FINDING THE CLOSEST PATH TO POINT OUT THE NEIGHBOUR WITH KEYWORD

YASARAM GANESH
M. Tech, CSE
Asst. Professor, Cristu Jyothi Institute of Technology under JNTU H.
Mail Id: yasarapuganesh@gmail.com

DHARAVATH CHAMPLA
M. Tech, CSE
Asst. Professor, Ashoka Institute of Engineering and Technology under JNTUH.
Mail Id: Champla.805@gmail.com

ABSTRACT:

Conventional spatial queries, such as range search and nearest neighbor retrieval, involve only conditions on objects' geometric properties. Today, many modern applications call for novel forms of queries that aim to find objects satisfying both a spatial predicate, and a predicate on their associated texts. For example, instead of considering all the restaurants, a nearest neighbor query would instead ask for the restaurant that is the closest among those whose menus contain "steak, spaghetti, brandy" all at the same time. Currently the best solution to such queries is based on the IR2-tree, which, as shown in this paper, has a few deficiencies that seriously impact its efficiency. Motivated by this, we develop a new access method called the spatial inverted index that extends the conventional inverted index to cope with multidimensional data, and comes with algorithms that can answer nearest neighbor queries with keywords in real time. As verified by experiments, the proposed techniques outperform the IR2-tree in query response time significantly, often by a factor of orders of magnitude.

1. INTRODUCTION:

A spatial database manages multidimensional objects (such as points, Rectangles, etc.), and provides fast access to those objects based on different selection criteria. The importance of spatial databases is reflected by the convenience of modeling entities of reality in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are often represented as points in a map, while larger extents such as parks, lakes, and landscapes often as a combination of rectangles. Many functionalities of a spatial database are useful in various ways in specific contexts. For instance, in a geography information system, range search can be deployed to find all restaurants in a certain area, while nearest neighbor retrieval can discover the restaurant closest to a given address. Today, the widespread use of search engines has made it realistic to write spatial queries in a brand new way. Conventionally, queries focus on objects' geometric properties only, such as whether a point is in a rectangle, or how close two points are from each other. We have seen some modern applications that call for the ability to select objects based on both of their geometric coordinates and their associated texts. For example, it would be fairly useful if a search engine can be used to find the nearest restaurant that offers "steak, spaghetti, and brandy" all at the same time. Note that this is not the "globally" nearest restaurant (which would have been returned by a traditional nearest neighbor query), but the nearest restaurant among only those providing all the demanded foods and drinks.

In this paper, we design a variant of inverted index that is optimized for multidimensional points, and is thus named the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. Meanwhile, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead. As a result, it offers two competing ways for query processing. We can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also leverage the R-trees to browse the points of all relevant lists in ascending order of their distances to the query point. As demonstrated by experiments, the SI-index significantly outperforms the IR 2 -tree in query efficiency, often by a factor of orders of magnitude.

2. System Analysis

a. Existing System:

Spatial queries with keywords have not been ex-tensively explored. In the past years, the community has sparked enthusiasm in studying keyword search in relational databases. It is until recently that attention was diverted to multidimensional data. Existing works mainly focus on finding top-k Nearest Neighbours, where each node has to match the whole querying keywords .It does not consider the density of data objects in the spatial space. Also these methods are low efficient for incremental query.

We have finished explaining how to build the leaf nodes of an R-tree on an inverted list. As each leaf is a block, all the leaves can be stored in a blocked SI-index as described in Section 6.1. Building the non leaf levels is trivial, because they are invisible to the merging-based query algorithms, and hence, do not need to preserve any common ordering. We are free to apply any of the existing R-tree construction algorithms. It is noteworthy that the non leaf levels add only a small amount to the overall space overhead because, in an R-tree, the number of non leaf nodes is by far lower than that of leaf nodes.

We have finished explaining how to build the leaf nodes of an R-tree on an inverted list. As each leaf is a block, all the leaves can be stored in a blocked SI-index as described in Section 6.1. Building the non leaf levels is trivial, because they are invisible to the merging-based query algorithms, and hence, do not need to preserve any common ordering. We are free to apply any of the existing R-tree construction algorithms. It is note worthy that the non leaf levels add only a small amount to the overall space overhead because, in an R-tree, the number of non leaf nodes is by far lower than that of leaf nodes.

b. Proposed system:

A spatial database manages multidimensional objects (such as points, rectangles, etc.), and provides fast access to those objects based on different selection criteria. The importance of spatial databases is reflected by the convenience of modeling entities of reality in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are often represented as points in a map, while larger extents such as parks, lakes, and landscapes often as a combination of rectangles. Many functionalities of a spatial database are useful in various ways in specific contexts. For instance, in a geography information system, range search can be deployed to find all restaurants in a certain area, while nearest neighbor retrieval can discover the restaurant closest to a given address.

Furthermore, as the SI-index is based on the conventional technology of inverted index, it is readily incorporable in a commercial search engine that applies massive parallelism, implying its immediate industrial merits.

3. IMPLIMENTATION

a. **Registration:** In this module an User have to register first, then only he/she has to access the data base.

b. **Login:** In this module, any of the above mentioned people have to login, they should login by giving their email id and password.

c. **Hotel_Registration:** In this module Admin registers the hotel along with its famous dish. Also he measures the distance of the corresponding hotel from the corresponding source place by using spatial distance of Google map

d. **Search Techniques:** Here we are using two techniques for searching the document 1)Restaurant Search,2)Key Search.

   i. **Key Search:** It means that the user can give the key in which dish that the restaurant is famous for .This results in the list of menu items displayed.

   ii. **Restaurant Search:** It means that the user can have the list of restaurants which are located very near. List came from the database.

e. **Map_View:** The User can see the view of their locality by Google Map(such as map view, satellite view) .

f. **Distance_Search:** The User can measure the distance and calculate time that takes them to reach the destination by giving speed. Chart will be prepared by using these values. These are done by the use of Google Maps.

4. CONCLUSION

We have seen plenty of applications calling for a search engine that is able to efficiently support novel forms of spatial queries that are integrated with keyword search. The existing solutions to such queries either incur prohibitive space consumption or are unable to give real time answers. In this paper, we have remedied the situation by developing an access method called the spatial inverted index (SI-index). Not only that the SI-index is fairly space economical, but also it has the ability to perform keyword-augmented nearest neighbor search in time that is at the order of dozens of milli-seconds. Furthermore, as the SI-index is based on the conventional technology of inverted index, it is readily incorporable in a commercial search engine that applies massive parallelism, implying its immediate industrial merits.

5. REFERENCES:

1. *S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: A system for keyword-based search over relational databases. In Proc. of International Conference on Data Engineering (ICDE), pages 5–16, 2002.*
2. *N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In Proc. of ACM Management of Data (SIGMOD), pages 322–331,1990.*
3. *G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudar- shan. Keyword searching and browsing in databases using banks. In Proc. of International Conference on Data ngineering (ICDE), pages 431–440, 2002.*
4. *X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, and M. L. Yiu. Spatial keyword querying. In ER, pages 16–29, 2012.*
5. *X. Cao, G. Cong, and C. S. Jensen. Retrieving top-k prestige-based relevant spatial web objects. PVLDB, 3(1):373–384, 2010.*
6. *X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. Collective spatial keyword querying. In Proc. of ACM Management of Data (SIG-MOD), pages 373–384, 2011.*
7. *B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal. The bloomier filter: an efficient data structure for static support lookup tables. In Proc. of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 30–39, 2004.*

8. *Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In Proc. of ACM Management of Data (SIGMOD), pages 277–288, 2006.*
9. *E. Chu, A. Baid, X. Chai, A. Doan, and J. Naughton. Combining keyword search and forms for ad hoc querying of databases. In Proc. of ACM Management of Data (SIGMOD), 2009.*
10. *G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. PVLDB, 2(1):337–348, 2009.*