

# R-Trees. A Dynamic Index Structure For Spatial Searching

Antomn Guttman

Abhishek Srivastava

Student ID: 861307778

January 31, 2017

CS 202, Winter 2017

---

## The problem:

The paper notes that conventional indexing methods are not appropriate for multi-dimensional geo-spatial databases, since storing spatial data is not well represented by point locations hence operations performed will not be efficient. Multiple variants of databases proposed had shortcomings on their storing and indexing methods.

## The contribution:

The authors propose an 'R-Tree' index structure that can represent intervals of multi-dimensional data, is dynamic and can also provides efficient Searching and Updating methods. The paper discusses implementation of different operations and the benchmarks of indexing methods for different entries per node and efficiency of insertion, search and deletion operations for different node splitting methods.

## The method:

The R-Tree have two kind of nodes: Leaf nodes, and Non-leaf nodes. Leaf nodes contains pointer to data-objects or disk pages. Leaf nodes in an R-tree contains entries of the form  $(I, \text{tuple-identifier})$ . Non-leaf nodes contain entries of the form  $(I, \text{child-pointer})$ .  $I$  is an  $n$ -dimensional bounding box of spatial object indexed.

R-tree satisfies following properties:

- Unless it is root each leaf/non-leaf node should contain between  $M/2$  and  $M$  index records/children's.  $M$  is maximum number of entries in a node. The root has at least two children.
- In leaf node each index record is smallest rectangle that contains the  $n$ -dimensional data object indicated by the tuple.
- In non-leaf node each entry is smallest rectangle that spatially contains rectangles in child node.

Search operation to find an index entry is done is the following way : From the root search the subtrees to find entry which overlaps if it is non-leaf search in its children node. If node is leaf check all the entries to determine qualifying record. Inserting a node is done by searching the leaf where data can be inserted and splitting the node if it exceeds max number of entries possible and propagate it upwards. Deletion is similar by finding the leaf node and deleting it and merging the nodes where necessary and move upwards.

The main challenge in operations is node-splitting which can be done exhaustively but it is very high cost. Alternate methods can be used which are Quadratic and Linear cost algorithms they provide a sub-optimal solution but in permissible time. **Comments:**

The paper presents a novel indexing structure that is shown to be efficient for multi-dimensional spatial data objects. It provides several advantages over traditional indexing systems and is certainly capable of providing higher performance. However, it also gives rise to a few problems:

- Node splitting done by Quadratic-Cost and Linear-Cost Algorithm does not provide the smallest split possible.
- Spatial structures are represented in the rectangle boundaries form but it wont be able to uniquely present objects of different shapes who have same rectangle boundaries.