International Research Journal of Computer Science (IRJCS)
Issue 05, Volume 4 (May 2017) SPECIAL ISSUE
ISSN: 2393-9842
www.irjcs.com

SURVEY ON FAST SEARCH IN SPATIAL DATABASE USING KEYWORD

Smt. Veena .G, Anil kumar B M, Bharath Babu N, Manjunath K
Department of Computer Science and Engineering,
Vemana Institute of Technology, Bangalore, India

Abstract*-- The main idea of nearest neighbour search is to get results (locations) for the query constraints given. Spatial database is used, it manages multi dimensional objects (such as points, rectangles etc.) And allows fast access to those objects based on queries. Data structures like R-tree and IR2-tree for spatial access method is used. For example, if we need to find hotels which are near with food items such as idly and Dosa, The data structure with the help of spatial database finds all the nearest hotels within a specific range from current location and lists them in order having those required facilities. In the existing system IR2-tree is used, which has few deficiencies that can impact its efficiency, for proposed system, a new access method called spatial inverted index is used that is an extension of conventional inverted index that can answer nearest neighbor queries with keywords in real time.

***Keywords*: spatial database, spatial inverted index, R-tree, IR2- tree.**

I. 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.

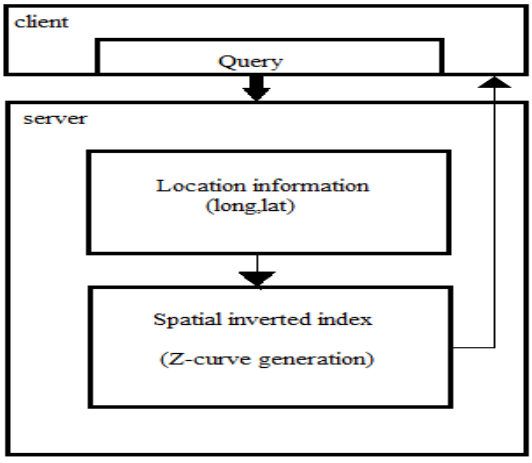

Figure 1: System Architecture showing interaction between client/user and server/admin.

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. The above figure is client server architecture where the client enters the keyword in dialog box. Server then with the help of spatial database gives back the nearby locations.

II. LITERATURE SURVEY

[1] A system for keyword-based search over relational databases. DB Xplorer is a system that enables keyword-based search in relational databases. It has been implemented using a relational database and web server and allows users to interact via a browsers (chrome, mozilla,etc.). Our key contributions are as follows: (i) we formally define the problem of keyword querying over relational databases that lack a-priori access to the database instance; (ii) we introduce the notion of a weight as a measure of the likelihood that the semantics of a keyword are represented by a database structure, i.e., a table, an attribute, or a value. We further distinguish the weights to intrinsic contextual, to emphasize that this likelihood does not depend only on the meaning of the keyword semantics when the keyword is considered in isolation, but also on the way the semantics of the remaining, especially the neighboring, keywords are represented in the data. (iii) we extend and exploit the Hungarian algorithm to develop a technique for the systematic computation of the contextual weights that leads into to the generation and ranking of the different interpretations of a keyword query in terms of SQL; finally, (iv) we experimentally evaluate our approach on real application scenarios

[2] The R*-tree: An efficient and robust access method for points and rectangles. The R-tree is the most popular access methods for rectangles it is based on the heuristic optimization of the area. the R*-tree which incorporates a combined optimization of area, margin and overlap of each enclosing rectangle in comparison it turned out that the R*-tree clearly outperforms the existing R-tree variants Guttman's linear and quadratic R-tree and Greene's variant of the R-tree. R*-tree holds for different types of queries and operations, such as map overlay. for both rectangles and multidimensional points m all experiments From a practical point of view the R*-tree is very attractive because of the following two reasons 1 It efficiently supports point and spatial data at the same time and 2 Its implementation cost is only slightly higher than that of other R-trees.

[3] Keyword search on spatial databases: The problems of nearest neighbor search on spatial data and keyword search on text data have been extensively studied separately. However, there is no efficient method to answer spatial keyword queries. In this reference paper an efficient method to answer top-k spatial keyword queries is used. So we have introduced an indexing structure called IR2 -Tree (Information Retrieval R-Tree) which combines an R-Tree with signature files. An algorithm is used to construct and maintain an IR 2 -Tree, and use it to answer top-k spatial keyword queries and are experimentally compared to current methods which shows better performance and scalability.

III. 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 neighbour search in time that is at the order of dozens of millli-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.

REFERENCES

[1] Yufei Tao and Cheng Sheng: "Fast Nearest Neighbor Search with Keywords".
[2] S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: "A system for keyword-based search over relational databases".
[3] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The R*-tree: "An efficient and robust access method for points and rectangles".
[4] I. D. Felipe, V. Hristidis, and N. Rishe:" Keyword search on spatial databases".

