ISSN: 2277-9655
Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

FAST NEAREST NEIGHBOUR SEARCH WITHIN ARBITRARY SUBSPACES

H.Praveena*, M.Saravanan

M.Tech II Year Department of Information Technology, Ganadipathy Tulsi's Jain Engineering College, Kaniyambadi, Vellore-632102.

Associate Professor Department of Information Technology, Ganadipathy Tulsi's Jain Engineering College, Kaniyambadi, Vellore-632102

DOI: 10.5281/zenodo.51024

ABSTRACT

A spatial database is a database that is optimized to store and query data that represents objects defined in a geometric space. Many spatial queries involve only conditions on objects' geometric properties for search but the case is modern applications are in need of queries that aim to find objects satisfying both a spatial predicate, and a predicate on their associated texts. There are some straight forward approach which first deals with spatial predicate and then on the non-spatial predicate as a process of reduction. But these approaches are not good with complex queries. So in this project propose an inverted index called as spatial inverted index (SI-index) which converts the multi-dimensional data objects into ids this reduces the required space for processing which is the main disadvantage of the existing systems. This project perform location querying in more arbitrary subspaces for example searching hospital with more information's like heart specialist and check rooms availability etc.

KEYWORD: spatial inverted index (SI-index), geometric properties.

INTRODUCTION

A spatial database is database. The importance of spatial databases is reflected by the conveniene of modeling entities of reality in a geo-metric manner. For example, locations of restaurants, hotels, hospitals and soon are of represented as points in a map, while larger extents such as parks, lakes, and scapes 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 restaurant's in a certain area, while nearest neighbor retrieval can discover the restaurant closest to a given address.Conventionally, queries focus on objects geometric properties only, such as whether a point is in a rectangle, or how closet 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 hospital that offers "rooms, beds"all at the sametime.

PROBLEM DEFINITIONS

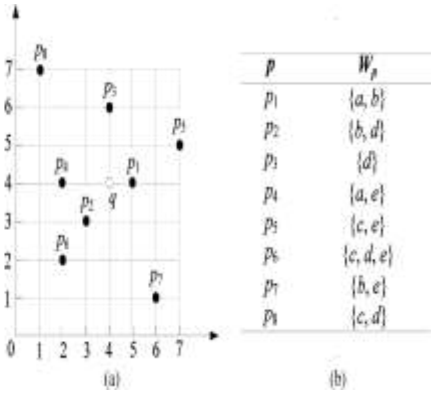
Let P be a set of multidimensional points. A sourgoalisto combine keyword search with the existing location-finding services on facilities such as hospitals, restaurants, hotels, etc., we will focus on dimensionality, but our technique can be extended to arbitrary dimensionalities with no technical obstacle. We will assume that the points in P have integer coordinates, such that each coordinate ranges in ½0;t, where t is a large integer.This is not as restrictive as it may seem, because even if one would like to insist on real-valued coordinates, the set of different coordinates represent able under a space limit is still finite and enumerable; therefore, we could as well convert everything to integers with proper scaling.

As with, each point p2 P is associated with a set of words, which is denoted as Wp and termed the document of p. For example, if p stands for a restaurant, Wp can be its menu, or if p is a hotel, Wp can be the description of its services and facilities, or if p is a hospital, Wp can be the list of its outpatient specialities. It is clear that Wp may potentially contain numerous words.

Traditional nearest neighbor search returns the datapoint closest to a query point. Following we extend the problem to include predicates on the more amount of content to the process so that only few process only accept the scheme

The rest of the paper is organized as follows. Section 2 defi the problem studied in this paper formally. Sec- tion 3 surveys the previous work related to ours. Section 4 gives an analysis that reveals.Section 6 proposes the SI-idnex, and establishes its theo- retical properties. Section 7 evaluates our techniques with extensive experiments.

objects' texts. Formally, in our context, a nearest neighbor (NN) query specifies a point q



P	W_p
p_1	$\{a, b\}$
p_2	$\{b, d\}$
p_3	$\{d\}$
p_4	$\{a, e\}$
p_5	$\{c, e\}$
p_6	$\{c, d, e\}$
p_7	$\{b, e\}$
p_8	$\{c, d\}$

Fig.1. (a) Shows the locations of points and (b) gives their associated texts.

In other words, Pq is the set of objects in P whose documents contain all the keywords in Wq. In the case where Pq is empty, the query returns nothing. The problem definition can be generalized to k nearest neighbor (kNN) search, which finds the k points in Pq closest to q; if Pq has less than k points, the entire Pq should be returned.

For example, assume that P consists of eight points whose locations are as shown in Fig.1a (t and their documents are given in Fig.1b. Consider a query point q at the white dot of Fig. 1a with the set of keywords Wq¼fc; dg. Nearest neighbor search find sp6, noticing that all points close r to q than p6 are missing either the query keyword cord. If k¼2 nearest neighbors are wanted, p8 is also returned in addition. The result is still fp6;p8 geven if k increase higher, because only two objects have the keywords c and d at the same time.

We consider that the dataset does not fit in memory, and needs to beind exed by efficient access methods in order to minimize the number of I/O s in answering a query.

RELATED WORK

Further we discuss on IR2-Tree, Solution based on inverted index and architecture diagram.

3.1The IR2-Tree

As mentioned before, the IR2-tree combines the R-tree with signature files. Next, we will review what is a signature file before explaining the details of IR2-trees. Our discussion assumes the knowledge of R-trees and the best-first algorithm for NN search, both of which are well known techniques in spatial databases.

word	*hashed bit string*
a	00101
b	01001
c	00011
d	00110

e	10010

Fig.2. Example of bit string computation with *l*¼5 and *m*¼2.

instantiations. It is designed to perform membership tests. It determine whether a query word of words. SC is conservative, in the sense that if it says “no”, then definitely do the project.

In the context of works in the same way as the classic technique of bloom filter. In preprocessing, it build a bit signature of length. W by hashing each word in W to a string of bits, and then taking the disjunction of all bit strings. To illustrate, denote by bit string of a word.

So we convert the bit string into common inverted index the solution are easily possible content of the computation scheme. In geographic web search, each webpage is assigned a geographic region that is pertinent to the webpage’s con- tents. It will communicate the corresponding problem definition` According to the experiments of [12], when Wq has only a single word, the performance of I-index is very bad, which is expected because everything in the inverted list of that word must be verified. Interestingly, as the size of Wq increases, the performance gap between I-index and IR2- tree keeps narrowing such that I-index even starts to outper- form IR2-tree at jWq j ¼ 4. This is not as surprising as it may seem. As jWq j grows large, not many objects need to be veri- fied because the number of objects carrying all the query keywords drops rapidly. On the other hand, at this point an advantage of I-index starts to pay off. That is, scanning an inverted list is relatively cheap because it involves only sequential I/Os,1 as opposed to the random nature of accessing the nodes of process.

3.2Solutions Based on Inverted Indexes

Inverted indexes (I-index) have proved to bean effective access method for keyword based document retrieval. In the spatial context, nothing prevents us from treating the Note that the list of each word maintains a sorted order of point ids, which provides considerable convenience in query processing by allowing an efficient merge step. For example, assume that we want to find the points that have words c and d. Let us Now that we have successfully would be natural to think about using a 3D SFC to cope with ids too. As far as space reduction is concerned, this 3D approach may not a bad solution. The problem is that it will destroy the locality of the points in their original space. Specifically, the converted values would no longer preserve the spatial proximity of the points, because ids in general have nothing to do with coordinates.

p_6	p_2	p_8	p_4	p_7	p_1	p_3	p_5
12	15	23	24	41	50	52	59

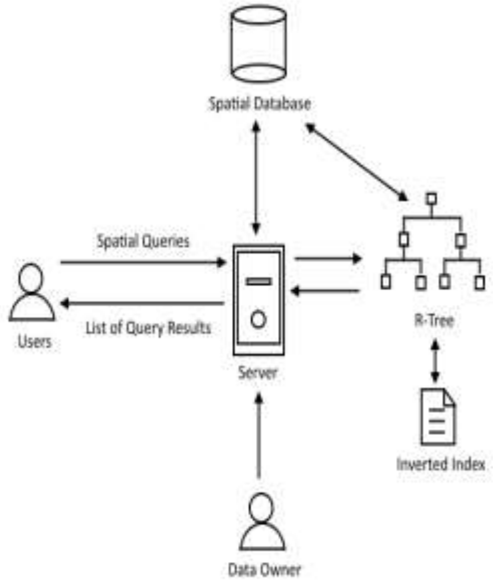
word	***inverted list***
a	p_1 p_4
b	p_1 p_2 p_7
c	p_5 p_6 p_8
d	p_2 p_3 p_6 p_8
e	p_4 p_5 p_6 p_7

Fig.4. Example of an inverted index.

Text description Wp of a point p as a document, and then, building an index. Fig.4 illustrates the index for the data set of Fig.1. Each word in the vocabulary has an inverted list, enumerating the ids of the points that have the word in their documents.

Note that the list of each word maintains a sorted order of point ids, which provides considerable convenience in query processing by allowing an efficient merge step. For example, assume that we want to find the points that have words c and d. This is essentially to compute the intersection of the two words 'inverted lists. It turns out that the complexity in the above lemma is already the lowest in the worst case, and no storage scheme is able to do any better. Remember that an SI-index is no more than a com- pressed version of an ordinary inverted index with coordinates embedded, and hence, can be queried in the same way as described. If one thinks about the purposes of having an id, it will be clear that it essentially provides a token for us to retrieve (typically, from a hash table) the details of an object, e.g., the text description and/or other attribute values. Further- more, in answering a query.Although keyword search has only started to receive attention in spatial databases, it is already thoroughly stud- ied in relational databases, where the objective is to enable a querying interface that is similar to that of search engines

3.3Architecture diagram



PROPOSED SYSTEM

- I 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 inverted index with small extra space to a delicate compact storage scheme.
- Compression is used to reduce the size of an inverted index in the conventional context where each inverted list contains only ids.

CONCLUSIONS

- Thus the technique is developed by the inverted index, it shows the short coming of the previous effects.
- Then the project is concluded to nearest hospitals for particular incidents.
- And find the specialist in that hospital and also the room availability.

REFERENCES

[1] S. Agrawal, S. Chaudhuri, and G. Das, “Dbxplorer: A System for Keyword-Based Search over Relational Databases,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 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 Rec- tangles," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 322-331, 1990.

[3] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, "Keyword Searching and Browsing in Databases Using Banks," Proc. Int'l Conf. Data Eng. (ICDE), pp. 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," Proc. 31st Int'l Conf. Conceptual Modeling (ER), pp. 16-29, 2012.

[5] X. Cao, G. Cong, and C.S. Jensen, "Retrieving Top-k Prestige- Based Relevant Spatial Web Objects," Proc. VLDB Endowment, vol. 3, no. 1, pp. 373-384, 2010.

[6] X. Cao, G. Cong, C.S. Jensen, and B.C. Ooi, "Collective Spatial Keyword Querying," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 373-384, 2011.

[7] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal, "The Bloomier Fil- ter: An Efficient Data Structure for Static Support Lookup Tables," Proc. Ann. ACM-SIAM Symp. Discrete Algorithms (SODA), pp. 30- 39, 2004.

[8] Y.-Y. Chen, T. Suel, and A. Markowetz, "Efficient Query Process- ing in Geographic Web Search Engines," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 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," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2009.

[10] G. Cong, C.S. Jensen, and D. Wu, "Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects," PVLDB, vol. 2, no. 1, pp. 337- 348, 2009.

[11] C. Faloutsos and S. Christodoulakis, "Signature Files: An Access Method for Documents and Its Analytical Performance Eval- uation," ACM Trans. Information Systems, vol. 2, no. 4, pp. 267-288, 1984.

[12] I.D. Felipe, V. Hristidis, and N. Rishe, "Keyword Search on Spatial Databases," Proc. Int'l Conf. Data Eng. (ICDE), pp. 656-665, 2008.

[13] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing Spatial- Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems," Proc. Scientific and Statistical Database Management (SSDBM), 2007.

[14] G.R. Hjaltason and H. Samet, "Distance Browsing in Spatial Data- bases," ACM Trans. Database Systems, vol. 24, no. 2, pp. 265-318, 1999.

[15] V. Hristidis and Y. Papakonstantinou, "Discover: Keyword Search in Relational Databases," Proc. Very Large Data Bases (VLDB), pp. 670-681, 2002.

[16] I. Kamel and C. Faloutsos, "Hilbert R-Tree: An Improved R-Tree Using Fractals," Proc. Very Large Data Bases (VLDB), pp. 500-509, 1994.

[17] J. Lu, Y. Lu, and G. Cong, "Reverse Spatial and Textual k Nearest Neighbor Search," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 349-360, 2011.

[18] S. Stiassny, "Mathematical Analysis of Various Superimposed Coding Methods," Am. Doc., vol. 11, no. 2, pp. 155-169, 1960.

[19] J.S. Vitter, "Algorithms and Data Structures for External Memo- ry," Foundation and Trends in Theoretical Computer Science, vol. 2, no. 4, pp. 305-474, 2006.

[20] D. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, and M. Kit- suregawa, "Keyword Search in Spatial Databases: Towards Searching by Document," Proc. Int'l Conf. Data Eng. (ICDE), pp. 688-699, 2009.

[21] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid Index Structures for Location-Based Web Search," Proc. Conf. Information and Knowledge Management (CIKM), pp. 155-162, 2005.