iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 Issue: VI Month of publication: June 2015

DOI:

www.ijraset.com Volume 3 Issue VI, June 2015
IC Value: 13.98 ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Density Based Quickly Accessible Neighbour Search with Keywords

Aiswarya S[1], Usha K[2]
[1,2]*Dept of Computer Science & Engineering,*
NSS College of Engineering, Palakkad, Kerala

Abstract— Traditional spatial queries, for example, range search and closest neighbor recovery, include just conditions on items location properties. Today, numerous present day applications request forms of queries that intend to discover articles fulfilling both a spatial predicate, and a condition on their related writings or text. For an example, as opposed to considering all the lodgings, a closest neighbour inquiry would rather request the inn that is the closest among those whose menus contain "steak, spaghetti, schnaps" in the meantime. Presently, the best answer for such queries is focused around the Ir^2-tree. It has a couple of inadequacies that essentially affect its effectiveness. Persuaded by this, another access technique called the spatial inverted list that broadens the conventional inverted index to adapt to multi dimensional information, and accompanies with algorithms that can answer closest neighbour questions with keywords in real time was proposed. The ranking of articles based on distance or the frequency of keywords were used in the above method. The computation of the shortest distance increased the complexity of the method. So here a new method is proposed that finds the neighbours with less space and time complexity.

Keywords— ***Nearest neighbour queries, inverted index, spatial queries, geographic information systems, latitude and longitude of earth.***

I. INTRODUCTION

A spatial database manages multidimensional articles, (for example, focuses, rectangles, and so forth.), that gives quick access to those items based on different selection criteria. For example, location of restaurants, stopping regions, inns, clinics so on are often represented as points in a map, while some larger extents such as lakes, parks and landscapes often as a combination of rectangles.. For example, in a geographic information system, range search can be utilized to discover all restaurants in a specific zone, while closest neighbor recovery can find the restaurant closest to a given location. Today, the extensive utilization of search engines has made it pragmatic to compose spatial queries in a brand new manner [1].

Different from the previous conventional spatial database question, spatial keyword query includes uncertainties when locating spatial information objects. There is regularly no precisely right response for a given query. That is, for spatial information objects with text based depictions, it is not ready to characterize an absolute correct rule to tell which one is the best for a given set of query keywords. So, in most cases, ad hoc patterns are used to find a set of reasonable objects. This issue is really acquired from classical information retrieval area where subjective standards are often used to judge the quality of results in mining information of interests, and, indeed, spatial keyword query has become a research problem covering traditional GIS, database, and information retrieval research. Numerous methods from text data recovery have been applied in the spatial keyword query methodology to implement intelligent data services for spatial object discovering [2].

Usually, queries concentrate on articles geometric properties just, for example, some modern applications that require the capacity to choose items focused around both of their geometric directions and their associated texts. For example, it would be genuinely helpful if a search engine can be utilized to discover the closest restaurant that gives "steak, spaghetti, and schnaps" all in the meantime. This is not the "globally" closest restaurant yet the closest restaurant among just those offering all the requested nourishments and beverages [1].

There are two ranks for selecting proper objects as per the query condition: one is the text rank that decides to what degree the text based description of an item match the query keyword; the other is the spatial rank which positions items as per the distance constraints of the query. Also, a final rank combining these two positioning elements will be processed to rank queries in a unified manner and return applicable objects as indicated by the final ranks. Individuals have effectively directed far reaching research in these two differentiated areas. However it is still a tough problem when joining these two ranking problems into one framework. Computing of one rank after the other takes a lot of time. Existing solutions give an integrated ranking framework that can compute two ranks in one run utilizing certain approximate ranking algorithm. Along these lines, the key issue is how to implement a well-organized synthetic object filter and ranking framework [3].

For instance, for the above given query, it would first get all the restaurants whose menus contain the set of keywords “steak, spaghetti, schnaps”, and afterward from the retrieved results, find the closest one. Thus, one could do it conversely by focusing on first the spatial conditions—search all the restaurants in ascending order of their separations to the query point until encountering one whose menu has all the keywords [1].

The significant disadvantage of the basic methodologies is that they will neglect to give ongoing replies on real-time inputs. A typical example is that the real closest neighbour lies far from the query point, while all the closer neighbours are missing at least one of the query keywords. They fairly coordinate two remarkable concepts: R-tree, a prevalent spatial index, and signature file, a powerful system for keyword based retrieval. By doing as such they developed a structure called the Ir^2-tree, which is having the qualities of both R trees and signature files. Like R-trees, the Ir^2- tree preserves objects spatial closeness, which is the key to solving spatial queries efficiently. On the other hand, in the same way as signature documents, the Ir^2- tree has the capacity to filter a huge segment of the queries that do not contain all the query keywords, therefore extensively lessening the quantity of items to be inspected.

The Ir^2-tree likewise acquires a downside of signature files: false hits. A signature file, may even now direct the search to a few items, despite the fact that they don't have all the keywords. It needs loading its full text depiction, which is expensive because of the resulting random access [2].

A variation of inverted index enhanced for multidimensional points, named the spatial inverted index (SI-record) is proposed. This method effectively consolidates point coordinates into conventional inverted index with small additional space, with a delicate and compact storage plan.

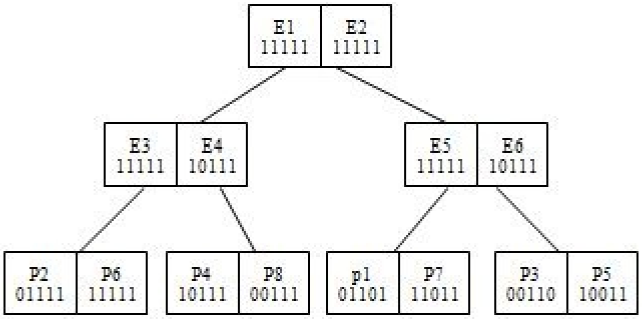

Fig1. Example of an IR^2-tree

An SI-index protects the spatial region of data points, and accompanies a R-tree based on every inverted list at little space overhead. Consequently, it offers two contending methods for query processing. Firstly by merging (consecutively) different lists all that much likes merging conventional inverted lists by ids. On the other hand, by forcing the R-trees to find the points of all relevant lists in increasing order of their distances to the query point [1].

Another paradigm can be added along to the distance estimation and query search as a ranking method. The density of the area can be figured out utilizing a convenient system. The denser area in locality is more liable to be chosen by individuals. We utilize the collective keyword query semantics to discover in a denser area, a group of spatial items whose keywords collectively match the query keywords. To efficiently handle the density based spatial keyword query, we utilize an IR^2-tree index as the base data structure to index spatial objects and their text contents and characterize a cost function over the IR-tree indexing nodes to approximately process the density of the area. We outline a heuristic algorithm that can proficiently prune the area as per both the distance and region density in handling a query over the IR-tree file

II. RELATED WORKS

A. Nearest Neighbor Queries

Cong et al. considered a manifestation of keyword based closest neighbor queries that is like our definition, yet contrasts in how objects associated text play a part in deciding the query result. The technique presents the pseudo neighbor, which is not the genuine closest neighbor; however another closest neighbor is chosen on the premise of estimation of weighted total of distances of k NN unclassified patterns in each class. At that point Euclidean distance is assessed and pseudo neighbor with more prominent weight is discovered and arranged for unknown sample.

1) *R-Tree Algorithm:* The first baseline algorithm, R-Tree, makes use of only an R-Tree data structure. Given a distance-first top-k spatial keyword query, the algorithm first finds the top-1 nearest neighbor object to the query point$Q.p$. Then it retrieves the object (since the R-tree only contains object pointers) and compares that object's textual description with the keywords of the query. If the comparison fails then that object is discarded, and the next nearest object is retrieved. The incremental NN algorithm is used. This process continues until an object is found whose textual description contains the query keywords. Once a satisfying object is found it is returned and the process repeats until k objects have been returned. The drawback of this algorithm is that it has to retrieve every object returned by the NN algorithm until the top-k result objects are found. This potentially can lead to the retrieval of many "useless" objects. In the worst case (when none of the objects satisfies the query's keywords) the entire tree has to be traversed and every object has to be inspected.

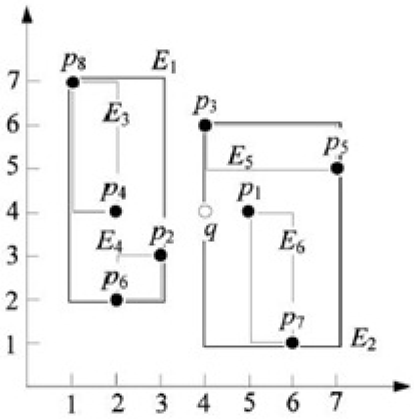

Fig 2. The MBRs of the underlying R-tree

2) *IIO Algorithm:* The IIO baseline algorithm makes utilization of an inverted index. It first discovers all the items (object ids) whose text record contains the query keywords by intersecting the lists returned by the inverted index. Let V be the set of articles in this intersection. At that point the articles in V are recovered and the distance between the query point and each of the items in V is figured. These items are sorted and the top-k articles are returned. The input parameters are the inverted index I and the distance-first top-k spatial keyword Q

3) *Distance-first IR^2-tree algorithm:* The distance first form of the IR^2-Tree algorithm yields the items that contain all query keywords ordered by their distance from the query point. The distance first IR^2-Tree calculation exploits the structure of the IR^2-Tree to productively answer distance-first top-k spatial keyword queries. The tree traversal is focused around the incremental closest neighbour calculation. The key advantage of this calculation is that it prunes entire sub trees if their root-node signature does not match the query signature.

This happens because the signature of an IR^2- Tree node is formed from all the signatures of its child nodes. This pruning happens in addition to the spatial pruning provided by the conventional Incremental Nearest Neighbour. By tightly incorporating these two pruning methods, the distance- first IR^2-Tree algorithm accesses a minimal set of IR^2-Tree nodes and objects to answer a distance-first top-k spatial keyword query. In geographic web search, every web page is allotted a geographic region that is related to the webpage's contents. In web search; such areas are considered so that higher rankings are given to the pages in the same region as the location of the computer issuing the query. Zhang et al. managed the supposed m-closest keywords issue. Cao et al. [6] proposed collective spatial keyword querying, which is focused on similar ideas, however goes for enhancing different objective functions. In [5], Cong et al. proposed the idea of prestige based spatial keyword seek. The main focus is to assess the similarity of an item p to a query by considering the items in the neighbourhood of p. Lu et al. consolidated the thought of keyword search with reverse closest neighbour queries. In past works, a result object needs to cover the entire set of the query key words. However in genuine applications, it is often excessively strict to discover such objects matching each of the query word. Collective spatial keyword query is utilized to find a set of items with the union of their keywords covering the entire query word set [9]. In an R-tree record with inverted file index is utilized to proficiently handle the collective spatial keyword query. The primary thought is to utilize a highly characterized cost function to heuristically seek the list tree, which can attain to a decent result set with pleasant execution.

However it still does not consider the density of a zone where the articles are found in. Our density based spatial keyword questioning is not quite the same as their systems in such a way that the densities of the area of the objects are given consideration. Our technique gives an engineered cost function to make a trade off between the query items distances and the item densities such that the result can give clients more candidates, that is, the hot zones with more choices. The IR^2-tree is the first access technique for evaluating NN queries with keywords. Likewise with numerous pioneering arrangements, the IR^2-tree additionally has a couple of downsides that influence its proficiency. The most genuine one of all is that the quantity of false hits can be truly huge when the object of the last result is far from the query point, or the result is just unfilled. A SI-index protects the spatial locality of data points and accompanies an R-tree based on every inverted list at little space overhead. Thus, it offers two contending courses for query handling. Firstly by consolidating (successively) numerous records all that much like merging conventional inverted lists by ids, followed by constraining the R-trees to browse points in increasing order of their distances to the query keyword.

III.PROPOSED SYSTEM

The spatial inverted list (SI-index) is essentially a good version of an inverted index with embedded coordinates. Query processing with an SI-index can be done either by merging or together with R-trees in a distance browsing manner. The distance calculation job is performed utilizing the latitude and longitude of each one point allocated in the Haversine formula. The formula expects a round earth. Nonetheless, the state of the earth is perplexing or complex. An oblate spheroid model will give better results. In the event that such accuracy is required, better decision is to utilize Vincenty converse equation. It can give 0.5mm exactness for the spheroid model. There is no perfect equation, since the genuine state of the earth is so unpredictable it would be impossible be communicated by a formula. Moreover, the state of earth changes because of climatic conditions furthermore changes over the time because of the turn of the earth. Additionally the space unpredictability is sort of high for this situation. The directions for each point are put away independently in the database for every individual substance. It requires to store the directions i.e. the latitude and longitude of the points on space as partitioned sets. It is a downside when the spatial directions are put away on the database for each one point as its crude value. It needs the distance processing at each one stage and it takes an extensive time to register the distance utilizing the formula. It was found that the time taken for distance calculation by using the above mentioned formulas were much higher. It increases the complexity of the method described above. So there is a need to simplify the computation procedure. The geographic coordinate system is mainly used to represent the spatial or geographic data. It mainly has three ways of representing the data. They are:

Degrees, Minutes, seconds
Decimal Degrees
Degrees, Decimal Minutes

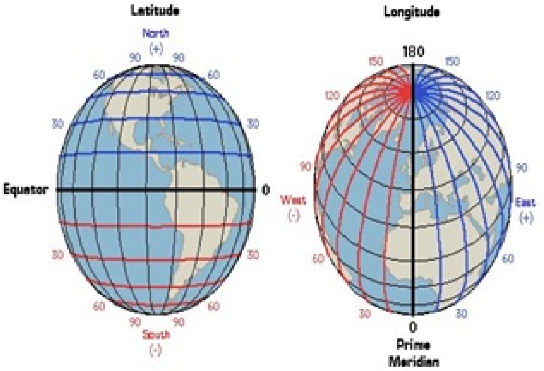

Fig 3. Latitude and longitude of earth

Among the three methods, the decimal degree representation is comparatively easier for the readers to grasp easily. The representation is as (latitude, longitude, elevation) triplets in their decimal values. The third value elevation is always not necessary in the context of locating a particular area. So it can be skipped. The representation of the points can be in the form of coordinates with two digit prior to the decimal point and up to six digits followed by the decimal point. For example, Palakkad, Kerala, India can be represented as (10.786730,76.654793). It is found that the coordinates of nearby locations often change by a factor of 0.000001-0.009999.This implies that there exist the same range of latitudes and longitudes over a range of area with only a few change in the values of least significant decimal points. There for instead of calculating the distance from the query point to the location, the range of any of the values of latitude or longitude can be checked. Suppose a point is having the location coordinate (x,y) and it is the query point. The latitude is given first consideration in this method. It will only retrieve those points with their latitudes nearer to the query point only i.e, (x-m) or (x+m) thus by considerably reducing the data points by a factor of m, where m<n (n is the total no of data points which is very large) and the time complexity occurred during the formula computation. The value of m depends upon to what max distance the user needs to find the closest neighbor . Normally m can be given a small value lesser than one. After filtering using the latitude alone, longitude can be considered. Suppose n=100, m=1 and x=10. The query will retrieve only those points with coordinates (9,y)and (11,y). Rest all are filtered out. The no of results retrieved is of order 2m. The complexity is thus reduced from O (n) to O (m).For filtering based on longitude, the time taken is of order m again. So the total complexity is of order O (m) which is very less when compared to that of distance calculation method. It also requires checking the density of population of the place searched. In high density regions like cities and towns, it is likely that number of results received may be higher when compared to low densely populated areas like villages. So if we add an additional parameter that determines the population density of each place the query results can be extended or reduced accordingly. In towns or cities there may be more number of hotels and restaurants within a small radius while it is a very few in remote villages. So the search radius can vary from area to area. This will enhance the searching time and efficiency of the technique considerably. Next important thing to be considered is the dataset and its storage. It is very efficient to have a sorted list of points in the database for retrieval. The quick sort can be used as it is having O (n log(n)) at its best case when compared to insertion sort with best case complexity O (n). But the worst case space complexity of quick sort is O (n) while that of insertion sort is O (1). Giving importance to space and since the sorting is done only at the time of setting up of the dataset; the insertion sort can be utilized. The worst case time complexities for both the methods are O (n^2).

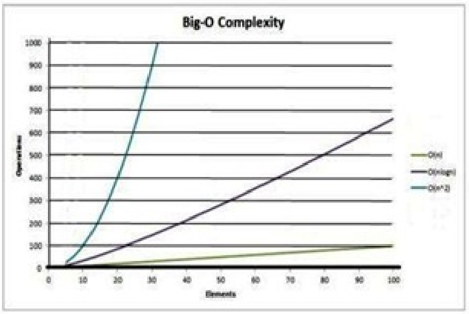

Fig 4. Comparison of Quick sort and Insertion sort

In spite of this spatial constraint, the tag or description of the point can be analysed for filtering out the objects based on query keyword. The points that collectively match the query keywords can be retrieved.

IV. CONCLUSIONS AND FUTURE WORK

There are a 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. Here it is remedied by developing an access method called the spatial inverted 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 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. Instead of ranking the points based on distance computation and text constraints, it is easy to find the neighbor by checking whether their spatial coordinates are in a range.. This can give accurate results within less time. It can be used to support many important information mining services in Web-GIS applications.

V. ACKNOWLEDGEMENT

The authors would like to thank professors of NSSCE, Palakkad for suggestions and support on this paper.

REFERENCES

[1] Yufei Tao and Cheng Sheng, “Fastest Nearest Neighbor Search with Keywords”, in IEEE Transactions on knowledge and data engineering, April 2014 ,vol. 26,no.4,pp 878- 888.

[2] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu , A . Skovsgaard , D. Wu ,and M.L. Yiu , “Spatial Keyword Querying,” Proc. 31st International Conference on Conceptual Modeling (ER), 2012 , pp. 16-29.

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

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

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

[6] Y.-Y. Chen, T. Suel , and A. Markowetz , “Efficient Query Processing in Geographic Web Search Engines,” Proc. ACM SIGMOD International Conference on Management of Data, 2006 ,pp. 277-288.

[7] 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.

[8] J.S. Vitter, “Algorithms and Data Structures for External Memory,” Foundation and Trends in Theoretical Computer Science, 2006, vol. 2, pp. 305-474.

[9] H. Zhuge, Y. Xing, “Probabilistic resource space model for managing resources in cyber physical society”, IEEE Transactions on Services Computing, 2012, vol. 3 pp. 404-421.

[10] De Felipe, V. Hristidis, N. Rishe, Keyword search on spatial databases, in:Proceedings of ICDE 2008, pp. 656–665.