



Nearest Search Keyword Based On Geometric Properties Using Spatial Inverted Index

¹ K. Usha Rani, ²N. Sireesha, ³T. Anusha ⁴P. Shruti ⁵T. Moulika

¹ Assistant Professor, Dept. of CSE, Lendi Institute of Engineering and Technology, Andhra Pradesh, India

^{2, 3, 4, 5} Dept. of CSE Lendi Institute of Engineering and Technology, Andhra Pradesh, India

Abstract — Spatial queries such as range search and nearest neighbour retrieval involves the geometric properties on an objects conditions. In these spatial database handles multi dimensional objects. Now-a-days many applications are used for many new queries to find the objects and satisfy both spatial predicate[2] and a predicate on their associated texts . for example, instead of considering all the colleges a nearest neighbour query would ask for the college that is the nearest among those whose department contains " CSE, ECE , EEE, MECH" . In the existing system IR2 trees are used to give the best results of the nearest neighbour it has few flaws like ,it fails to provide real time answers on multiple inputs. So we implement a new algorithm called Spatial Inverted Index(SI-index)[1] which is the extension of the conventional inverted index[6] to manage with multi dimensional data. Thus, the proposed algorithm is easy to find the required objects.

Keywords — spatial inverted index[1], spatial predicate[2], nearest neighbour search[3], keyword search[4], multi dimensional data[5],inverted index[6],R-trees[7],signature files[8],IR2-trees[9].

I. INTRODUCTION

A spatial database manages multidimensional objects such as points, polygons, circles, and provides fast access to those objects based on various selection criteria. The concern of spatial databases is reflected by the accessibility of modeling entities of reality in a geometric manner. For example, locations of colleges, hospitals ,schools and so on are usually represented as point in a map, while larger extents such as universities, lakes, and oceans usually as a combination of rectangles. various functionalities of a spatial database are useful in many ways in specific conditions. For sample, in a geography information system, range search can be extend to find all colleges in a certain area, while nearest neighbour retrieval can identify the colleges closest to a given address. Today, the general use of search engines has made it realistic to write spatial queries in a new way. Generally, queries focus on objects' geometric properties only, such as whether a point is in a polygon. 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 useful if a search engine can be used to find the nearest colleges that offers departments . Note that this is not the "globally" nearest colleges but the nearest colleges.

There are easy ways to support queries that combine spatial and text features. for example, for the query , we could first fetch all the colleges whose departments contain the set of keywords(CSE,ECE,EEE,MECH) and then from the retrieved colleges , find the nearest one . similarly, one could also do it reversely by targeting first the spatial conditions (browse all the colleges in ascending order of their distances to the query point until encountering one whose college has all the keywords).The major fault of these straightforward approaches is that they will fail to provide the real time answers on different inputs. An example is that the real nearest neighbour lies quite faraway from the query point, while all the closer neighbours are missing at least one of the query keywords

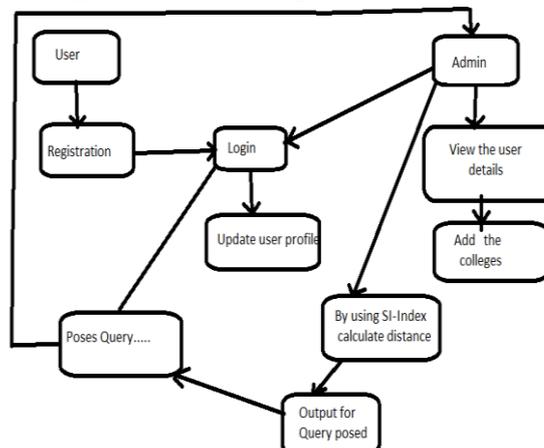


Fig1: Architecture

II. RELATED WORK

Inverted indexes (I-index)[6] have proved to be an effective access method for keyword-based document retrieval. In the spatial context, nothing prevents us from treating the text description of a point p as a document, and then, building an I-index. 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. As both lists are sorted in the same order, we can do so by merging them, whose I/O and CPU times are both linear to the total length of the lists.

ALGORITHMS USED

Spatial inverted index Algorithm:

Input: Query, Cache Queries

Output: Result set generated for query

Procedure:

If Query available in cache

Result related to query: =

ForwardToTreeprocess (Query)

Else

Result related to query: = GeocodingtreeProcess (Query)

Geocoding process(Query):

Parameters

Qi—Input Spatial Query

Qj (j=1...n) ---Set of Queries contains same Location

Dist[j] (j=1.....n)----Array for set of distances

Procedure:

(xi,yi)---Geocodings of Qi

(xj,yj)--- Geocodings of all queries with respect to location

Dist[i]=Euclidean distance between the geocodes

While not leafnode

Read nodes from tree For Q.features

If Q.features[i]==Q.features[j]

Add to list

End while

Sort list by feature and distance

Return list.

ForwardToTreeprocess ()

1. Build an empty list

2. Make a root node

3. if Qi in cache and status=false

For j=0 to n

Compare features(Qi,Qj) status=true;

For Each child in tree

If(status==true)

Getnodebyfeature (Qi);

Getnodebyfeature (Qj);

End

Else

Empty list ()

End For Each

4.Add nodes to list

5.Return list

Inverted index:

Inverted indexes (I-index)[6] have proved to be an effective access method for keyword-based document retrieval

III. EXISTING SYSTEM

Spatial queries with keywords have not been largely explored. In the past year the community has sparked enthusiasm in studying keywords search in relational databases. They nicely integrate two known concepts: r-tree, spatial index, and signature file[8], an effective method for keyword based retrieval. By doing so they develop a structure called the ir2-tree, which has the strengths both r-trees[7] and signature files[8]. Like r-trees[7], IR2-tree preserves objects proximity, which is the key to solving spatial queries efficiently. In another way that we have the signature files[8] and the ir2-trees[9] are used to filter a big portion of the objects that does not contain all the query keywords. Therefore it significantly reduce the number of objects to be examined.

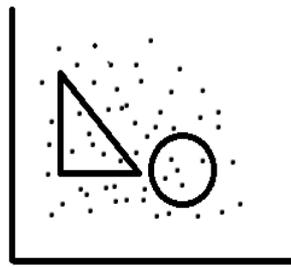


FIG2:EXISTING SYSTEM

IV. PROPOSED SYSTEM

we design a variant of inverted index[6] that is optimized for multi-dimensional points, and is thus named the spatial inverted index[1]. This access method successfully merges the point coordinates into a conventional inverted index[6] with small extra space, due to the delicate compact storage scheme. Meanwhile an SI index preserves the spatial locality of data point and comes with an r-trees[7] built on every inverted list. As a result it offers two competing ways for query processing. We can merge multiple lists very much like merging traditional inverted lists by Id's. Alternatively we can also leverage the r-trees[7] to browse the points of all relevant lists in ascending order of their distance to the query point. By this the SI index performs better than Ir2- tree in query performance.

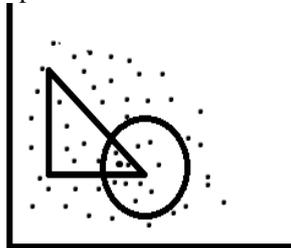


FIG3:PROPOSED SYSTEM

V. CONCLUSION

we have seen plenty of applications calling for search engine that is able to efficiently support different forms of the spatial queries that are combined with keyword search[4]. The existing results to such queries either to earn preventive space consumption are unable to give real time answers. In this paper we have changed the situation by developing an access method called the spatial inverted index [1]. Not only that the SI index is quite efficient but also it has the ability to find keywords that are nearest neighbour search[3] gives the results in a short time, furthermore as the SI index is based on the typical technology of inverted index[6].

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. Sudarshan. Keyword searching and browsing in databases using banks. In Proc. of International Conference on Data Engineering (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 prestigebased 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 (SIGMOD), 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.