Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 8, August 2014, pg.267 – 274

RESEARCH ARTICLE

A Practical Approach to Faster Spatial Queries with Keywords

B.ChinnaReddaiah[1], K.R.K.Satheesh[2]

[1]Student, Department of CSE, Madanapalle Institute Of Technology & Science, JNTUA University, A.P, India

[2]Associate Professor, Department of CSE, Madanapalle Institute of Technology & Science, JNTUA University, A.P, India

Reddaiah509@gmail.com, satheeshkrk312@gmail.com

Abstract - ***In database search the similar data with multimedia databases requires support of near-neighbor to access high dimensional data for accessing of individual queries. The respondents of voronoi based accessing based cell to allow the high datasets. In the proposed mechanism the efficiency of the data and distributed data can access the cells, in the R-tree Patterns. In this method the data can be verified and classified in to the root node to child node for identifying the neighbor node for the similar multimedia datasets. In our new approach, we therefore pre compute the result of any nearest-neighbor search which corresponds to a computation of the voronoi cell of each data point. In a second step, we store the voronoi cells in an index structure efficient for high-dimensional data spaces. As a result, nearest neighbor search corresponds to a simple point query on the index structure. Although our technique is based on a precipitation of the solution space, it is dynamic, i.e. it supports insertions of new data points. This kind of spatial query or nearest neighbor query is hard to achieve. This is the open problem that can be addressed.***

Index terms - Fast nearest-neighbor search, Information retrieval, spatial index, keyword search

1. INTRODUCTION

Keyword search in document performed with various approaches ranked retrieval results, clustering search results & identifying the nearest neighbor Keyword search on xml document categorized as two different approaches one is Keyword search on xml document which can be performed by ranking the searched results based on match or the answer to keyword & finding the nearest neighbor of keyword by using GST or by Xpath Query. In paper [2] the problem of returning clustered results for keyword search on documents the core of the semantics is the conceptually related relationship between keyword matches, which is based on the conceptual relationship between nodes in trees. Then, we propose a new clustering methodology for search results, which clusters results according to the way they match the given query.

Spatial databases are the databases that hold geographical information. They contain data that is essentially multidimensional. Such databases can be used for obtaining information based on queries. However, the queries of this kind will have more number of criteria making it so complex. Based on the query point and predicates, the underlying system should process the request. For instance query might be made on hotels, locations, schools, hospitals, lakes, parks and so on with various combinations of criteria. Many location based services in the world need nearest neighbor searches. The NN search is not simple. It needs to process huge amount of geography information. As the number of geographic properties are more and multi-dimensional in nature, developing such systems is considered non trivial. Search speed is an essential element in such systems. With respect to a web application for making searches on geographical data, speed is an important factor. The application should produce desired results in short span of time. This is a challenging problem needs to be addressed.

The report of Internet has recognized grow to an forever increase amounts of text data associate with multiple dimension (attributes), for example, client reviews in shopping website (e.g., Amazon) are continually associate with attribute like prices, models, and rates. A traditionals OLAP information cubes can be in nature extensive to review and navigate structured data together with unstructured text data. Such a cube model is called text dice [1]. A cell in the text cube aggregate a set of document/items with identical attribute values in a split of size. Keyword inquiry, one of the most accepted and easy to use ways to recover helpful data from a group of basic documents is being extensive to R-DBMS to regain data from text-rich attribute [2], [3]. Given a set of keywords, accessible methods aim to find important items or join of objects (e.g., link by distant keys) that control all or several of the keywords. Conventional IR technique can be used to status documenters‟ accord to the import. In a large text DB, however, the number of related documents to a queries could be great, and a user have to use much time impression them. If a document is connected with attributes information's, in a data cubes model („a multidimensional space induced by the attributes‟) e.g., the passage cube, a cell aggregate the document with identical value in a separation of attribute. Such a assembly of documents is associate with each cell, consequent to an object that can be directly optional to the user for the given query. These paper study the problems of keyword-based top-k search in text cubes, i.e., given a keyword query, find the top-k most significant cells in a text cube. When user want to regain information from a text cube using keyword question, believe that important cells, rather than important documents, are prefer as the answers, because: 1) important cells are easy for users to browse; and 2) important cells present users insight about the association b/w the values of relational attributes and the text data.

Many existing solutions use R-Tree for holding indexing information. The R-Tree is one of the data structures which are dynamic in nature. It is best used to hold information that will change frequently. When new data is added, or an existing data is modified, the R-Tree is bound to get changes. This makes the data structure so dynamic and it needs to have huge amount of indexes so as to support faster access to the data which resides in the underlying database. The R-Tree structure is suitable for making NN queries faster. However, this structure has been improved, as found in the literature, and new indexing structure IR-Tree came into existence.

Spatial queries with complex parameters were not explored more. Felipe et al. explored it recently and the indexing structure IR-Tree [1] is used. This tree has some drawbacks as it is not able to cope with huge amount of data containing multiple dimensions. This problem is also known as "curse of dimensionality".

To overcome the drawbacks of IR-Tree, recently, Yufei Tao and Cheng Sheng proposed a new indexing structure named "spatial inverted index" which has been derived from conventional inverted index. This new indexing structure holds huge amount of indexes and dynamic in nature. It is capable of coping with multi-dimensional data. It helps users to have faster access to geographical data. However, Yufei Tao and Cheng Sheng [2] did not apply this indexing structure to some real world application.

2. RELETED WORK

In the paper „fast nearest neighbor search with keywords‟, there are methods like spatial index, inverted index, nearest neighbor search. The first method spatial index is used for creating indices because there is huge amount of data need to be stored for searching that data stored in the form of xml documents. If the data storage created in the form of indices then space required is less also time needed for searching the keyword is less. Second method is reversed index. The reversed index data structured in a central module of a usual search engine indexing algorithms. A goal of a search engine presentation is to optimize the speed of the query: find the documents where word occurs. Once an index is residential, which provision lists of words per document; it is next inverted to develop an inverted index. Querying the index would need sequential iteration throughout every document and to every word to verify a matching document. The time memory and dispensation possessions to affect such a query are not always notionally practical. Instead of listing the words per article in the index, the inverted index

data structure is urbanized which lists the documents per word. The inverted index produced, the query can now be determined by jumping to the word id in the inverted index. These were effectively inverted indexes with a small amount of supplementary explanation that required a implausible amount of attempt to produce.
Third method is near neighbor searching. Nearest neighbor search (NNS), also identified as closeness search, parallel search is an optimization problem for finding closest points in metric spaces. In the paper „Efficient Keyword-Based Search for Top-K Cells in Text Cube" methods used are inverted-index one-scan, document sorted-scan, bottom-up dynamic programming, and search-space ordering. In the top k cells, there is a searching of nearest key to the query. Cubes forms clusters of single unique group which shows its identity.

The spatial dimension is requiring increasing the online objects with geo-location and a text description. The web browsers are satisfied increasing being geo-located and geo-coded also textual information of place is important. To find the different location quickly spatial keyword queries are helpful to strangers. The techniques allow the indexing of data, which contain the both textual information and geo location to answer the spatial keyword queries. The spatial keyword queries are helpful in real life applications.

For example Google maps for each point of location the square shapes geo-tagged documents can be produced. In the twitter also tweets can be extracting. In the field of research community development also the spatial keyword queries are more interest. We have three types of spatial keyword queries. Those are

1) Boolean kNN query
2) Top-kNN query
3) Boolean range query.

Boolean kNN Query**:** Extract the k objects nearby the users nearby positions such that each objects text description contains the keywords tasty, pizza and cappuccino".

Top-k kNN Query**:** extracting the k objects with the top ranking scores, calculate from distance between query position and the relevant of their text description to the query keywords tasty, pizza and cappuccino.

Boolean Range Query**:** "Extract all objects whose text report contains the keywords tasty, pizza, and cappuccino and whose location is within 10 km of the query location.

We introduced the no of geo-textual indices. These indices generally mix with spatial index and a text index structure.

According to spatial index structure the existing indices can be characterized as

1) R-tree based indices
2) Grid based indices
3) Space filling curve based indices

According to text index structure these indices also classified as

1) Inverted file based indices
2) Signature file based indices

The previous geo-textual indices often report on empirical studies that imply the proposed indices are very competitive with baseline indices. To decide which type of index is most suitable for particular setting is difficult even though by using state of affairs.

3. EXISTING SYSTEM METHODOLOGY

The existing application proposed by Tao and Shang builds a novel access method named "spatial inverted index". This index is different from conventional index. It can deal with multi-dimensional data compatible with NN queries. The search is made using keywords. This method has higher performance when compared with IR2-tree in terms of query response time.

Drawbacks of Existing System

- The existing system does not apply the proposed indexing to a search engine kind of application.
- It does not provide a prototype application to demonstrate proof of concept practically.

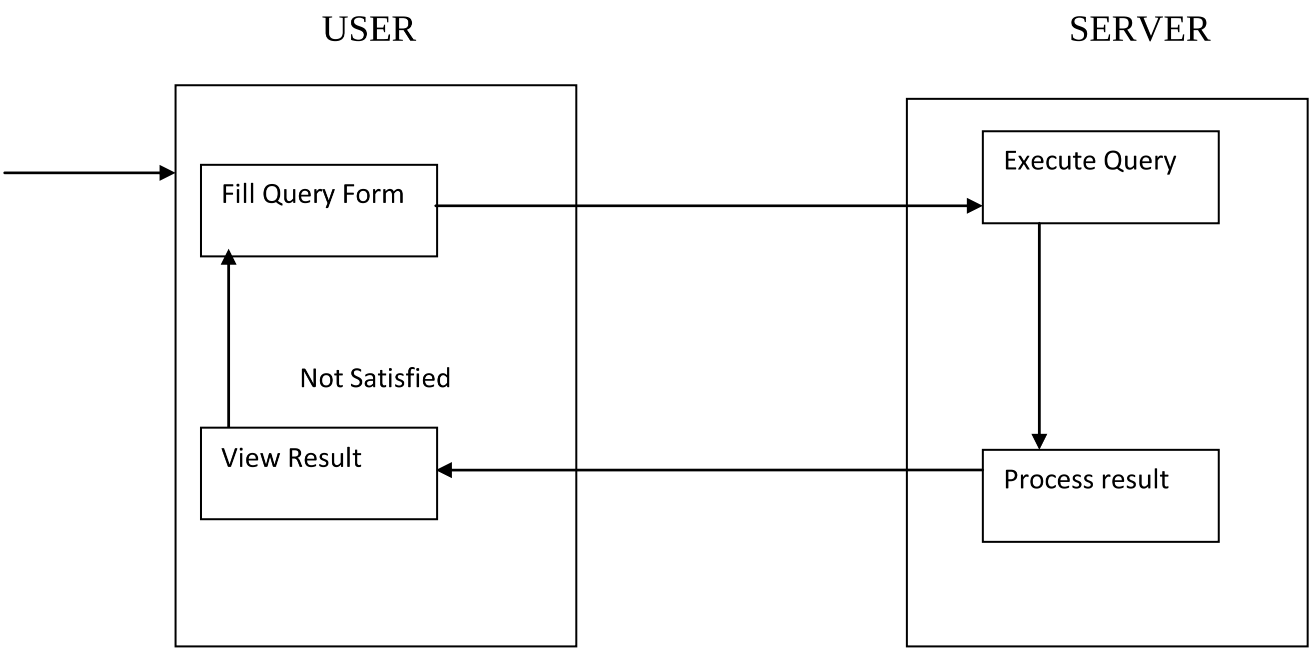
4. PROPOSED SYSTEM

The proposed system is an extension to the existing system which practically employs the proposed indexing method for faster retrieval of results pertaining to spatial or NN queries by building a search engine kind of application. We built a prototype application that demonstrates the proof of concept.

Advantages of Proposed System

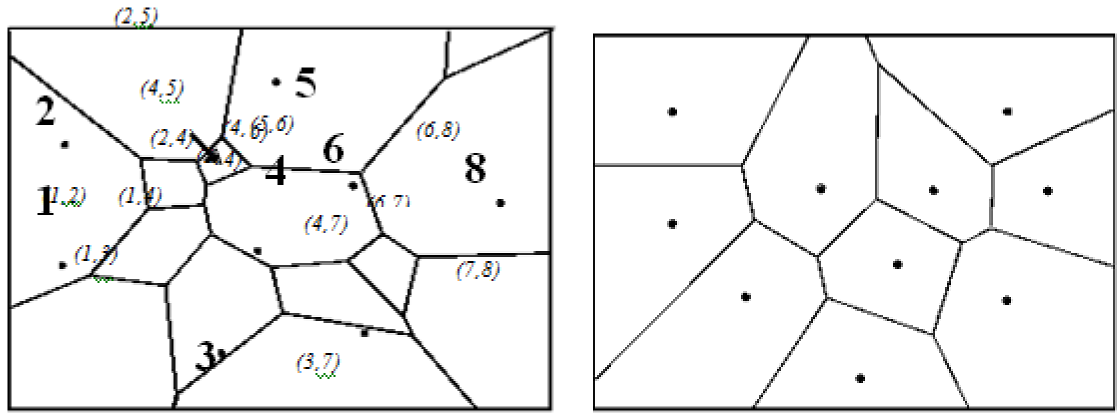
- Faster NN search.
- Prototype application to demonstrate the proof concept.
- Search engine to demonstrate the faster searches.

5. SYSTEM ARCHITECTURE



Algorithm

Our new advance to solve the adjacent national problem is based on pre calculate the solution space. Pre-calculating the solution space means determining the Veroni diagram (cf. figure 1a) of the data points in the database. In the following, we recollect the definition of Veroni cells as provided in [Ro 91].



a. Voroni Diagram of order 2 (cf [PS 85])   b. NN-Diagram

According to designation 2 the sum of the volume of The NN-Diagram of a database of point DB is define as the equivalent estimate diagram for two independent equally spread dimension, figur 2c and d show the two diagrams for a usual multidimensional uni-form distribution, and figure 2e and f show the diagrams for a spare delivery. Let us now consider examples of the NN cell and their MBR approximation for a number of different data distributions. Figure 2a and b show the NN drawing and A consistent delivery is typically generate by using a arbitrary number producer to construct the data values for both of the dimension separately.

6. IMPLEMENTATION

In this article, we propose a newest explanation to in order near national searching which is base on pre calculating and index the explanation break instated of index the information. The solution liberty can be characterized by a total and be related free partition of the information hole into cell, each contain closely single information spots. Every cell consists of all possible reservation point which has the consequent data point as a adjacent fellow citizen. The cell consequently corresponds to the d-dimensional Verona cell [PS 85]. Deter mining the adjacent national of a queries position now become equal to decisive the Verona cells in which the queries point is situated. While the Verona cell can be relatively multipart “high-dimensional polyhedral” which involve too greatly disk space after stored clearly, we estimated the cell by least bound rectangle and accumulate them in a “multidimensional directory agreement” such as the R-tree.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

1. **Registration**
2. **Login**
3. **Hotel_Registration**
4. **Search Techniques**
5. **Map_view**
6. **Distance_Search**

Spatial Inverted List:

The essential compressed version of I-index with embedded coordinates is spatial inverted list (SI-index). Using SI-index we can do query processing by two ways. One is by merging or second is together with R-tress in a distance browsing. The defect of conventional I-index is eliminated by compression because SI-index use small amount of space.

Compression Scheme:

To reduce the size of an inverted index in the conventional context the compression scheme is mostly used. One unique id is allocated to for each inverted list. This is best useful approach is to record the gaps between consecutive ids and opposite ids.

The each element of list is a point p as (id_p,x^p,y_p). The each point contains the Id and two coordinate values of point. The gap keeping have need of a sorted order, this we can apply on only one attribute of the triplet. If we apply the sorting order in list ids, it works effectively the gap keeping achieves the space saving. But if apply on coordinates of point it's not works effectively.

To solve this problem first we apply the gap keeping on coordinates of element of list. But here each element has two coordinates. we change it into only one then gap-keeping apply successfully. To change the multidimensional point into 1D we use Space Filling Curve (SFC) tool. If the two values are in the same space their 1D value will be same. Gap-keeping works effectively after sorting the 1D value.

Building R-trees:

The compressed version of inverted index is SI-index with converting of co ordinates. It achieved by the merging a number of inverted lists. In these modules we index of each inverted list by R-tree. To process the query by using distance browsing the R-tree works efficiently. The ultimate goal of R-tree is every leaf node of R-tree is block of an inverted list.

The existence of coordinates in the inverted lists logically motivates the creation of an R-tree on each list indexing the points. Next, we discuss how to perform keyword-based nearest neighbor search with such a combined structure.

Registration:

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

Login:

In this module, any of the above mentioned person have to login, they should login by giving their email id and password.

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

Search Techniques:

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

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.

Restaurant Search:

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

Map_View:

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

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.

7. CONCLUSION

We planned a new method for professional neighboring state look for in a set of high-dimensional point. Our method is based on the precipitation of the solution freedom of any chance „nearest neighbor search". This corresponds to the division of the voronoi cell of the data point. Since voronoi cells may become quite compound when leaving to elevated dimension, we present a new algorithms for the approximate of high-dimensional voronoi cell using a set of least bound (hyper) rectangles. Though our practice is based on a precomputation of the clarification spaces, it is active, i.e. it wires insertion of original facts point. We lastly show in experiment estimation that our method is efficient for different kind of data and visibly outperforms the status of the art "nearests-neighber algorithms".

8. Feature Work

Spatial data is the data that reflects objects in the space. It contains data pertaining to real world objects such as cities, roads, rivers, buildings and so on. Some non-spatial data is also stored along with spatial data. Such data is in a repository known as spatial database. Mining process on such data is complex which is known as spatial data mining. Spatial queries are the queries that are answered based on certain criteria which might use both spatial and non-spatial content. The existing solution for spatial query processing was supporting single keyword query. However in the real world there might be related keywords and need to processing multi-keyword query. Towards this improvement is made for the single keyword search mechanism to support multi-keyword search using spatial data mining approach. A prototype application is built which demonstrates the multi-keyword search. Possible future directions include the improvement of visualization and application of spatial data mining for various domains.

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 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.

[11] C. Faloutsos and S. Christodoulakis. Signature files: An access method for documents and its analytical performance evaluation. *ACM Transactions on Information Systems (TOIS)*, 2(4):267–288, 1984.

[12] I. D. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 656–665, 2008.

[13] R. Hariharan, B. Hore, C. Li, and S. Mehrotra. Processing spatialkeyword (SK) queries in geographic information retrieval (GIR) systems. In *Proc. of Scientific and Statistical Database Management (SSDBM)*, 2007.

[14] G. R. Hjaltason and H. Samet. Distance browsing in spatial databases. *ACM Transactions on Database Systems (TODS)*, 24(2):265–318, 1999.

[15] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In *Proc. of Very Large Data Bases (VLDB)*,pages 670–681, 2002.
[16] I. Kamel and C. Faloutsos. Hilbert R-tree: An improved r-tree using fractals. In *Proc. of Very Large Data Bases (VLDB)*, pages 500–509, 1994.
[17] J. Lu, Y. Lu, and G. Cong. Reverse spatial and textual k nearest neighbor search. In *Proc. of ACM Management of Data (SIGMOD)*, pages 349–360, 2011.
[18] S. Stiassny. mathematical analysis of various superimposed coding methods. *Am. Doc.*, 11(2):155–169, 1960.
[19] J. S. Vitter. Algorithms and data structures for external memory. *Foundation and Trends in Theoretical Computer Science*, 2(4):305–474, 2006.
[20] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 688–699, 2009.
[21] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma. Hybrid index structures for location-based web search. In *Proc. of Conference on Information and Knowledge Management (CIKM)*, pages 155–162, 2005.

