

A Navigation Supporting System Using R-Tree

Arpitha Y.C, Bhargavi A.G, Mahima K.T, Nagavi K.K, Meenakshi H.N*

Department of CSE, VVIET, MYSURU

DOI: <https://doi.org/10.21467/proceedings.1.74>

* Corresponding author email: meen.hn79@gmail.com

ABSTRACT

Searching or querying is a basic task carried out in the field of computer science. The task involves searching for the numbers, characters, string and many more. Due to increase in the deployment of computers in the various fields like Computer Aided Design(CAD), gaming, geo spatial science, graphics processing it has become important to search for the specific objects in a spatial image. This kind of spatial search is frequently used in CAD for limiting the data to be worked with, isolate the object and features and otherwise discover patterns in the data. It is used in searching electronic component in Printed Circuit Board(PCB) manufacturing. Further in GIS for extracting the desired data through spatial query which is a process of retrieving the data subset by directly working with the map features. Using conventional data structures, data such as numbers, characters, strings are handled. But when it comes to images, special data structures such as B-Tree, B+-Tree etc are used. Till date, only 2D images are handled but not multi-dimensional images. In this work, we propose an approach to handle multi-dimensional images using R-Tree data structure to support navigation system.

General terms: Navigation supporting system using R-Tree.

Keywords: Navigation system, R-Tree, Bounding box, Spatial image, Spatial indexing,

1 INTRODUCTION

Antonin Guttman proposed R-tree for efficient retrieval of objects from spatial database according to their spatial locations. R-Tree is the most popular dynamic data structure used in spatial indexing application [5]. Spatial index has been one of the active focus areas in recent database research. Several applications such as GIS, CAD use spatial data that in turn require spatial indexing which is matter of interest in research these days. The spatial database consists objects where these objects are of various types like points, lines, polygons etc. these objects are require to be indexed so that searching becomes possible [2]. Spatial indices are used by spatial databases to optimize spatial queries. Conventional index types do not efficiently handle spatial queries such as how far two points differ, or whether points fall within a spatial area of interest. Most commonly B-Tree, B+-Tree and R-Tree are used for handling spatial data. B-Tree is a self-balanced search tree with multiple keys in every node and more than two children for every node. Here, number of keys in a node and number of children for a node is depend on the order of the B-TREE.[3]B-Tree uses a various leveled file to minimize the amount of



storage per user and also keeps the record adjusted with an exquisite recursive calculation. But calculations can get exceptionally intricate especially if numerous qualities are indeterminate and/or if numerous results are joined. A B+ tree is a type of tree which represents sorted data in a way that allows efficient insertion, retrieval and removal of records each of which is identified by a key. [4] In contrast to a B-tree, all records are stored at the leaf level of the tree; only keys are stored in interior nodes. B+ tree provides a reasonable performance for direct access and greater performance for sequential and accesses in range. But The insertion and deletion mechanisms in B+ tree is more complex than other B-trees.

2 R-TREE

R-tree are tree data structures used for spatial access methods, i.e., for indexing multidimensional information such as geographical coordinates, rectangles or polygons. The R-tree was proposed by Antonin Guttman in 1984 [1] and has found significant use in both theoretical and applied contexts. A common real-world usage for an R-tree might be to store spatial objects such as restaurant locations or the polygons that typical maps are made of: streets, buildings, outlines of lakes, coastlines, etc. The R-tree is a height-balanced tree with index records in its leaf nodes containing pointers to data objects. For a large number of data objects, the R-tree itself is also large and should be disk-resident so that a node in an R-tree corresponds to a disk block. The R-tree index structure is completely dynamic because inserts and deletes can be intermixed with searches and no periodic reorganization is required. A spatial database consists of a collection of spatial objects and each object has a unique identifier that can be used to retrieve it. Each node of an R Tree has a variable number of entries. Each entry within a non-leaf node stores two pieces of data: a pointer to a child node and the bounding box of all entries within the linked node. Each entry within a leaf node stores two pieces of data: an implicit pointer to the physical location, and the MBR of spatial data. R-Tree indexes features bounding boxes and cells change every time you insert and update data. The R-tree index is significantly smaller, saving both disk space and the time to build and maintain the index. Leaf nodes in an R-tree contain index record entries of the form

(I, tuple -identifier)

where tuple -identifier refers to a tuple in the database and I is an n-dimensional rectangle which is the bounding box of the spatial object indexed.

$i = (I_0, I_1, I_2, \dots, I_{n-1})$

Here n is the number of dimension and I_i is a closed bounded interval [a,b] describing the extent of the object along dimension i. Alternatively I_i may have one or both endpoints equal to infinity, indicating that the object extends outward indefinitely. Non-leaf nodes contains entries of the form

(I, child -pointer)

where child-pointer is the address of a lower node in the R-tree and I covers all rectangles in the lower node's entries.

2.1 Bounding Box

Arbitrary geometric objects are handled by representing each object by its minimum bounding rectangle, which is a smallest upright rectangle which encloses the object. MBR is the main tuple of the tree, also known as bound boxes, adopted for describing two dimensional spaces. The R-tree access method organizes data in a tree-shaped structure called an R-tree index. The index uses a bounding box. Bounding boxes can enclose data objects or other bounding boxes. It is an expression of the maximum extents of a 2-dimensional object (e.g. point, line, polygon) or set of objects within its (or their) 2-D (x, y) coordinate system, in other words (min(x,y), max(x,y)). In geometry, the minimum or smallest bounding or enclosing box for a point set is the box with the smallest measure (area, volume, or hypervolume in higher dimensions) within which all the points lie.

Bounding boxes are usually stored as a set of coordinates of equal dimension as the bounded object. For example, the minimum bounding box for a two-dimensional circle is a square whose side is equal to the diameter of the circle. The minimum bounding box for a three-dimensional sphere is a cube whose edge is equal to the diameter of the sphere. In this system, the MBR should be as small as possible to cover/ include the object.

2.2 R-Tree construction

An R-Tree is a height-balanced tree with index records containing pointers to data objects. The structure should be designed so that a spatial search requires visiting only small number of nodes. This means that insertion and searching should not require periodic reorganization of the data structure. A spatial data base consists of a collection of terms that represents the spatial objects where, each term has a unique identifier which can be used to retrieve the objects. R tree consists of root node, internal node and leaf node. Each node of the R-tree is represented as (R,O) where, R indicates the rectangle covering an object and represents a pointer to an objects in the database. Each entry consists of a Rectangle and a pointer O. Root node holds a pointer reference to largest region (initially the entire data). The number of entries of a node (except the root) in the tree is between m and M, where, M is the maximum number of entries that can fit in one node and $m \leq M/2$ be a parameter specifying the minimum number of entries in a node.

When an index is built from the top down approach, spatial objects are inserted into the R-tree one at a time. Objects are grouped together in the R-tree according to their spatial proximity to one another.

2.3 Insertion

Inserting an object into the r-tree can be carried out in three different ways. Suppose the given input image has K number of objects, then T indicates the root node from the inserting of a new node in the next level can be done. To illustrate an insertion a hypothetical geospatial image is considered as shown in figure3.3a. It also shows the corresponding R-Tree build assuming $m=2$ and $M=4$.

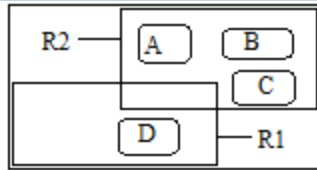


Figure 3.3a

Let X be an object to be inserted into the R-tree. If the object X automatically fit into the existing MBR then it is inserted. For example, the location of the object X is within the first bounding box R1. Therefore, X gets inserted into the leaf node of R1 as shown in figure 3.3b.

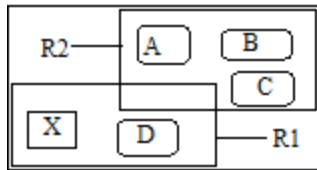


Figure3.3b

Suppose, X object cannot fit into MBR, then one of the MBR has to augmented (enlarged) so that, the object can be included as shown in figure 3.3c.

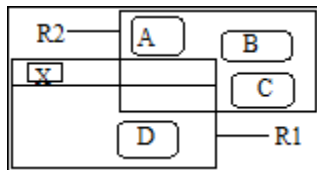


Figure 3.3c. aggregation of R1

Since the object location exceeds the bounding box R1 and also X is spatially near to R1, therefore, R1 is augmented until X object can be included.

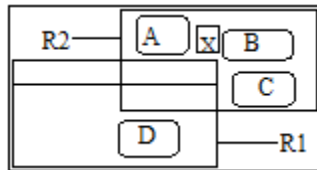


Figure 3.3d. partition of bounding box

It is also possible that, the object may not be near to a bounding box whose node is free but spatially near to a node which may be filled with the other objects. In such case, the nearest MBB gets partitioned. This can be observed in figure 3.3d, where X is not near to R1 but near to R2. Suppose R2 is not free then, it should be split to include the new node X.

2.4 Searching

Given an R-tree whose root nodes is T, find all index records whose rectangles overlap a search rectangle S. If T is not a leaf node, then check each entry in the tree to determine whether the MBR overlaps the search rectangle S. If T is a leaf node then check, whether the MBR overlaps the S. If it overlaps, then that MBR consists of the required Object.

2.4.1 Nearest Neighbor Search for the given object.

Dealing with image and creation of R-tree is discussed in the previous section. The object is given as the input and eventually, all the objects near that specific object is searched. The objects in the image are searched using searching algorithm which is the step by step procedure used to locate specific data among a collection of object. It is considered a fundamental procedure in computing. When searching for data, the difference between a fast application and a slower one often lies in the use of the proper search algorithm.

The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning), or vary based on the local density of points (radius-based neighbor learning). The distance can, in general, be any metric measure. Neighbors-based methods are known as non-generalizing machine learning methods.

Despite its simplicity, nearest neighbors has been successful in a large number of classification and regression problems, including handwritten digits or satellite image scenes. Being a non-parametric method, it is often successful in classification situations where the decision boundary is very irregular. Nearest neighbor search (NNS), as a form of proximity search, is the optimization problem of finding the point in a given set that is closest (or most similar) to a given point. Closeness is typically expressed in terms of a dissimilarity function: the less similar the objects, the larger the function values.

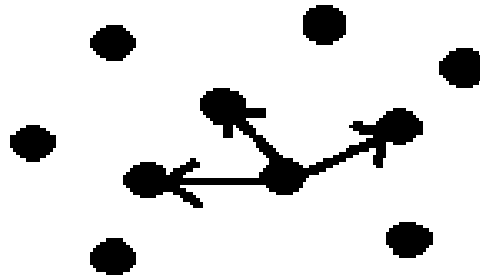


Figure 3.4.1. illustration of NN search

The steps to search are as follows:

1. Calculate the distances from the query point to every other point.
2. Sort those points by distance.
3. Return the first K items.

Given a query point P and an object O enclosed in its MBR, the two metrics are used for NN search namely, minimum distance (mindist) of the object O from P and minimum distance of the maximum possible distances (minmaxdist) from P to a face of MBR containing O. i.e., (mindist) and (minmaxdist) offer a lower and an upper bound on the actual distance of O from P. These bounds are used by the nearest neighbor search algorithm to order the objects. For

every newly visited non-leaf node, the algorithm computes the metric bounds for all its MBR and sorts them.

2.4.2 Search for the objects in the given area

A specific area from the geo-spatial image is given as input and all the objects that are present in the given area are searched.

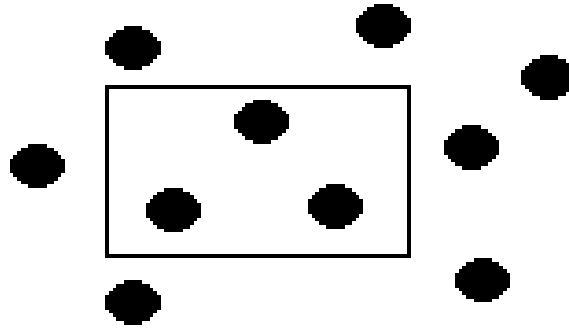


Figure 3.4.2. Illustration of search for the objects in an area

Given a geo spatial image to the R-tree which creates indices for each object in the image. Further the area in which the objects have to be searched is taken as input from the user and the coordinates are passed on to the searching algorithm. Eventually all the objects that are present in the specific area are obtained.

3 CONCLUSION

The conventional data structures for handling the numbers, strings and characters are not applicable for dealing with the objects or images. The storing and retrieval of objects requires an efficient and complex data structure which is R-Tree. In this work, an approach for searching the objects in a given area and nearest neighbor search is implemented.

REFERENCE

- [1] Antonin Guttman "R-Trees. A dynamic index structure for spatial searching" University of California Berkeley.
- [2] Tukaram Gawali and R.B Wagh," Smooth Query Processing In Spatial Data", IEEE 4th ICCNT-2013 july 4-6.
- [3] Simran Bijral, Debajyoti Mukhopadhyay "Efficient Fuzzy Search Engine With B-Tree Search Mechanism" International Conference of Information Technology, 2014.
- [4] Stefen Werner, Marc Stlzner "Hybrid FPGA approach or a B-Tree in a semantic web database system".
- [5] Liang Wang, Songnian Yu, Feng Chen, "A New Solution of Node Splitting to the R Tree Algorithm", International Conference on Intelligent Control and Information Processing August 13-15, 2010 - Dalian, China.