**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**
**IMPACT FACTOR: 5.258**

***IJCSMC, Vol. 5, Issue. 9, September 2016, pg.89 – 92***

# Review on Spatial Database Using R Tree

**Miss. Harshali Panchal, Mr. Ghungrad S. R.**

CSE, Dr. Babasaheb Ambedakar University, India
CSE, Dr. Babasaheb Ambedakar University, India
panchalhr7@gmail.com; shesh_01@rediff.com

***Abstract— At present, for novel forms of queries many modern applications call that want to discover objects fulfilling equally a spatial predicate, and a predicate on their linked texts. Such as, in place of bearing in mind all the restaurants, a nearest neighbor query would as an alternative raise for the restaurant that is the closest amongst those whose menus contain "steak, spaghetti, brandy" all in unison. At this time the best solution to such queries is based on the IR$^2$ -tree, has a few deficiencies that acutely impact its efficiency. And the R-tree, one of the most popular access methods for rectangles, is based on the heuristic optimization of the area of the enclosing rectangle m each inner node By running numerous experiments m a standardized tested under highly varying data, queries and operations.***

***Keywords— nearest neighbor search, keyword search, spatial index, B+ tree.***

## INTRODUCTION

A spatial database offers fast access to those objects based on different selection criteria and manages multidimensional objects like as points, rectangles, etc. The significance of spatial databases is reflected by the expediency of modeling entities of reality in a geometric manner. For instance, locations of restaurants, hotels, hospitals and so on are frequently represented as points in a map, while larger extents for example parks, lakes, and landscapes often as a amalgamation of rectangles. Many functionalities of a spatial database are helpful in various ways in exact contexts. Such as, in a geography information system, range search can be deployed to discover all restaurants in a certain area, while nearest neighbor retrieval can discover the restaurant closest to a known address.

Nowadays, the widespread use of search engines has made it realistic to write spatial queries in a brand new way. Typically, queries focus on objects' geometric properties only, for instance whether a point is in a rectangle, or how close two points are from each other. We have seen some modern applications that call for the capability to choose objects based on *both* of their geometric coordinates and their associated texts. As, it would be moderately helpful if a search engine can be used to discover the nearest restaurant that offers "steak, spaghetti, and brandy" all at the same time. Note that this is not the "globally" nearest restaurant (which would have been returned by a traditional nearest neighbor query), but the nearest restaurant among *only* those provided that all the demanded foods and drinks.

There are simple ways to prop up queries that combine spatial and text features. Such as, we could first fetch all the restaurants whose menus contain the set of keywords, for the above query {steak, spaghetti, brandy}, and then from the retrieved restaurants, find the nearest one. In the same way, to the query point until encountering one whose menu has all the key- words, one could also do it reversely by targeting first the spatial conditions—browse all the restaurants in ascending order of their distances.

They will fail to provide real time answers on difficult inputs, this is the main negative aspect of these straightforward approaches. A classic example is that the real nearest neighbor lies quite far away from the query point, whereas all the closer neighbors are missing at leastone of the query keywords.

The IR$^2$-tree, nevertheless, also inherits a disadvantage of signature files: *false hits*. Explicitly, a signature file may still direct the search to some objects by reason of its conservative nature, even though they do not have all

 

the keywords.

The punishment thus caused is the need to *verify* an object whose rewarding a query or not cannot be resolute using only its signature, but requires loading its full text description, which is exclusive due to the resulting random accesses.

It is significant that the false hit problem is not definite only to signature files, but also exists in other methods for approximate set membership tests with dense storage. Consequently, the problem cannot be remedied by simply replacing signature file with any of those methods.

And also Now Spatial data come up in many applications, including: Cartography, Computer-Aided Design, computer vision also in robotics traditional databases.[3] i.e. it increase the importance of R tree. Internet search engines now are famous for the keyword based search pattern whereas traditional database management systems proffer powerful query languages (as they do not allow keyword-based search).[4]

**Morality of R-trees**

An R-tree is nothing but B+-tree which is similar to structure which stores multidimensional rectangles while complete Objects lacking clipping them or else transforming them to higher dimensional points before A non-leaf node which hold entries in the form (cp, Rectangle) here the address of a child node in the R-tree is cp and the minimum bounding rectangle of all rectangles is Rectangle which are entries in that child node A leaf node contains entries of the form (Old, Rectangle) here Old refers to a record in the database, which is describing a spatial object as well as Rectangle is the enclosing rectangle of that spatial object Leaf nodes containing entries of the form (data object, Rectangle) are also achievable This will have not an effect on the basic structure of the R-tree.[2]

R-tree is absolutely dynamic, insertions and deletions can be intermixed with queries and no periodic global reformation is required obviously, the structure must permit overlapping directory rectangles. Hence, it cannot assurance that only one search path is obligatory for an exact match query.

The parameters which are vital for the retrieval performance are as follows, interdependences between different parameters and optimization criteria are analyzed.

A. The region roofed by a directory rectangle should be minimized explicitly. The region covered by the bounding rectangle but not covered by the enclosed rectangles, the dead space, should be minimized Thus will take on higher levels if it improves performance since decisions which paths have to be traversed.
B. The overlap between directories rectangles should be minimized. It helps to decrease the number of paths to be traversed.
C. The border (margin) of a directory rectangle should be minimized. Here the border (margin) is the sum of the lengths of the edges of a rectangle, the object with the smallest margin is the square, by considering fixed area therefore minimizing the margin rather than the area, the directory rectangles will be produced more quadratic. Basically, with only little variance of the lengths of the edges, clustering rectangles into bounding boxes will decrease the area of directory rectangles.
D. Storage consumption should be optimized. As the height of the tree higher storage utilization will normally reduce the query cost will be kept low obviously. Since the concentration of rectangles in several nodes, the query types with large query rectangles are predisposed more will have a stronger effect while the number of found keys is higher.

Before R tree, the IR2-tree was the first access method for answering NN queries with keywords.[1] And The R-tree can be an object hierarchy in the form of a balanced structure inspired by the B+-tree [5]. In each node the maximum number of entries is termed its *node capacity* or *fan-out* and may be different for leaf and non leaf nodes. R-tree can be used to index a space of arbitrary dimension and arbitrary spatial objects rather than just points.[5]

Following are some existing studies and approaches related for optimization of broadcasting methods for improving the routing performance:

Yufie Tao and Cheng Sheng[1], urbanized a new access method called the spatial inverted index that extends the conventional inverted index to cope with multidimensional data, as well as comes with algorithms that can answer nearest neighbor queries along with keywords in real time. we propose a variant of inverted index so as to optimized for multidimensional points, and is thus named the spatial inverted index (SI-index). This access method productively incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. In the meantime, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead.

 

Cao et al. [2], planned collective spatial keyword querying, they present the new problem of retrieving a group of spatial objects, each associated with a set of keywords. We develop approximation algorithms with provable approximation bounds and exact algorithms to solve the two problems.

Lu et al. [3], combined the notion of keyword search with reverse nearest neighbor queries. propose a hybrid index tree called IUR-tree (Intersection-Union R-Tree)that effectively combines location proximity with textualsimilarity. Based on the IUR-tree, we design a branch-and-bound search algorithm.

Cong et al.[4], proposed the concept of prestige-based spatial keyword search. The central idea is to evaluate the similarity of an object p to a query by taking also into account the objects in the neighborhood of p.

G. Cong, C.S. Jensen, and D. Wu [5] proposed a approach that computes the relevance between the documents of an object p and a query q. This relevance score is then integrated with the Euclidean distance between p and q to calculate an overall similarity of p to q. The few objects with the highest similarity are returned. In this way, an object may still be in the query result, even though its document does not contain all the query keywords.

I.D Felipe, V. Hristidis and N. Rishe,[6], object texts are utilized in evaluating a boolean predicate, i.e., if any query keyword is missing in an object's document, it must not be returned. Neither approach subsumes the other, and both make sense in different applications. As an application in our favor, consider the scenario where we want to find a close restaurant serving "steak, spaghetti and brandy", and do not accept any restaurant that do not serve any of these three items. In this case, a restaurant's document either fully satisfies our requirement, or does not satisfy at all.

Y.-Y. Chen, T. Suel, and A. Markowetz[8], have studied efficient query processing in geographic web search engines. They discussed a general framework for ranking search results based on a combination of textual and spatial criteria, and proposed several algorithms for efficiently executing ranked queries on very large collections. They integrated their algorithms into an existing high-performance search engine query processor and evaluated them on a large data set and realistic geographic queries. Their results show that in many cases geographic query processing can be performed at about the same level of efficiency as text-only queries.

V. Hristidis and Y. Papakonstantinou [10], presented DISCOVER, a system that performs keyword search in relational databases. It proceeds in three step. First it generates the smallest set of candidate networks that guarantee that all *MTJNT*'s will be produced. Then the greedy algorithm creates a near-optimal execution plan to evaluate the set of candidate networks. Finally, the execution plan is executed by the DBMS.

According to Jo˜ao B. Rocha-Junior?, Orestis Gkorgkas, Simon Jonassen, and Kjetil Nørv°ag a spatial location and a set of keywords, a top-k spatial keyword query returns the k best spatio-textual objects ranked according to their proximity to the query location and relevance to the query keywords.

There are many applications handling huge amounts of geotagged data, such as Twitter and Flickr, that can benefit from this query. Unfortunately, the state-of-the-art approaches require non-negligible processing cost that incurs in long response time. In this paper, we propose a novel index to improve the performance of top-k spatial keyword queries named Spatial Inverted Index (S2I).

Our index maps each distinct term to a set of objects containing the term. The objects are stored differently according to the document frequency of the term and can be retrieved efficiently in decreasing order of keyword relevance and spatial proximity. Moreover, we present algorithms that exploit S2I to process top-k spatial keyword queries efficiently. Finally, we show through extensive experiments that our approach outperforms the state-of-the-art approaches in terms of update and query cost.

The S2I maps each term t to an aggregated R-tree (aR-tree) or to a block that stores the spatio-textual objects p that contain t. The most frequent terms are stored in aR-trees, one tree per term. The less frequent terms are stored in blocks in a file, one block per term. Similarly to a traditional inverted index, the S2I maps terms to objects that contain the term.

However, we employ two different data structures, one for less frequent terms and another for more frequent terms that can be accessed in decreasing order of keyword relevance and spatial proximity efficiently.

## CONCLUSION

Evidently, As The number of false hits can be really large when the object of the final result is far away from the query point, or the result is simply empty. Hence R-tree can be utilized as an access method in database systems organizing together, multidimensional points and spatial data. This paper is a review of R tree.

 

# 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] Norbert Beckmann, Hans-Peter begel Ralf Schneider, Bernhard Seeger "The R*-tree:An Efficient and Robust Access Method for Points and Rectangles+" Praktuche Informatlk, Umversltaet Bremen, D-2800 Bremen 33, West Germany.

[3] Ibrahim Kamel and Christos Faloutsos "Hilbert R-tree: An Improved R-tree Using Fkactals" University of Maryland College Park, MD 20742 and Institute for Systems Research (ISR) University of Maryland College Park, MD 20742.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

 