ISSN: 2321-7782 (Online)
Volume 2, Issue 1, January 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

Linked R-Tree for Spatial Data

Subba Rao Voore[1]
M.Tech.(CSE),JNTU, Hyderabad
Asst.Professor,
Dept. of Computer Science & Engg
RGCER, Nagpur, India

Shashibhushan Ivaturi[2]
M.Tech.(CSE) IIIT Hyderbad
Software Programmer Analyst,
Streamnet (India) Pvt. Ltd
Bangalore, India

Abstract: This research paper is proposing an enhancement to the spatial data structure R-tree, called Linked R-tree, intended to provide an efficient searching mechanism. This paper gives importance of the idea of theoretical approach of the data structure and an idea of implementation of linked r-tree.

Keywords: spatial data; geographic information systems; r-tree, linked r-tree

I. INTRODUCTION

The Spatial Data is the data or information that identifies the geographic location of features and boundaries on Earth, such as natural, constructed features and oceans. Spatial data that have a spatial component, it means that data are connected to a place in the Earth. Spatial data is often accessed, manipulated or analyzed through Geographic Information Systems (GIS). This research paper is proposing an enhancement to the spatial data structure R-tree, called Linked R-tree, intended to provide an efficient searching mechanism. This paper gives importance of the idea of theoretical approach of the data structure and an idea of implementation of linked r-tree[1]

a) Linked R Tree Features

- Reduced Complexity in Finding Features
- Reduced Complexity in moving through the regions
- Linked at multiple levels

II. EXISTING SYSTEM

a) R- Tree

R-Trees are tree data structures used for spatial access methods, i.e., for indexing multi-dimensional information such as geographical coordinates, rectangles or polygons. A common real-world usage for an R-tree might be to store spatial objects such as…

- Restaurant locations typical maps are made of: streets, buildings, outlines of lakes, coastlines, etc.
- Find answers quickly to queries such as "Find all museums within 2 km of my current location"
- Retrieve all road segments within 2 km of my location.

In the below figure Rectangle R1 has sub rectangles R3, R4, R5. R1 is a big rectangle covering a region, R3, R4, R5 in turn covering sub regions in R1. If number of features in region R3 is increasing, after certain limit R3 will be divided into two subregions. There will be a maximum limit for each region.

Each rectangle will store the coordinates(x axis latitude, y axix longitude) of its top left corner and bottom right corner. If any feature's coordinates come within this region then that feature information is added to the corresponding region.

Region{

Coordinates: {x1,y1 } {x2,y2}

List of Features: resturants, hospitals(each feature will have its coordinates) etc

Link to its sub regions.

}

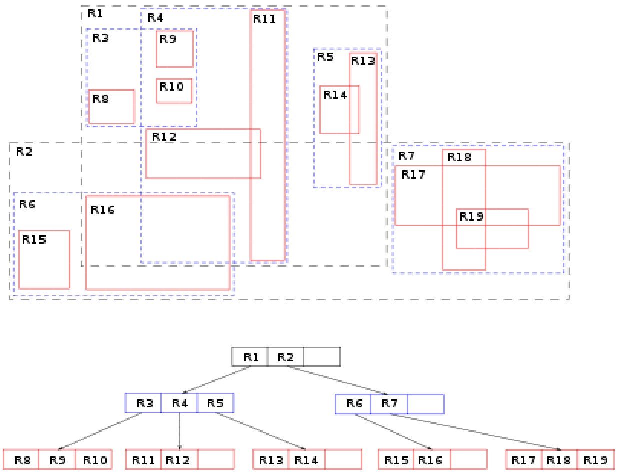


Fig1. Rectangle R1 has sub rectangles R3, R4, R5.

Ex: R1 corresponds to a region Nagpur R3, R4, R5 will correspond to localities in Nagpur .Each of those localities will have the information of roads, shops, inform of points lines etc [2]

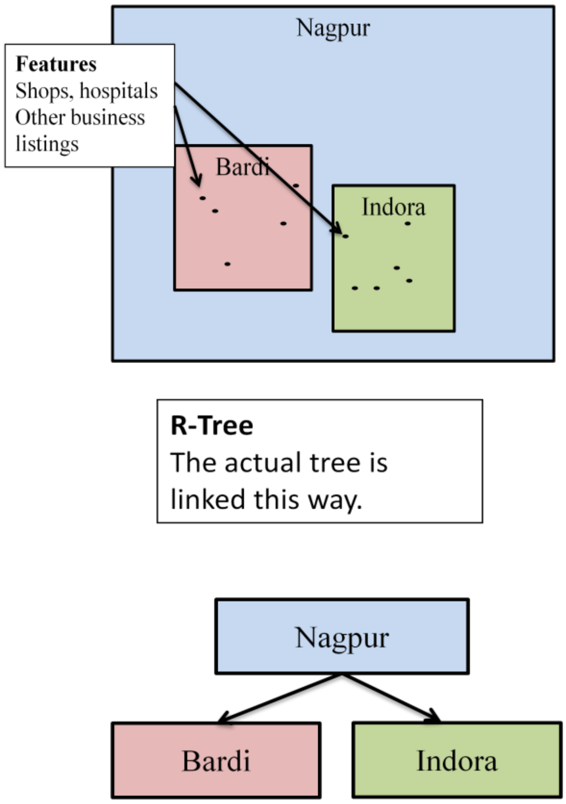


Fig2. Rectangle R1 has sub rectangles R3, R4, R5.

In the above figure, root node, main rectangle represents Nagpur region. It has direct links to its childs Bardi and Indora. These regions contain the latitude and longitude information of the features in those regions. Hence, the tree has two levels.

If the number of regions in Bardi increase certain limit again this region can be divided further into two or more sub regions. Then Bardi will have link to its sub regions.

b) *Existing Search*

- Find all features within the proximity of P
- Needs to access two regions R2 and R3(in proximity of p).

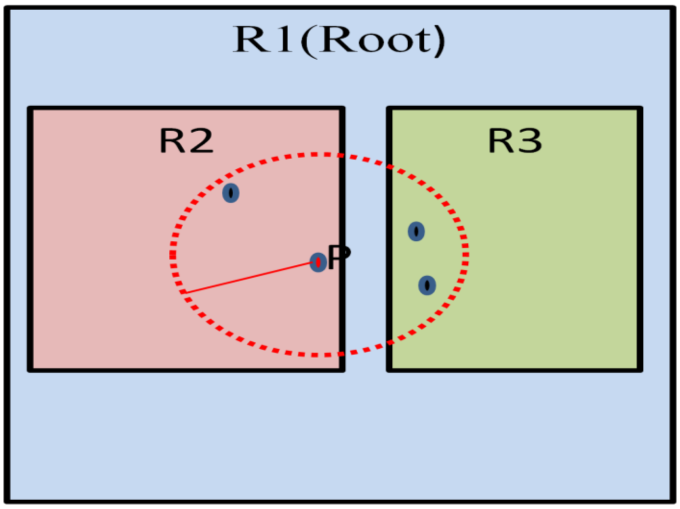


Fig3. Region R1 has sub regions R2, R3

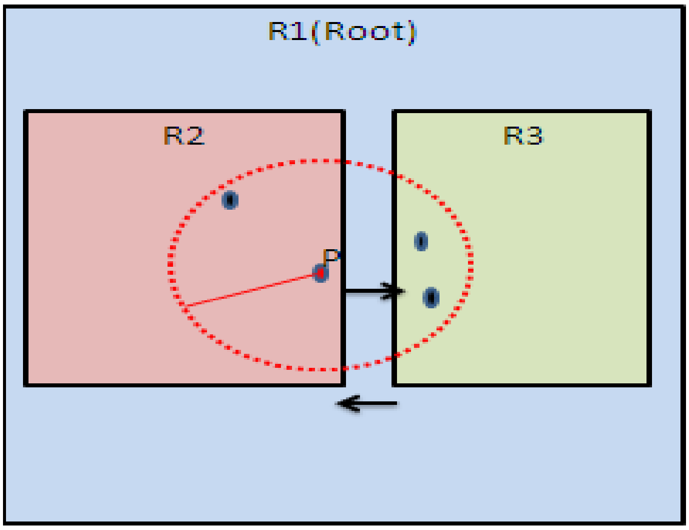


Fig4. Region R1 has sub regions R2, R3 with link

III. PROPOSED SEARCH

Linked Rtree, if there are any regions in the same level and if those regions are just next to that region, those regions are linked. For example, Among the regions Nagpur, Bardi and Indora…. Bardi and Indora are at the same level, but not Nagpur (this is at city level). So Bardi is linked to Indora from the East.

1) Advantage of our Linked RTree over RTree

In the above figure for example, consider those two regions as R2, R3 as Bardi and Indora. In order to find out all the restaurants near the feature P, as mentioned above, some features will be nearer but fall under Bardi region. In order to access Bardi, Again you need to start from root and come down to Bardi, because there is no link between Bardi and Indora.

Suppose these two regions are in some 10^{th} or 15^{th} level and if you have to access multiple regions, for each region you need to start from the root and descend the tree[3]

International Journal of Advance Research in Computer Science and Management Studies
Volume 2, Issue 1, January 2014 pg. 215-222

2) Linked –R-tree Structure

Adjacent regions in the same level are linked to each other

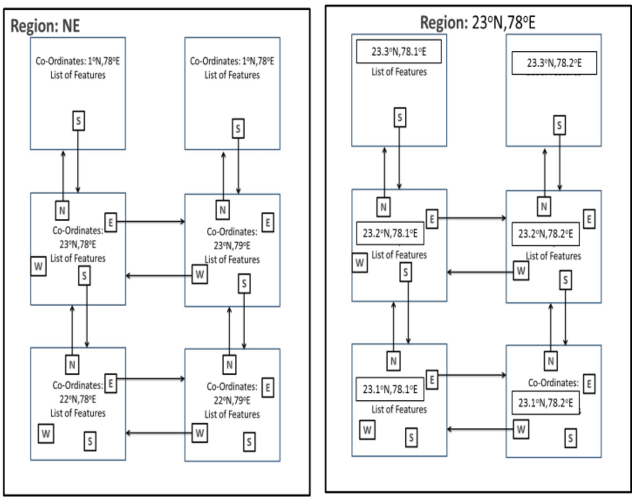


Fig5. Region are linked with other regions

3) Algorithm for Data Cell Structure

```
StructDataCell {
        Float co-ordinates;
DataCell *SubRegion;
DataCell *East;
DataCell *West;
DataCell *North;
DataCell *South;
List<Feature> features;
  }
Struct Feature {
            String type; //Point or Polygon or Line
DataCell *Next;
```

Each region will have four links to its adjacent regions (East, West, North, South).

- Need to descend from root in order to access R3
- Need to descend from root N times to access N regions

IV. FUTURE SEARCH

The structure design is simple. Based on the co-ordinates (latitude, longitude) entire globe is divided into four regions (North-East (NE), South-East (SE), North-West (NW) and South-West (SW)) and each region contains 90 * 180 data cells. Each data cell represents co-ordinates of a place and is linked with its neighboring cells [4]

The attributes of a data cell are its co-ordinates, list of features present in that region and links to its neighboring cells from four directions .i.e., North, East, West and South.

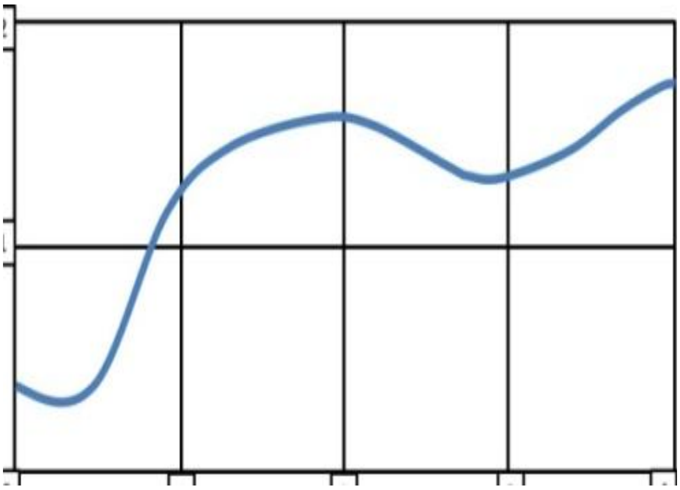
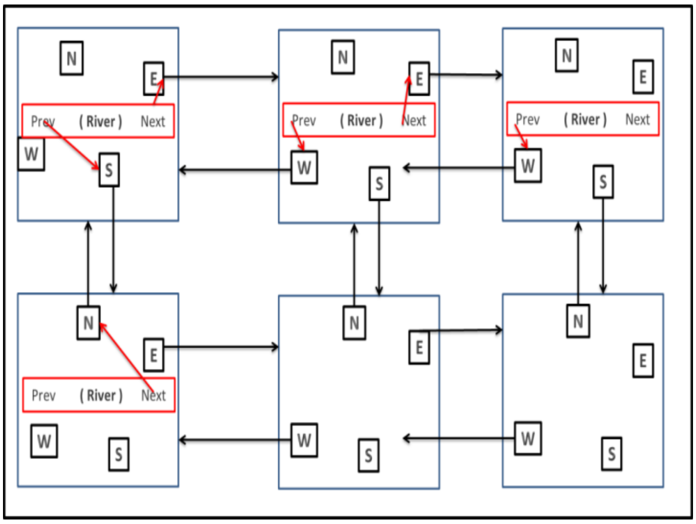


Fig.7 River flows according to regions Fig.



8. Every regions are linked with previous and next links

In the above example the river passes through regions D, A, B and C. The river passes through a certain region (D according to Rtree) and enters another region (A).

Hence in the linked R-Tree for every feature in a region has a link to its next region if it is extending further. Here after region D, feature river's next region points to D->north, which is pointing to region A and the river's next region in A is A->East and previous region is A->South.

The existing R-Tree structure cannot direction based feature search efficiently. Co-ordinates of every feature of that region are compared to check whether the feature is located towards required direction from the given location.

Example:

(i) List all the hotels located near the railway station

(ii)List all the restaurants to the left side of the highway

- R2 is linked to R3
- Representation R2->East = R3

1) Algorithm Feature Search

List<Feature>findFeatures(Feature f, DataCellloc,

int distance)

```
{
        List<Feature>fList;
Begin
 - For each cell in loc.neighbors(East, West,
                    North, South)
-   For each feature in cell.features
Begin
-   If(feature.type equals f.type)
fList.add(feature);
End
-   Return fList
End
```

2) Importance of Granularity

Granularity is the extent to which a system is broken down into small parts. This is a generalized data structure that can be used for any kind of spatial data. If consider world map, at the root level four connected cells (NE, SE, NW and SW) exist.

The co-ordinates at the root level are natural numbers. Based on the requirement, each cell in turn consists of 90 * 180 connected cells and each of these cells contain cell network based on the granularity of the latitude and longitude[2].

For example,

In the following figure shows the NW regions links with NE region with prescribed cell**.**

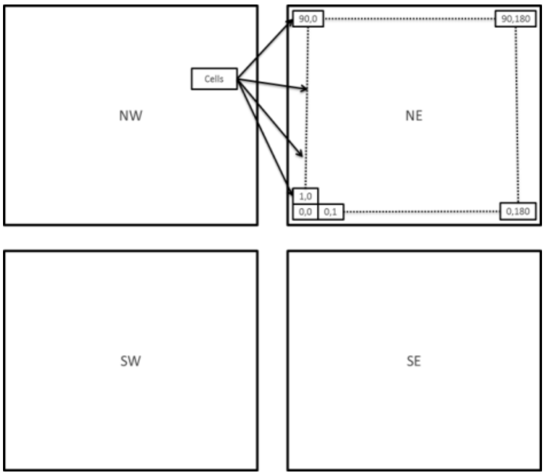


Figure 9. Linked R-tree at root level

Proposed

- Linked R-Tree provides efficient way of searching features in the proximity of a location.
- From the given location the data cell towards the required direction can be accessed directly and the features located in that region can be listed.

Key Features

- Linked R-Tree is designed as a compromise between R-Tree and R+-Tree, showing the difference in search performance.
- Time Complexity of feature search is always O(n).
- Efficient in loading the features.
- Efficient in displaying a required feature without loading parts of its neighbors.
- Efficient in searching features in proximity.

3) Overlapping

Like R+-Trees, linked R-Trees avoid overlapping of internal nodes by inserting an object into multiple leaves if necessary. Coverage is the entire area to cover all related rectangles. Overlap is the entire area which is contained in two or more nodes. Minimal coverage reduces the amount of "dead space" (empty area) which is covered by the nodes of the R-tree.

Minimal overlap reduces the set of search paths to the leaves (even more critical for the access time than minimal coverage). Efficient search requires minimal coverage and overlap. Two more similarities between R+ - Tree and linked R-Tree is that,

- Nodes are not guaranteed to be at least half filled
- An object may be stored in more than one node
- But unlike R+-Trees, objects can be stored in non- leaf nodes in linked R-Tree.

Whereas, in linked R-Trees, instead of descending the tree multiple times for each object, descend once to reach the location and from there continue traversing to its neighboring regions with in the proximity.

Time Complexity (nearest neighbors)

Table -1

	Best Case	WorstCase
R-Tree	O(logn)	O(nlogn) or O(n)
Linked R-Tree	O(logn)	O(logn)

Figure 10. Linked R-tree at root level)

Linked R-Trees can handle large data sets in a better way, because they can access the neighboring regions(rectangles) without descending (starting from the root) for every region.

4) Future Work

Linked T-tree is a very basic data structure, which should tested, generalized and enhanced more.

Our future work is focused on the following:

- Implementing and testing the features. Deciding the precision of the co-ordinates at each level of the structure based on the data.
- Placement of the features depending on the scope of its visibility at each view level of the map

5) Performance Analysis Comparison – R-Tree Vs Linked R-Tree

- Operations (insert, delete, update, split, merge) of linked R-Tree same as that of R-Tree, Shows the improvement in performance of search operation, after splitting the splitted nodes are linked with each other.
- Searching algorithm different from the one used in R-trees. (In R-Tree, in order to search for nearest neighbors, the idea is to first decompose the search space into disjoint sub-regions and for each of those descend the tree until the actual data objects are found in the leaves.)
- Notice that a major difference with R-trees is that in the latter sub-regions can overlap, thus leading to more expensive searching. Hence, it is **not possible to guarantee good worst-case performance** on the other hand with most kinds of data the update algorithms will maintain the tree in a form that allows the search algorithm to eliminate irrelevant regions of the indexed space, and examine only data with in the same region(rectangle)[1]

V. CONCLUSION

This research paper is focusing on importance of linked r-tree over r-tree. The paper proposing an enhancement to the spatial data structure R-tree, called Linked R-tree, intended to provide an efficient searching mechanism of a particular place or region in the earth. This paper gives importance of the idea of theoretical approach of the data structure and an idea of implementation of linked r-tree for more enhancing the searching algorithm in a better way.

VI. Acknowledgment

It is a great opportunity for me to write about subject like "Linked R-Tree for Spatial Data". At the time of preparing this paper I am gone through different books and websites which help me to get acquainted with new topics. I acknowledge with gratitude to colleagues and co-authors, who has always been sincere and helpful in making me understanding the different system of legal research and conceptual problems in my paper.

I acknowledge my sincere thanks to Head of the Department, Principal and Management giving infrastructure and facilities for writing this research paper.

References

1. Performance Analysis of R*-Trees with Arbitrary Node Extents Yufei Tao and DimitrisPapadias
2. R-Trees. A Dynamic Index StructureFor Spatial Searching http://www-db.deis.unibo.it/courses/SI-LS/papers/Gut84.pdf A new spatial data representation in GIS Zu-Kuan WEI1 Sook-Kyoung CHO1 Jae-Hong KIM2 Hae-Young BAE1.
3. Spatial Data Representation Model Oriented To Geographical Problem Solving Hongjun Su*, Yongning Wen, Min Chen, Hong Tao, JingweiShen
4. 4.Generalising spatial data and dealing with multiplerepresentations http://www.geos.ed.ac.uk/~gisteac/ gis_book_abridged/ files/ch10.pdf
5. Quad Trees http://www.cs.umd.edu/~hjs/pubs/ Samet85k.pdf
6. .R+ Tree: A dynamic index for multi-dimensional objects. http://www.cs.umd.edu
7. R-Treephttp://www-db.deis.unibo.it/courses/SI-LS/papers /Gut84.pdf

Author(s) Profile

Voore Subba Rao is an Associate Professor at the Department of Computer Science and Engineering Department, Rajiv Gandhi College of Engineering and Research, Nagpur (MS) India. He did his M.Tech.(CSE) JNTU, Hyderbad, India. His research interest is in the area of Computer Science, Networking and Databases. He received the M.Tech.in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh, India in 2013.

Shashibhushan Ivaturi is a Software Developer in Streamnet (India) Pvt. Ltd, Bangalore, India. He did his M.Tech.(CSE) IIIT Hyderbad, India. His research interest is in the area of Database and Java. He received the M.Tech.in Computer Science and Engineering from I.I.I.T.,Bangalore India in 2013.Presently he is working as a Software Programmer Analyst, Streamnet (India) Pvt. Ltd, Bangalore, India