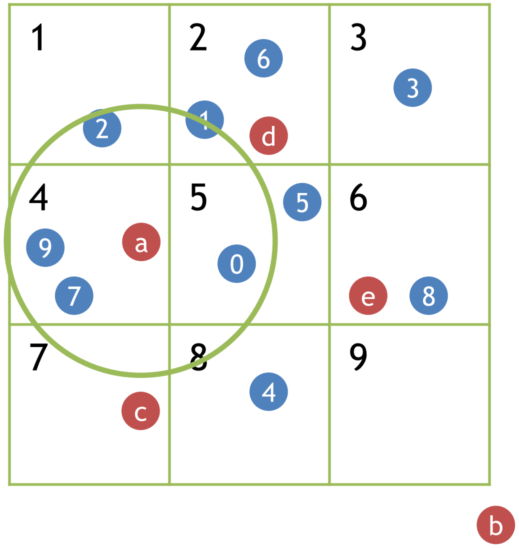
- Spatially sort the reference points



Algorithm Walkthrough

- Spatially sort the reference points

2 1 6 3 9 7 0 5 8 4

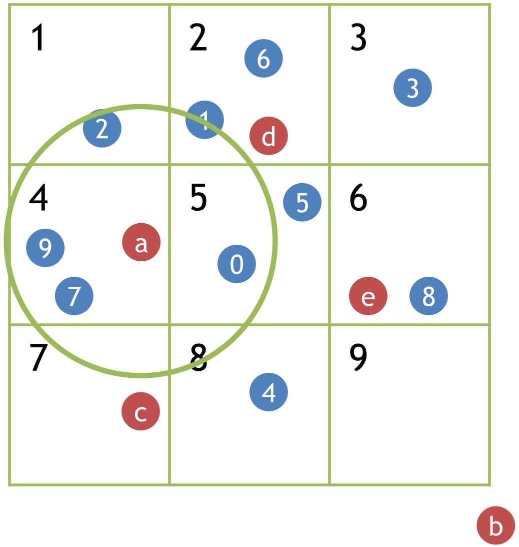


Algorithm Walkthrough

- Spatially sort the reference points

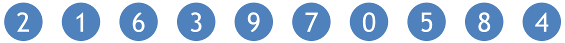
2 1 6 3 9 7 0 5 8 4

- More coherent reads!

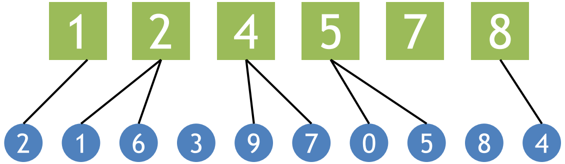


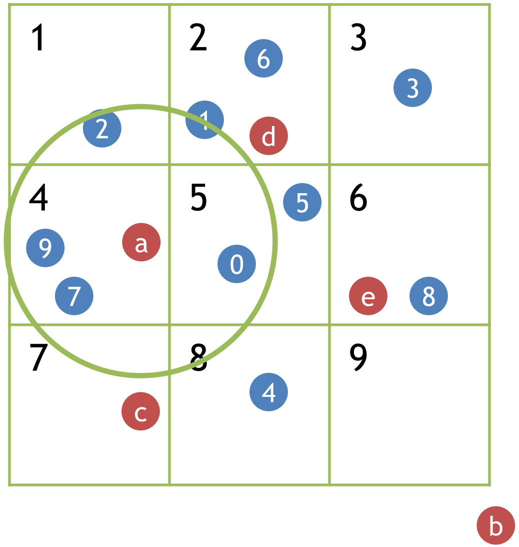
Algorithm Walkthrough

- Spatially sort the reference points



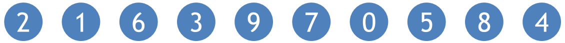
- More coherent reads!



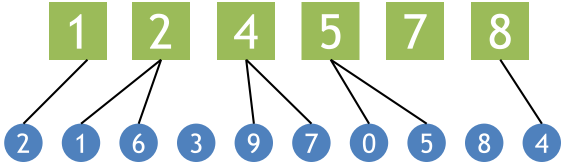


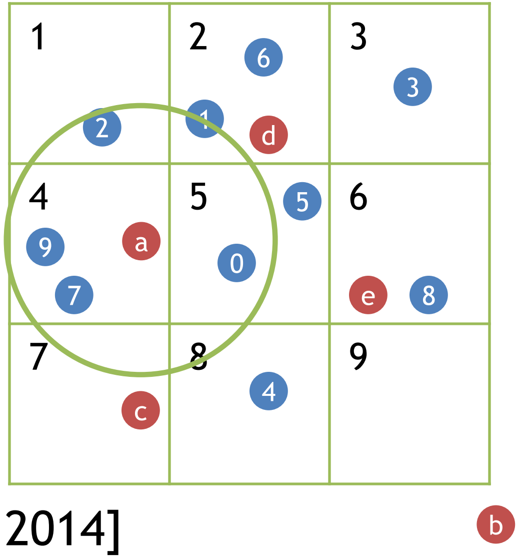
Algorithm Walkthrough

- Spatially sort the reference points



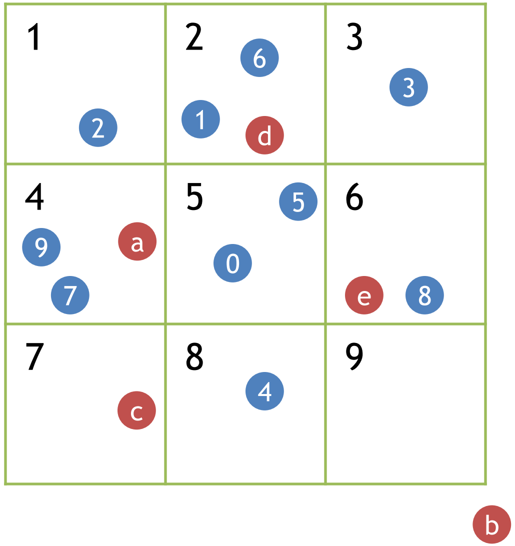
- More coherent reads!





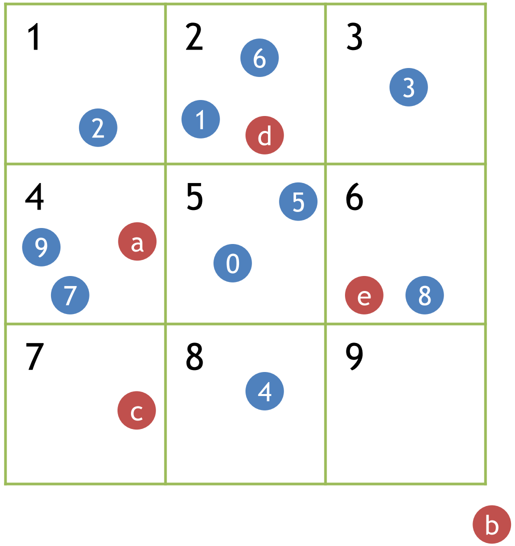
- Radix Sort [Green 2008]; Counting Sort [Hoetzlein 2014]
- 3 ~ 7x speedup compared to Pytorch3D’s KNN

Algorithm Walkthrough



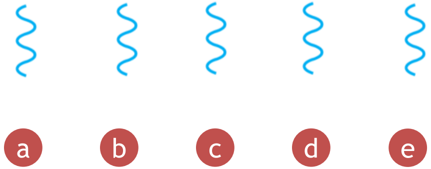
Algorithm Walkthrough

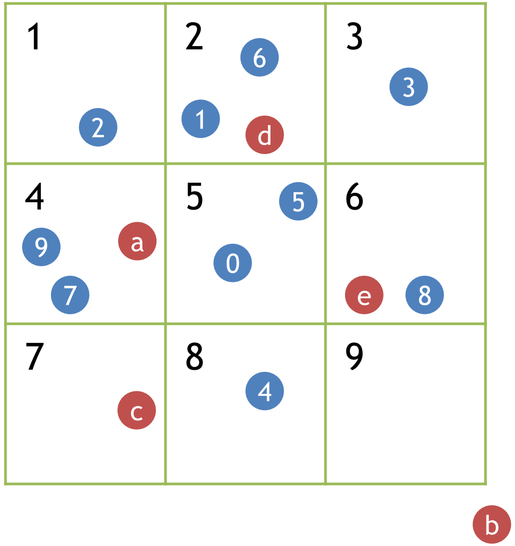
- Each thread takes a query point



Algorithm Walkthrough

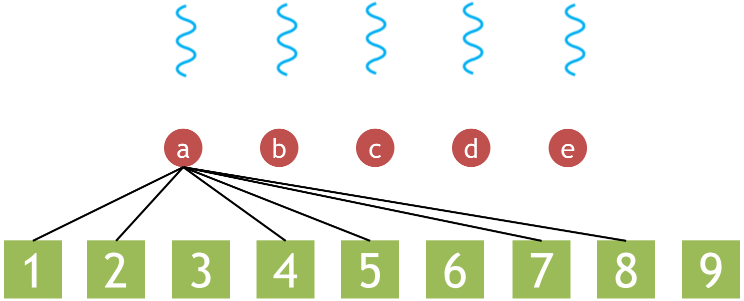
- Each thread takes a query point

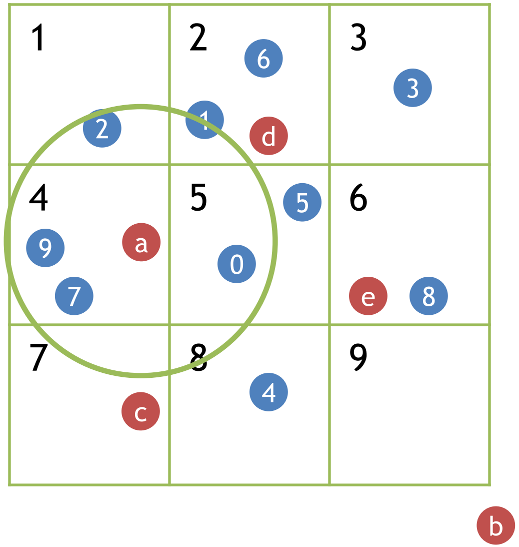




Algorithm Walkthrough

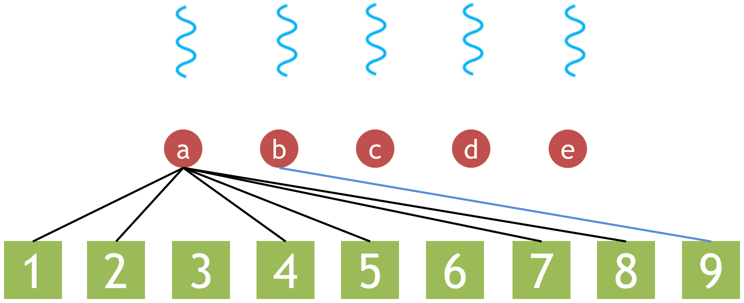
- Each thread takes a query point

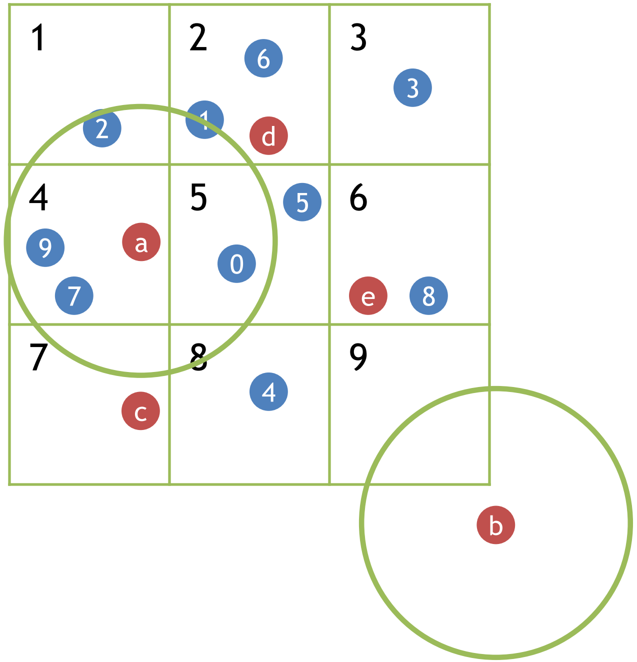




Algorithm Walkthrough

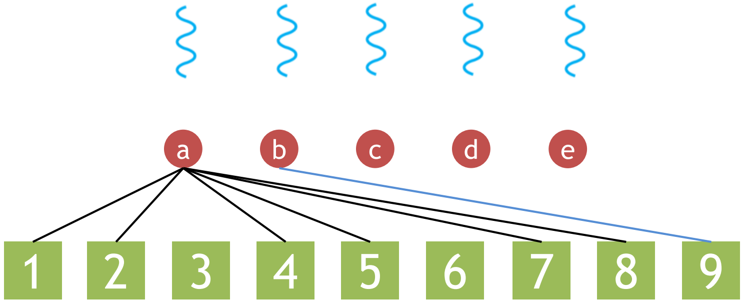
- Each thread takes a query point



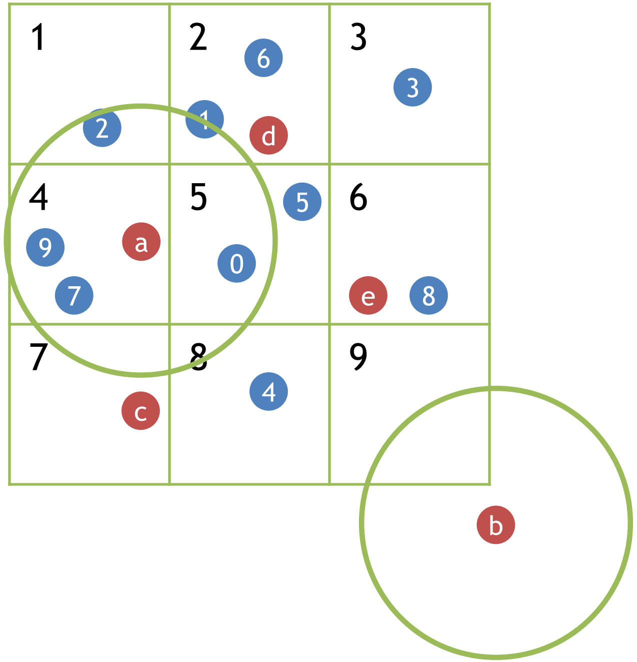


Algorithm Walkthrough

- Each thread takes a query point

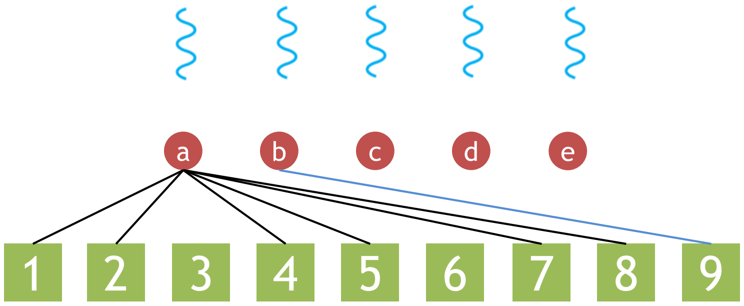


- Sort the query points

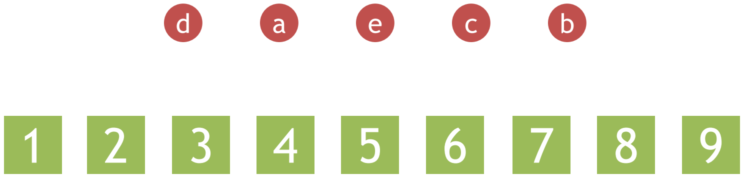


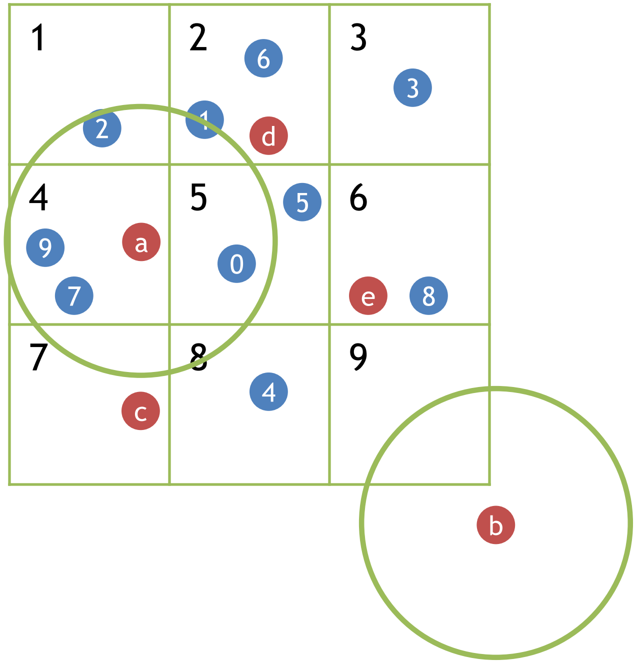
Algorithm Walkthrough

- Each thread takes a query point



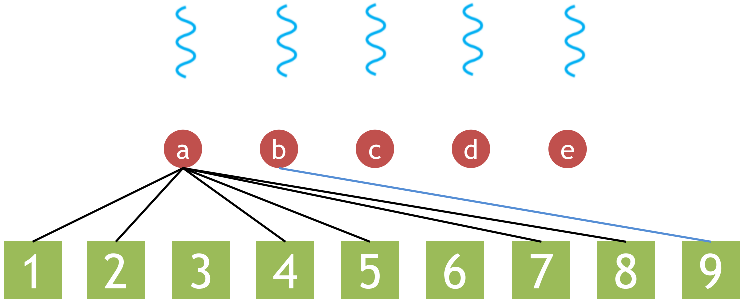
- Sort the query points



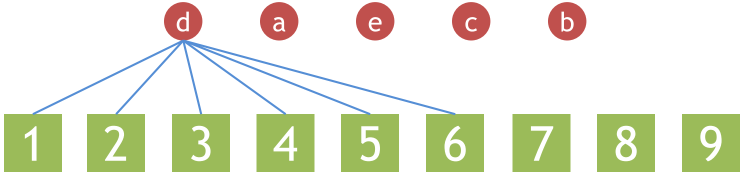


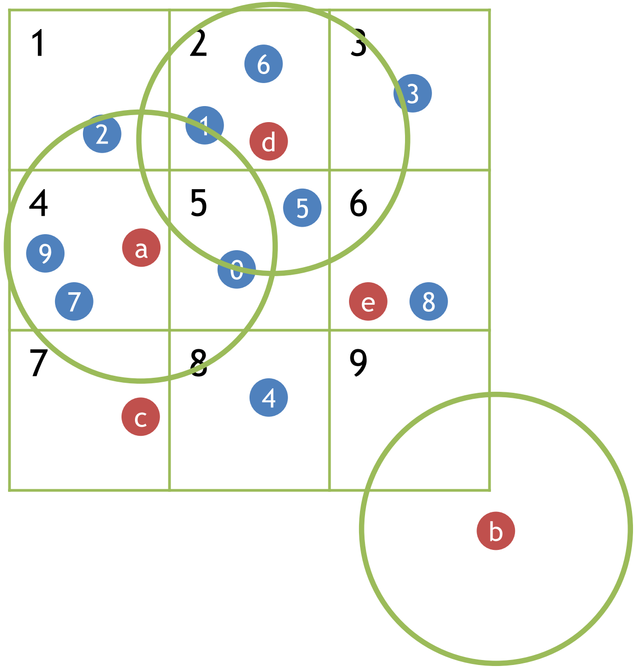
Algorithm Walkthrough

- Each thread takes a query point



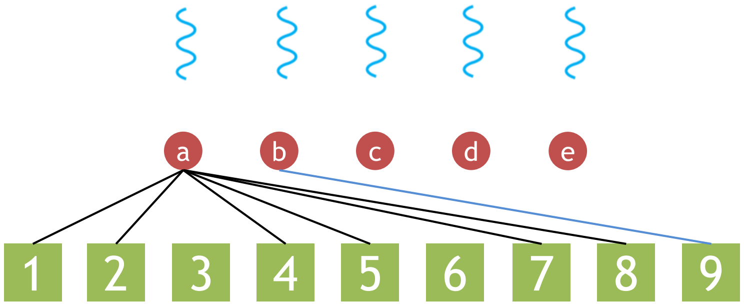
- Sort the query points



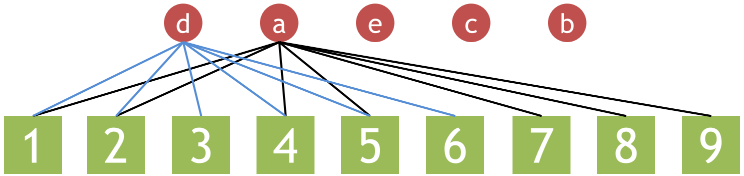


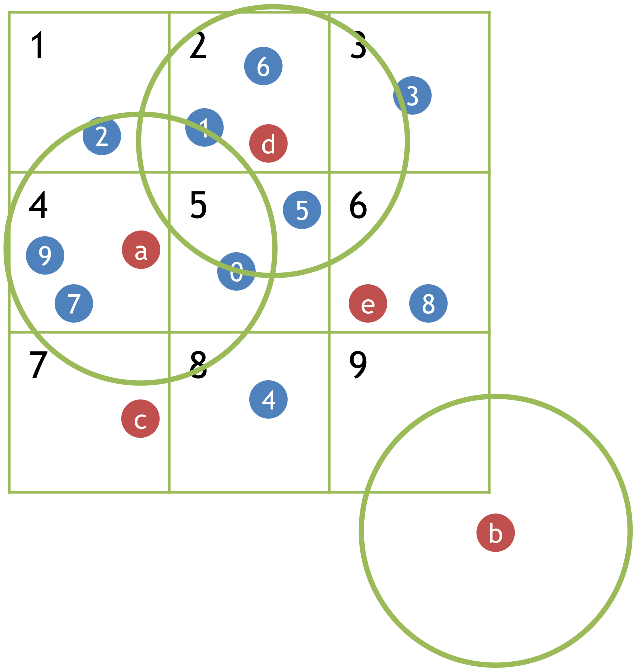
Algorithm Walkthrough

- Each thread takes a query point



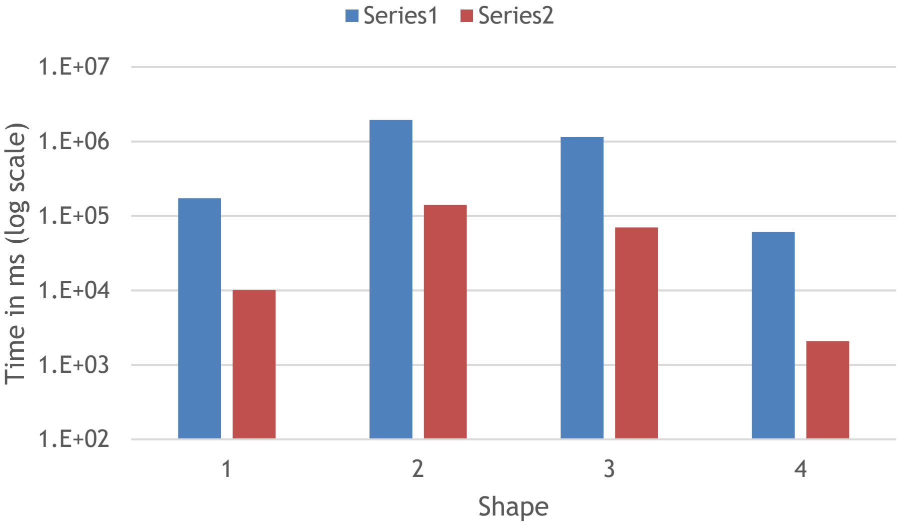
- Sort the query points





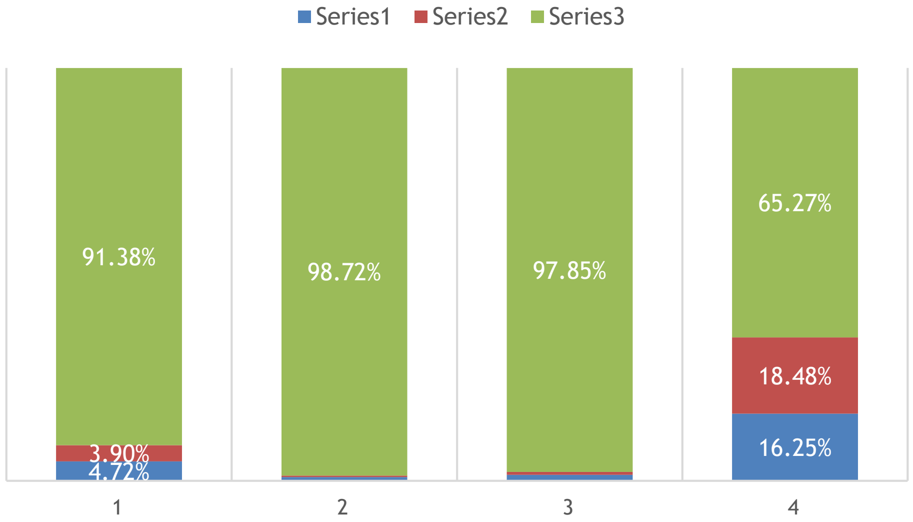
Overall Performance

- At least an order of magnitude speedup



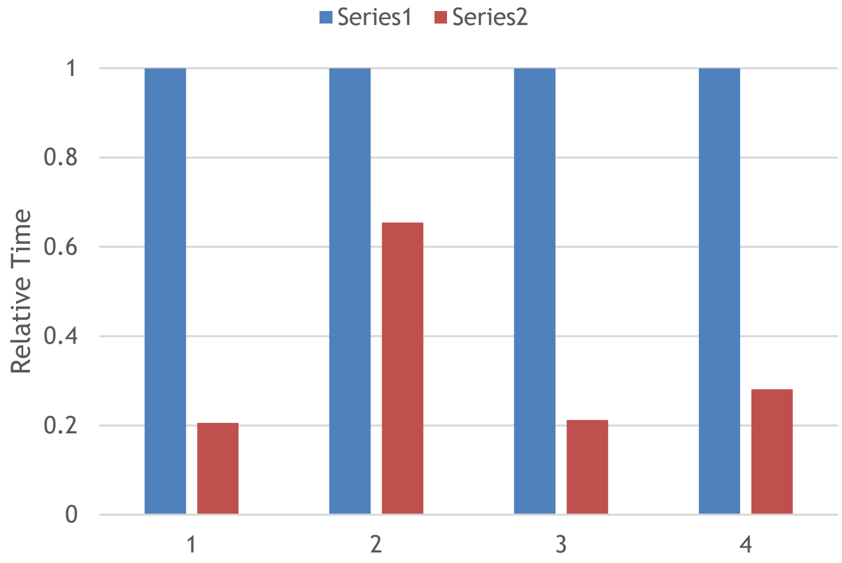
Overhead Breakdown

- Grid construction and sorting are cheap



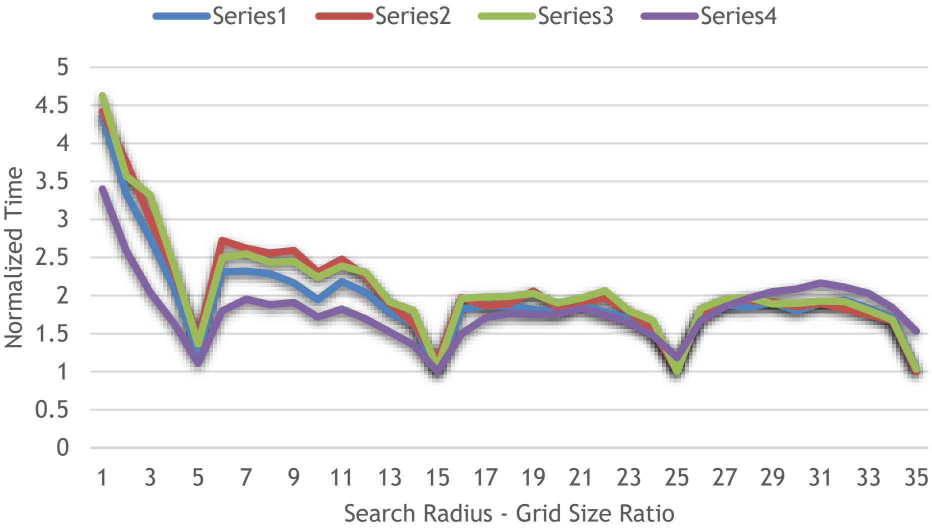
Sorting Query Points

- Sorting query points is important



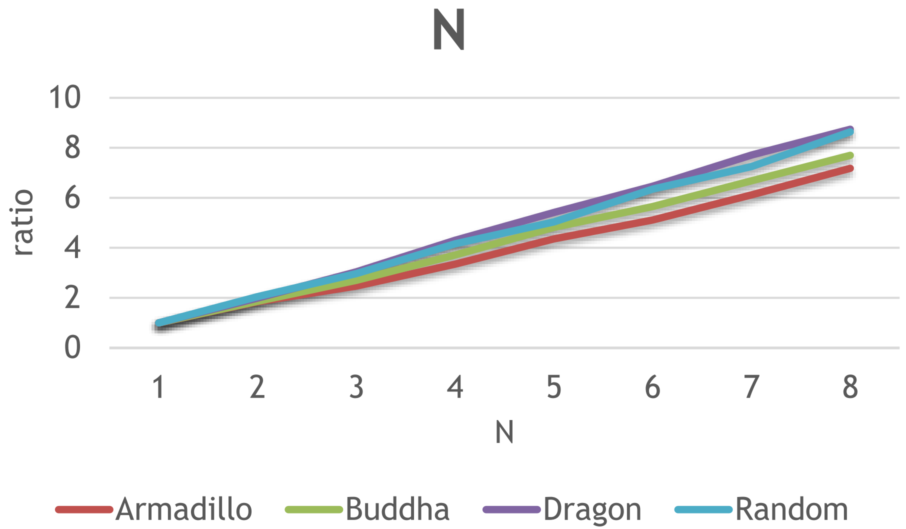
How to Choose Grid Size

- Let search radius be the multiple of grid size



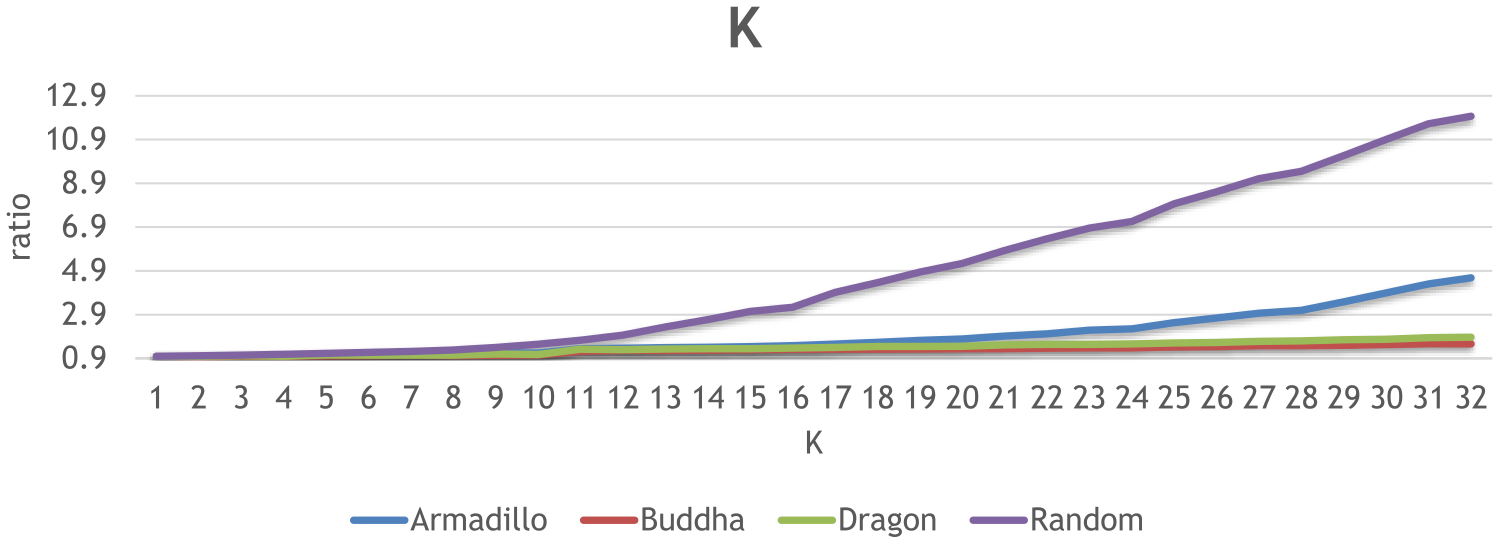
Impact of batch size N

- Runtime grows linear with N



Impact of neighbors K

- Code optimized for small K



Next Steps

- Support any dimensions
  - E.g. faster graph construction in dynamic graph CNN
- (Approximate) KNN with uniform grid
  - Setting proper search radius is not trivial

Summary

- Fixed Radius Nearest Neighbor Search
- Fast, General, Easy to use
- Publicly available at

https://github.com/lxxue/FRNN

Thank You

INTERACTIVE GEOMETRY LAB

October 20, 2020

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich