GEOLOCATION SYSTEM TO FIND NEAREST NEIGHBOR

Anirudh . V
Department of Electronics and Communication Engineering,
Amrita School of Engineering,
Amrita Vishwa Vidhyapeetham,
Coimbatore.

Arphudha. T. K
Department of Electronics and Communication Engineering,
Amrita School of Engineering,
Amrita Vishwa Vidhyapeetham,
Coimbatore.

Ashmita. D
Department of Electronics and Communication Engineering,
Amrita School of Engineering,
Amrita Vishwa Vidhyapeetham,
Coimbatore.

Dhanu Raghavi . S
Department of Electronics and Communication Engineering,
Amrita School of Engineering,
Amrita Vishwa Vidhyapeetham,
Coimbatore.

Abstract— **Geolocation-based Nearest Neighbor Search System, addressing the challenges posed by the escalating demand for efficient spatial data retrieval in location-aware applications. The system employs advanced spatial indexing techniques and diverse distance metrics, utilizing spatial data structures like Quad-trees to optimize the retrieval of nearest neighbors. Designed for real-time queries on large-scale datasets, the system exhibits notable improvements in query response time and accuracy compared to conventional methods. Through comprehensive experiments on synthetic and real-world datasets, our approach demonstrates scalability and adaptability, positioning it as a robust solution for applications spanning location-based services to geospatial analytics. This research contributes significantly to the advancement of geospatial technologies, providing a valuable tool for fast and accurate proximity queries in diverse geographic contexts.**

Keywords— **Quadtree,Spatial Indexing,Geolocation, folium,Reward system, GUI Components, MapApp**

I. INTRODUCTION (*HEADING 1*)

In today's connected world, there's a growing demand for location-based services, especially in busy urban areas with dynamic transportation needs. This report focuses on the need for a Geolocation System that can quickly find the nearest vehicles in real-time. In cities where everything is closely linked, we require a strong system to efficiently and accurately locate nearby vehicles.

This study looks into how we can come up with and put into action an advanced Geolocation-based Nearest Vehicle Search System. This smart system uses the latest geospatial algorithms, spatial indexing tricks, and real-time data processing to tackle the challenges of finding and reaching vehicles in specific areas. The system has wide applications, including improving urban mobility, aiding emergency services, and streamlining logistics. The goal is to introduce advanced technologies that will bring about a new era of smart and responsive transportation systems.

Throughout this report, we'll explore the methods we used to develop the system, the technology that makes it work, and the potential impact it could have on how we move around in cities. By sharing insights into how feasible and effective the system is, we're contributing to making urban transportation more efficient and responsive. This report is part of the ongoing efforts to make transportation systems smarter and better suited to the needs of our evolving cities.

II. RELATED WORK

Urbanization is a rapidly growing global phenomenon that has given rise to various challenges and risks. The expansion of mega-cities has led to a surge in the importance of spatial information, which plays a critical role in ensuring sustainable development in urban areas. The use of advanced technologies like remote sensing, global positioning systems, laser scanning, and aerial surveying drones has made it easier to capture spatial data and utilize it effectively in urban planning and management.

However, the significance of spatial data intensifies as spatial data trends shift towards three-dimensional(3D) . Presently, the acquisition of 3D spatial data is more accessible across various scales and resolution, due to the advancements in technologies. Several researchers have given various solutions[1] to the problem of finding the nearest neighbor, one such key contribution was using the k-

d tree for classification and to find the nearest neighbor[2]. However, this solution can be expensive for mass datasets. Another recent approach is the use of R-Tree for spatial indexing[3] such as for much more optimized solutions. Recent work by Quadtree data structure has demonstrated the large-scale application of Quadtree in geospatial indexing[4].

As we continue to witness the rapid growth of urban areas, it is essential to develop innovative approaches to manage and analyze spatial data effectively, and quadtree data structure provides a promising solution for this challenge [5]

III. METHODOLOGY

Here we use , Folium, Geopy, and a QuadTree. The goal is to find and show the nearest vehicles around a selected city on an interactive map. Here's a

1. QuadTree Class:

- The `QuadTree` class smartly organizes locations, dividing an area into parts (quadrants) to make finding nearby points efficient.
- It's like a virtual map that helps quickly locate points close to a chosen spot.
- The method `generate_quadtree_points` randomly places points within a specified area to fill the QuadTree.

2. Spatial indexing:

- Spatial indexing is a technique used in spatial databases and geospatial applications to efficiently organize and retrieve spatial data based on their geometric properties.
- The quadtree structure, employed in this project, acts as a spatial index designed to accelerate spatial queries by organizing data in a way that reduces the search space. Spatial indexing structures are designed to manage the complexities of geometric relationships between spatial objects such as points, lines, and polygons

3. MapApp Class:

- The `MapApp` class is the main part of the program, creating a window for the user and handling crucial user interaction.
- It includes various tools like dropdown menus, buttons, and a map display.
- Users can pick a district and city, decide to use a combo box, and choose points using radio buttons

4. Initialization and UI Setup:

- The program sets up its look and feel in the `initUI` method.
- Dropdown menus are filled with options, and radio buttons are created for selecting different points.

5. Show Map Functionality:

- When users press the "Show on Map" button, the `show_map` method is activated.
- It finds the chosen city's coordinates using the `Nominatim` geocoder.
- A Folium map is built with markers for the city and randomly placed points within a set area.
- A QuadTree is used to efficiently find the five closest points to the city.
- The map is saved and displayed, and information about the nearest points is shown.

6. Select Point Functionality:

- Clicking the "Select Point" button triggers the `select_point` method.
- It figures out which radio button is chosen and prints details about that selected point.

7. reward system:

- The reward system in this project is related to a discount mechanism based on user interactions with the map and the selection of points
- The discount_counter variable keeps track of the number of discounts a user has received.
- It is initially set to 0.
- The stack is used to count the number of point selections.
- The discount logic is triggered when the number of selections reaches a multiple of 5.
- The discount counter is checked, and if it reaches a specific threshold, it is reset to 0.

8. Timer Execution:

- Integrate QTimer which enables periodic updates for refreshing random points on the map.
- Functions like ‘select_point’, ‘refresh_random_points’, and ‘refresh_random_points_after_selection’ handle various aspects of the user interaction flow, ensuring a dynamic experience.

9. Execution:

- The script functions as a standalone program, presenting a window for the users to interact with the map, select points, and explore the closest vehicles in the vicinity.

This program functions as a virtual guide, allowing you to choose a location on the map. The tool not only displays nearby places but also highlights the closest vehicles in your vicinity on the map. It provides an interactive and informative experience, helping users explore their surroundings with ease.

1V. RESULTS

Quadtree is used for spatial indexing and efficient searching of nearest points. This project involves a Geospatial Map Application with features such as displaying maps, generating random points, finding the nearest points, and implementing a reward system.

The map rendering time is fast and consistent, even with a large number of points, providing a smooth user experience. The application accurately places cities on the map, with minimal deviation from their actual locations.

Fast and accurate retrieval of nearest points, providing relevant information to the user. Consistent and timely updates of random points, enhancing the dynamic nature of the map.

Users engage with the reward system; discounts are correctly applied. The discount logic effectively encourages user participation, which leads to increased usage and satisfaction.

V.CONCLUSION

In conclusion, this geolocation-based interactive mapping application successfully combines various technologies to provide an engaging and informative experience for users. The integration of PyQt5 for the graphical user interface, Folium for map visualization, and a QuadTree data structure for efficient spatial indexing contribute to the creation of a robust and user-friendly tool.

The project's QuadTree implementation allows for the efficient organization and retrieval of spatial data, enhancing the application's ability to find and display the nearest points around a selected city. This spatial indexing mechanism, coupled with the power of the Folium library, results in dynamic and visually appealing maps that showcase the selected city and its surroundings.

The user interface, designed with PyQt5 widgets, offers a seamless and user-friendly. Users can easily select a district, enter or choose a city, and visualize relevant points on the map. The inclusion of features such as radio buttons for point selection and a web browser component for map display further enhances the interactivity of the application.

The project also demonstrates proficiency in utilizing geolocation services, as exemplified by the integration of the Nominatim geocoder for obtaining precise coordinates based on user input. The application's ability to generate random points, populate the QuadTree, and identify the nearest points to a chosen city showcases the successful implementation of key functionalities.

Overall, this project serves as an effective demonstration of integrating spatial data structures, mapping libraries, and user interface components to create a practical and visually appealing geolocation system. Its versatility and simplicity make it a valuable tool for individuals interested in exploring and discovering points of interest within cities. As technology continues to advance, the project lays a foundation for further enhancements and applications in the field of geospatial data analysis and visualization.

VI. FUTURE SCOPE

For the future scope of this project, many enhancements and expansions can be considered. More features can be added to improve the user experience, like route planning, information overlays on the map, and search capabilities. Customization options can be added. The project combines several different technologies, such as Folium for web-based mapping, PyQt5 for the desktop GUI, and geocoding services (like Nominatim) for geographical coordinates. This integration demonstrates adaptability and the capacity to use a range of tools to provide a coherent solution.

The project, with its map-centric features, can serve as a foundation for various business applications. For example, it could be extended for use in location-based marketing, logistics optimization, or territory planning. Open-sourcing or engaging with a community of developers and users can foster collaboration and lead to continuous improvement.

The project can be adapted to mobile platforms, catering to the growing trend of mobile applications. This adaptability is crucial for reaching a broader audience and aligning with the increasing use of mobile devices.

REFERENCES

[1] Azri, Suhaibah & Ujang, Uznir & Anton, François & Mioc, Darka & Rahman, Alias. (2013). Review of Spatial Indexing Techniques for Large Urban Data Management.

[2] W. Hou, D. Li, C. Xu, H. Zhang and T. Li, "An Advanced k Nearest Neighbor Classification Algorithm Based on KD-tree," *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, Chongqing, China, 2018, pp. 902-905, doi: 10.1109/IICSPI.2018.8690508.

[3] S. Yi, H. Jung, J. P. Park and Y. D. Chung, "On processing view field nearest neighbor queries on the R∗-tree," *2011 International Conference on Electrical and Control Engineering*, Yichang, China, 2011, pp. 4838-4841, doi: 10.1109/ICECENG.2011.6058108.

[4] K. Aizawa and S. Tanaka, "A Constant-Time Algorithm for Finding Neighbors in Quadtrees," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no.

[5] A. M. Karadeniz, C. Hajdu, L. T. Kóczy and Á. Ballagi, "Robot environment representation based on Quadtree organization of Fuzzy Signatures," 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2021, pp. 509-514, doi: 10.1109/SACI51354.2021.9465566.

[6] A. M. Karadeniz, C. Hajdu, L. T. Kóczy and Á. Ballagi, "Robot environment representation based on Quadtree organization of Fuzzy Signatures," 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2021, pp. 509-514, doi: 10.1109/SACI51354.2021.9465566.

[7] A. M. Karadeniz, C. Hajdu, L. T. Kóczy and Á. Ballagi, "Robot environment representation based on Quadtree organization of Fuzzy Signatures," 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2021, pp. 509-514, doi: 10.1109/SACI51354.2021.9465566.