Fast Nearest Neighbor Search with Keywords in Spatial Databases

Bolumalla Prasanna Jyothsna[1]
M.Tech(S.E) Student
Department of Computer Science and Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Bachupally Hyderabad-90 Telangana, India
e-mail: [1]josh.rebecka@gmail.com

Mr. Yeruva Sagar[2]
Associate Professor
Department of Computer Science and Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Bachupallay Hyderabad-90 Telangana, India
e-mail: [2]sagar_y@vnrvjiet.in

***Abstract*—** In these days, many modern purposes name for novel varieties of queries that purpose to find objects pleasing both a spatial predicate, and a predicate on their related texts. Present answer for such queries has a couple of deficiencies that critically influence its effectivity. Prompted by way of this, in this venture, development of a new entry process called the spatial inverted index that extends the conventional inverted index to cope with multidimensional data, and is derived with algorithms that may reply nearest neighbor queries with key words in actual time. As tested via experiments, the proposed approaches outperform the IR^2-tree in question response time tremendously, more commonly through a factor of orders of magnitude.

Keywords- *keyword search, spatial database, spatial inverted index, nearest neighbor search,*

*****

I. Introduction

A spatial database manages multidimensional objects (comparable to points, rectangles, and so on.) and presents quick access to these objects situated on unique choice standards. The value of spatial databases is mirrored with the aid of the convenience of modelling entities of truth in a geometrical manner. For example, places of eating places, hotels, hospitals etc are typically represented as aspects in a map, even as larger extents akin to parks, lakes, and landscapes commonly as rectangles. Many functionalities of a spatial database are valuable in more than a few approaches in detailed contexts. For illustration, in a geography know-how method, variety search can be deployed to find all eating places in a targeted subject, at the same time nearest neighbor retrieval can observe the restaurant closest to a given handle. It is to be noted that some ultra-modern applications that decision for the potential to opt for objects situated on the both of their geo-locational information and their associated texts. A spatial database is used to retailer huge amounts of area related knowledge akin to map, clinical imaging data and many others. The significance of spatial database is to provide a easy way to mannequin the entities of truth in a geometric method.

II. LITERATURE SURVEY

A. *Keyword Search*

With the development of the net, there was a fast broaden within the number of users who have to access on-line databases while not having a designated skills of the schema or of question languages even moderately easy query languages designed for non authorities are too tricky for them. Key based search as in [2] is the most information discovery methodology hence of the user would not need to be compelled to realize either a question language or the underlying structure of the information.

B. *Spatial Keyword Query Processing*

Geo-textual indices play a principal position in spatial keyword querying. The present geo-textual indices have not been when compared systematically. This makes it complex to examine which indexing method satisfactory helps special performance. There is a benchmark that allows for the assessment of the spatial key phrase query efficiency and also record on the findings bought when making use of the benchmark to the indices, accordingly uncovering new insights that can consultant index resolution.

C. *Spatial Database*

A database as in [2] is simply a set of structured information. In order to store, manage, and access computer-based data a variety of database on the relational database structure. One of the objectives in setting up a spatial database is to store only necessary information and to enable efficient and effective access to the data. Efficient organization of data is important for spatial data analysis. In particular spatial database combines data into a single integrated database rather than storing each layer in discrete files. It has the built in rules which help to maintain the integrity of the database and reduce database maintenance.Most spatial databases allow representing easy geometric objects similar to features, lines and polygons. Some spatial databases manage more tricky structures corresponding to 3D objects, topological coverages, linear networks. Database systems use indexes to swiftly look up values and the best way that most databases index information is not foremost for spatial queries alternatively, spatial databases use a spatial index to pace up database operations.

III. EXISTING SYSTEM

Spatial queries with key words have no longer been generally explored. Previously, the neighborhood has sparked enthusiasm in learning keyword search in relational databases. Like R-trees, the IR^2-tree preserves objects spatial proximity, which is the key to fixing spatial queries efficaciously. On the other hand, like signature files, the IR^2-tree is ready to filter a considerable component of the objects that don't contain the entire question key terms, consequently enormously lowering the number of objects to be examined.

A. *IR^2-Tree*

The IR^2-tree combines the R-tree with signature documents. The IR^2-tree [3][6] is an R-tree where each (leaf or nonleaf) entry E is augmented with a signature that summarizes the union of the texts of the objects in the subtree of E.

B. R-Tree

The R-tree, one of the crucial popular entry methods for rectangles is founded on the heuristic optimization of the area of the enclosing rectangle m each and every miner node. R*-tree which corporate a combined optimization of subject, margin and overlap of each and every enclosing rectangle m the listing using a standardized proven m an exhaustive performance evaluation. It became out that the R*-tree certainly outperforms the existing R-tree.

C. Signature Files

Signature file refers to a hashing based frame work, whose instantiation is known as superimposed coding.

D. Challenges of Existing System

- The existing system failed to provide real time solutions on complex inputs.
- The real nearest neighbour lies particularly far away from the query point, at the same time the entire near neighbours are lacking as a minimum one of the vital question key phrases.

IV. PROPOSED SYSTEM

Within the proposed procedure design of a variant of inverted index that's optimized for multidimensional aspects, and is consequently named the spatial inverted index (SI-index). This access process effectually accommodates point coordinates right into a conventional inverted index with small additional area, as a result of a tender compact storage scheme. It will sequentially merge more than one lists very much like merging normal inverted lists by way of ids. Alternatively, it may also leverage the R-trees to browse the facets of all vital lists in ascending order of their distances to the question point. As validated through experiments, the SI-index greatly outperforms the IR2-tree in query effectivity, mostly by way of a aspect of orders of magnitude.

A. Spatial Data Analysis

Geographical or spatial data plays a vital role in many parts of daily life. Either directly, as in the use of a map for navigating around a city, or indirectly, where we use resources like water or gas we are dependent on information where they are located and their attributes. Geographical Information System (GIS) play a key role in this context. GIS provides a means of generating, modifying, managing, analyzing and visualizing spatial data.

B. Data and Data Models

A model is simply a means of representing ‘reality’ and spatial data models provide abstractions of spatially referenced features in the real world. The representation of real-world features are often divided into two categories called entity and fields.

Here as an example of entity is the locations and the values of the location are called fields.

Entity: Entities are conceptually distinct objects like point locations, roads, or administrative boundaries. The location here for example is Hitechcity, Tajbanjara hotel. The entity may be any keyword or query that the user searches

Ex: Hitechcity, Tajbanjara

Fields: Fields convey the idea of values of some property at all locations. Here in this the values are the geo-referenced values called latitude and longitude values.

Ex: fields of entity called Hitechcity
Latitude value: 17.447412 Longitude value: 78.376230
List of items: forummall, majeeramall
Ex: fields of entity called Tajbanjara
Latitude value: 17.391636 Longitude value: 78.440065
List of items: Steak, franky, Spaghetti

The two well known data models are Raster and Vector Representation. The information is stored using these two data models.

C. Grid

A grid is a regular tessellation of a manifold or 2-D surface that divides it into a series of contiguous cells, which can then be assigned unique identifiers and used for spatial indexing purposes. A wide variety of such grids have been proposed or are currently in use, including grids based on "square" or "rectangular" cells, triangular grids or meshes, hexagonal grids and grids based on diamond-shaped cells. There are two types of representation they are called raster and vector data. Currently we use vector data representation

1) *Vector Data:* Vector data comprises points(with x- and y-coordinates), lines(line segments (or arcs) connected by points) and area polygons (lines with the same start and end point). Vector data can be stored as what are sometimes called spaghetti data-that is , strings of unconnected line segments. However, explicit information on the relationships between the objects reduces the computational demands of subsequent analyses.

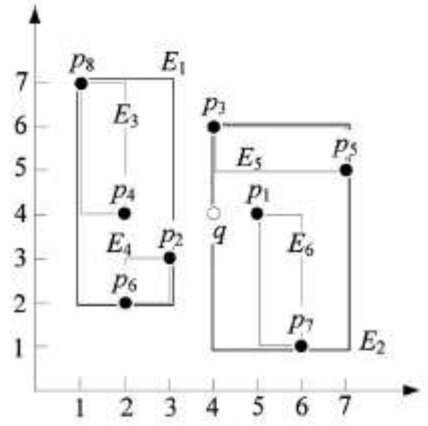

Fig 1: Representation of vector data in polygons

In the proposed system the vector data with Cartesian coordinates can be represented in the form of polygons in the above figure.

Here q is the user query or keyword. $P_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8$ are the nearest neighbours and $E_1, E_2, E_3, E_4, E_5, E_6$ are the polygons in which the given query may present. In this paper when the user enters the query q, then the searching takes place in the database. Here every spatial data has the unique ids and are stored in the database in an ascending order in the database. These have the relationship among them through these ids. Here in this project the Cartesian coordinates are represented by ids and their attributes are connected by the unique values. For the query point q the real nearest neighbours are obtained by indexing. In this project the indexing is done through hash mapping.

D. *Geo-Referencing*

The location of spatial objects is usually recorded using some kind of spatial referencing system such as longitudes and latitudes, or eastings and northings using some kind of national grid system. This can also be called as geographic coordinate system. The latitude and longitude values are called as Cartesian coordinates.

1) Latitude: The "latitude" (abbreviation: Lat., φ, or phi) of a point in the world's floor is the angle between the equatorial airplane and the straight line that passes by means of that factor and through (or just about) the centre of the Earth.
Ex: 17.447412

2) *Longitude:* The "longitude" (abbreviation: Long., λ, or lambda) of a point on the Earth's surface is the angle east or west from a reference meridian to another meridian that passes through that point. Ex: 78.376230

E. *Spatial Inverted Index*

The Spatial Inverted Index as in [12] is basically a compressed variation of an I-index with embedded coordinates. Query processing with an SI-index can be executed either by merging, or at the side of R-trees in a distance shopping manner. An SI-index is not more than a com-pressed variation of an traditional inverted index with coordinates embedded, and therefore, can be queried by way of merging a couple of inverted lists.

The inverted index is the list of words, and the documents in which they appear. In the web search example, you provide the list of words (search query), and Google produces the documents (search result links).They are both indexes - it's just a question of which direction you're going. Forward is from documents-to-words, inverted is from words-to-documents. An Inverted Index is a structure used by search engines and databases to make search terms to files or documents and the speed writing the document to the index for searching the index later on. There are two versions of an inverted index, a record-level index which describes which documents contain the term and a fully inverted index which describes both the document a term is contained in and where in the file it is.

Example

{0} - "Turtles love pizza"
{1} - "I love my turtles"
{2} - "My pizza is good"

Then we can store them in a Inverted Indexes like this:

	Record Level	Fully Inverted
"turtles"	{0, 1}	{ (0, 0), (1, 3) }
"love"	{0, 1}	{ (0, 1), (1, 1) }
"pizza"	{0, 2}	{ (0, 2), (2, 1) }
"i"	{1}	{ (1, 0) }
"my"	{1, 2}	{ (1, 2), (2, 0) }
"is"	{2}	{ (2, 2) }
"good"	{2}	{ (2, 3) }

The record level sets represent just the document ids where the words are stored, and the fully inverted sets represent the document in the first number inside the parentheses and the location in the document is stored in the second number.

So now if you wanted to search all three documents for the words "my turtles" you would grab the sets (looking at record level only):

"turtles"	{0, 1}
"my"	{1, 2}

Then one can intersect those sets, coming up with the only matching set being 1. Using the Fully Inverted Index would also let us know that the word "my" appeared at position 2 and the word "turtles" at position 3, assuming the word position is important your search. In this paper Hash tables are used for indexing data.

Pseudo code for Inverted Index

```
public class InvertedIndex
{
      //Here the keywords are taken into the hashset
  private readonly Dictionary<string, HashSet<int>> _index
  = new Dictionary<string, HashSet<int>>();
  private readonly Regex _findWords = new Regex(@"[A-
  Za-z]+");

  public void Add(string text, int docId)
  {
      // if the given keyword matches
    var words = _findWords.Matches(text);

    for (var i = 0; i < words.Count; i++)
    {
      var word = words[i].Value;

      if (!_index.ContainsKey(word))
        _index[word] = new HashSet<int>();

      if (!_index[word].Contains(docId))
        _index[word].Add(docId);
    }
  }

  public List<int> Search(string keywords)
  {
      //search for the given keyword
    var words = _findWords.Matches(keywords);
    IEnumerable<int> rtn = null;

    for (var i = 0; i < words.Count; i++)
    {
      var word = words[i].Value;
      if (_index.ContainsKey(word))
      {
```

```
                    rtn = rtn == null ? _index[word] :
            rtn.Intersect(_index[word]);
             }
             else
             {
                   //return the list of documents
               return new List<int>();
             }
           }

           return rtn != null ? rtn.ToList() : new List<int>();
       }
   }
```

In this work the administrator enters the Cartesian coordinates, fields and names in to the database. Here the administrator has all the permissions for storing, maintaining, modifying the data. All the places and values that are given by the users are maintained in the database. The user has to get registered by providing all the details that are required. All the users can access by providing their login details. The users can search the places which are nearest to them like hotels restaurants hospital etc and the required list of items in it. If the given keyword search matches the existing keywords in the database, then that document will be retrieved. If not, no record found information will be displayed to the users. Through this the user will only get the real nearest neighbour and also reduces the keyword search time.

V. Results

A comparison study has done on IR^2-tree and Inverted indexes. Indexing through Spatial Inverted (SI) index gives a better performance than IR^2-trees . The time consumption which means the query response time is better performed through SI-index rather than IR^2- Trees. Here we have examined a data set of seventy thousand points to test the performance of query execution in spatial database in comparison to IR^2- trees. Their locations are uniformly distributed and the nearest neighbours are obtained with in less time. The mapping to different tables in the database through inverted indexes gives the better performance through hash mapping. 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. This method has 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 has the ability to perform keyword-augmented nearest neighbor search in time

The SI-index, accompanied by the proposed access method, has presented itself as an excellent trade-off between time and query efficiency. Compared to IR^2-trees, it consumes significantly less time, and yet, answers queries

D. Advantages of the Proposed System

- Distance searching is easy with Spatial inverted indexes to output data points in ascending order of their distances.
- It is straight forward that compression scheme can be extended to any dimensional house.

VI. Conclusion

It has been observed that 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 there is a remedy for the situation by developing an access method called the spatial inverted index (SI-index). The SI-index has the ability to perform keyword-augmented nearest neighbor search in a significant manner rather than IR^2-trees.

Acknowledgment

I feel privileged to offer my sincere thanks and deep sense of gratitude to Dr. C. Kiranmai (Principal & Professor), Dr. B. Raveendra Babu (Professor and Head of the Department, CSE Department),Mrs.R.Vasavi (Assistant Professor, Project Coordinator, CSE department), of VNRVJIET for extending their co-operation in doing this project. I express my sincere gratitude to my Project Guide, Mr. Y. Sagar,(Associate Professor, CSE department)**,** of VNRVJIET for valuable time and guidance ,throughout the M.Tech Project. My special thanks to all the Faculty members of the Department for their valuable advices at every stage.

References

[1] Yufei Tao and Cheng Sheng “Fast Nearest Neighbor Search with Keywords” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 4, APRIL 2014.

[2] 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.

[3] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, “The R - tree: An Efficient and Robust Access Method for Points and Rectangles,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 322-331, 1990.

[4] 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.

[5] 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.

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

[7] 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.

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

[9] Y.-Y. Chen, T. Suel, and A. Markowetz, “Efficient Query Processing in Geographic Web Search Engines,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 277-288, 2006.

[10] 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.

[11] 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.

[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.