Searching Based on KD Tree
NSD 2024 Spring

Bo Han, Chen

National Yang Ming Chiao Tung University, Taiwan
bhchen312551074.cs12@nycu.edu.tw

June 3, 2024

Presentation Overview

Topic Overview

- Binary tree data structure
- Used for organizing points with partitioning
- Performing searches in multidimensional space

Development

System Architecture

- C++: KD Tree implementation
- Python: User interface
- Makefile: Build system & test
- Git: Version control
- Github Actions: Continuous integration

- Operating System: Ubuntu 22.04.4 LTS
- Compiler: GCC 10.5.0
- Python: 3.11.7
  - pybind11: 2.12.0
  - pytest: 8.2.1
  - scipy: 1.13.1

- Design requirements and corresponding interfaces
- Research related data structures and searching algorithms
- Main development phase
  - Implement data structure / searching algorithm
  - Design test cases
  - Fix bugs and refactor code

Implementation

KD Tree Data Structure

- Point
  - Coordinate
  - Dimension
- Node
  - Point
  - Axis
  - Left / Right child
  - Region
- KD Tree
  - Root node
  - Insertion / Deletion
  - nearest neighbor search
  - range search

Implementation

Searching Algorithm - Nearest Neighbor Search

- Recursive search
  - Traverse the tree to find the k nearest neighbors
- Bounded Priority queue
  - Data structure for comparing the distance between the target point and the current nearest neighbor

Implementation

Searching Algorithm - Range Search

- Recursive search
  - Traverse the tree to find the points within the given range
- Query Range (Rectangle)
  - Define the query range of each axis
  - Data structure for checking if the region of the node intersects with the given range

Implementation

User Interface

- Build the system with Makefile
- Python interface
- Perform searching with KD Tree

Implementation

Tests & Verification

- Test each data structure's operation
- Test the correctness of the searching algorithm
  - With scipy KD Tree
- Github Actions for automatic testing

Discussion & Conclusion

- Improving performance and scalability of the implementation
- Extending the functionality of the KD Tree
- Applying the KD Tree to next project

Thanks for Listening

Q & A