Performance Comparison of Hybrid Spatial Index

Data Structures - Project Proposal

Winter 2022

Introduction

A spatial index is a data structure that allows for accessing a spatial object efficiently, which is a commonly used technique in spatial databases. Without indexing, searching would require a "sequential scan" of all records in the database, thereby increasing the processing time. The Minimum Bounding Rectangle (MBR), often termed the Bounding Box (BBox), serves as an object approximation in a spatial index construction process.

Different types of Spatial Index

Various spatial indices have been developed that yield measurable performance differences. Broadly, there are two types of structures for spatial index:

1. Space Driven Structures (Fixed grid index, Quadtree, and KD-tree)
2. Data Driven Structures (R-tree and variants).

Quadtree and **KD-tree** are among the popular Space Driven Structures for spatial indexing. A quadtree is a specialized form of a grid whose resolution varies depending on the spatial objects' density. On the other hand, a KD-tree is a binary tree where each node specifies an axis and splits the set of points based on whether its coordinate along that axis is greater than or less than a particular value (coordinate median, for example).

Data-driven Structures, such as an **R-tree**, consists of a hierarchical index on the MBRs (Minimum Bounding Rectangle) of the geometries in the layer, based on the heuristic optimization of the area of MBRs in each node to improve the access efficiency.

Hybrid Spatial Index

Different spatial indexing techniques have advantages and disadvantages when compared with each other. However, some of the clear and prominent differences are that Quadtree works the best when the spatial objects are polygons as compared points. Similarly, KD-tree works the best for points, therefore considering a 2D plane with points, lines, and polygons, using the best of Quadtree and KD-tree is likely to yield better results, but with an increase in space for the index [2].

A Quad-KD tree is a hybrid tree, where the working for a polygon remains the same as the Quadtree; however, for points, a rectangular bounding box within the Quadtree has a KD-tree for faster access. Similarly, a combination of a R-tree for polygons and a KD-tree for points can be an alternative for better access time.

The research paper [1] compares the performance of a Quadtree, KD-tree, and Quad-KD tree for points and polygons and summarizes how the hybrid tree improves the access time.

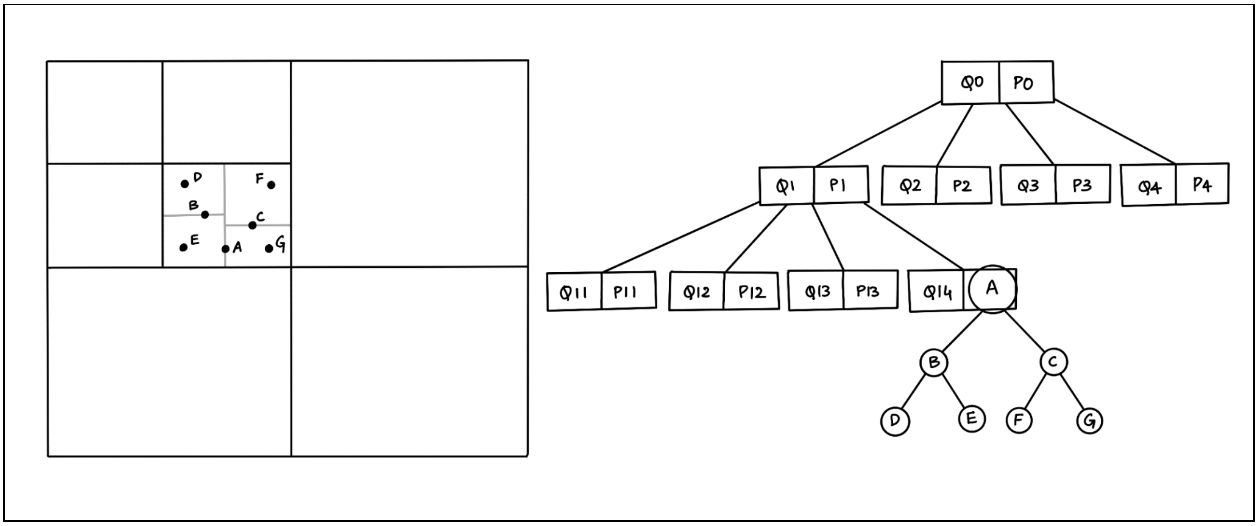

Figure 1: Hybrid Quad-KD tree; Edited version from [1]

Proposal

To implement a hybrid Quad-KD tree [1] and compare the access performance against the KD-tree and Quadtree for points and rectangles in a 2D plane. Further, extend the comparison by implementing an R-KD tree, where the R-tree is meant to be used for polygons and the KD-tree within a bounding box of the R-tree for points.

The use of the proposed hybrid trees in spatial databases, including the MySQL spatial index, is new for commercial use, translating to a lack of readily available libraries. The experiment will be conducted for points and rectangles in a 2D plane. The proposed approach uses Quadtree or R-tree for rectangles and KD-tree for points. Furthermore, the project will be implemented in Java and may use the necessary libraries for performance analysis and plotting graphs for visualization.

References

[1] Mahmood, Mahmood. (2019). A Proposed Hybrid Spatial Data Structure based on KD Tree and Quad Tree. Jokull. 69. 2-6.

[2] Bereczky, Nikolett & Duch, Amalia & Németh, Krisztián & Roura, Salvador. (2015). Quad-kd trees: A general framework for kd trees and quad trees. Theoretical Computer Science. 616. 10.1016/j.tcs.2015.12.030.

[3] Zhang, X and Du, Z. (2017). Spatial Indexing. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2017 Edition), John P. Wilson (ed). DOI: 10.22224/gistbok/2017.4.12

[4] Sun, Y.; Zhao, T.; Yoon, S.; Lee, Y. A Hybrid Approach Combining R*-Tree and k-d Trees to Improve Linked Open Data Query Performance. Appl. Sci. 2021, 11, 2405. https://doi.org/10.3390/ app11052405