Implementing an R-Tree Data Structure to Index Spatiotemporal Data

CS 123 Project Proposal
Rupesh Jeyaram

Project Description

Database systems can be used to store and query spatiotemporal data. To increase their efficiency in querying multidimensional data, an index called the R-Tree index can be implemented. This can be built on top of an existing database, without rewriting the storage layer. The overarching goal of this project is to learn more about how these R-Tree indices work by implementing the structure in a database and writing the queries to navigate it. Specifically, throughout the course of this project I would like to:

- Build an R-Tree data structure in a relational database.
- Develop a suite of stored procedures to query meaningful data out of the database via the R-Tree index.
- Explore the variants of the R-Tree data structure, and determine the most efficient and powerful variants to use for obtaining information about the changes in features over time.

Implementation Details

I will not be extending an already existing project, so there is no need to obtain code or software. However, I plan on using an already-existing database, MySQL. I am choosing this database with the motivation that this is the database used in CS 121, and it will be a familiar environment. I already have MySQL Workbench installed and ready to go.

As for the code, I plan on writing stored procedures to navigate the R-tree structure. This will be helpful in the development phase so that I can quickly write and test commands that perform the query I need within MySQL Workbench. Developing these commands within Workbench will also help me avoid bugs that could arise from syntactical errors from calling the stored procedures through Python.

On the flip side, I need to be able to visualize all of these data points and make sure the queries are indeed running properly. It will be tedious to simply examine the coordinates of resulting datapoints when I run queries such as “select all datapoints located in California”. For this, I will need to develop a Python wrapper that can interact with the database, retrieve the output of a query, and visualize the output data on a map. Conveniently, I have several Jupyter notebooks from ESE 156 (Remote Sensing) with simple mapping code that I can use with minor modifications. All the *actual* development involving the R-Tree structure will be done in MySQL.

I will run the final tests on a month's worth of Tropomi satellite data provided to me by my lab group. I believe this will be on the order of a few gigabytes. For development, however, I will use a fixed randomized sample of these datapoints.

I will use git/Github for version control. I think it will be appropriate for this project. Please let me know if there is another tool that would be more suitable. (I am not super-familiar with the different types of version control.)

Project Outline

Here are the high-level tasks that need to be completed:

Task	Allocated Time	Approx. Deadline
Obtain the Tropomi data from my lab, determine a reasonable number of records to use, randomly sample this data, and insert the result into a MySQL table.	6-9 hours	Week 2 04/08
Read and understand the details of the basic R-Tree data structure. Clarify the exact algorithms that are used to run insertions, deletions, and lookups. Determine the schema that should be implemented alongside the data such that it can be traversed through MySQL stored procedures. Implement this schema.	6-9 hours	Week 3 04/15
Write the stored procedures for inserting, deleting, and updating records in the R-Tree. This will involve a lot of debugging and testing. By this point, I will be able to clearly demonstrate how to build this project, the progress I've made, and the knowledge I've gained in studying R-Trees. **[Milestone 1]**	6-9 hours	Week 4 04/22

Load commonly used geopolitical data (county lines, country boundaries, etc.) into a separate table in MySQL. Obtain benchmarks about how fast queries run using the R-Tree index, whether they provide a significant improvement over brute-force querying, and whether these queries return accurate results. This will use the geopolitical data table from above.	10-12 hours	Week 5 04/29
Read about the different R-Tree variants provided in the Springer R-Tree book. Determine which variants would be best for the actual spatiotemporal database model to query the Tropomi data. Also research the different implementations of the basic R-Tree. Try to implement these variants, and record any improvements or interesting features. This will most likely extend into the following week.	15-20 hours	Week 6 05/06
Finalize the research poster that will be used for the Meeting of the Minds. At this point, I should have decent results (positive or negative) to include in the poster. I will be able to report on how well the R-Tree queries run, and whether the results are reasonable. **[Milestone 2]**	6-9 hours	Week 7 05/13 **MotM 05/17**
Input the rest of the Tropomi data and run more benchmarking queries. I am leaving this week as a buffer week. Interesting topics will likely spin out of trying the R-Tree variants, so I will focus on developing the best ideas.	6-9 hours	Week 8 05/20

Create a Python wrapper that interfaces with the MySQL database. Up to this point, I will have written a few scripts to quickly visualize the queries, but this wrapper should be a full-fledged access-point to the database. This would allow me to condense all the work done throughout the term into one program, where you can access everything and see some cool visualizations.	10-12 hours	Week 9 05/27
Prepare final presentation and final report. **[Milestone 3]**	6-9 hours	Week 10 & 11 06/03 & 06/10

Times Available to Meet

Mondays: Any interval other than 10-12 pm or 2-3 pm
Tuesdays: Any interval other than 1-4 pm

☺