Proceedings of International Conference on Advances in Computer Technology and Management (ICACTM)
In Association with Novateur Publications IJRPET-ISSN No: 2454-7875
ISBN No. 978-81-921768-9- 5
February, 23rd and 24th, 2018

INTRODUCTION OF SPATIAL DATABASE SPATIAL DATA TYPES, SPATIAL INDEXING, GEOGRAPHIC INFORMATION SYSTEMS (GIS) & QUERY STRUCTURE

DR. P. G. KHOT
Professor Dept. of Statistics, RTM Nagpur University Nagpur, Maharashtra, India, pgkhot@gmail.com.

SANJAY SRIVAS
Research Associate, Dept. of Electronics & Information Technology, RTM Nagpur University Nagpur, Maharashtra, India, sanjaymsc@gmail.com, +91-9860195188.

ABSTRACT
This paper covers the overview of Spatial Database System which offers Spatial Data Structure along with the Spatial Datatypes. It is most commonly used in Geographic Information Systems (GIS) and other applications. Many applications operate on spatial data which include lines, points, regions and polygons etc. Spatial data are large in size and complex in structure and relationship hence spatial indexes are required to retrieve the desired result-set from the large dataset in optimum timeframe. The main emphasis on hierarchical data structures, including the number of indexing techniques like HHCode, R-tree, R+tree, Quadtree, Octree, UBtree etc. which are often used to improve query processing time in spatial database. Such techniques are also known as spatial indexing methods. Comparative study also covered in the paper for the R*-Tree and R-Tree indexing techniques, and results shown that the better performance of the queries using appropriate index.

KEYWORDS- Spatial Index, GIS, Quadtree, R-tree, R*tree,

I. INTRODUCTION

Spatial data denotes to all types of data objects or elements that are present in a geographical space or horizon. It enables the tracing of any locating, individuals or devices anywhere in the earth. Spatial data is also known as geospatial data, spatial information or geographic information. Spatial data is used in GIS and other location or Positioning Services. Spatial data consists of points, lines, polygons and other geographic and geometric data primitives, which can be mapped by location, stored with an object as metadata or used by a communication system to detect user position. Spatial data can be classified as scalar or vector data. Each one provides different information related to geographical or spatial positions.

The data that contains the particulars of any location {latitude and longitude, or height and depth} objects is the spatial data. When the object is rendered, this spatial data is used to project the locations of the objects on a 2-dimensional piece of paper. A GIS is frequently used to store, retrieve, and render geographic spatial data. Other types of spatial data which can be stored using the Spatial Data Option besides GIS data. It includes data from Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) systems. Instead of operating on objects on a geographic scale, CAD/CAM systems work on a smaller scale such as for an auto-mobile machine or much smaller scale as for printed circuit boards.

A. Point Data

Point is basically a location of any particular place which can be identify with the name or landmark. Generally point data is used to denote unconnected locations. Points have no dimensions; hence it can neither be used to measure length nor the area of the location. Points have various examples like: hotel, School and training center etc. The example below shows the location of ATM and Bank Branch. Point features are also used to represent abstract points**.** For example, point locations could represent city locations or name of any places as mentioned in the picture (Fig 1) below:

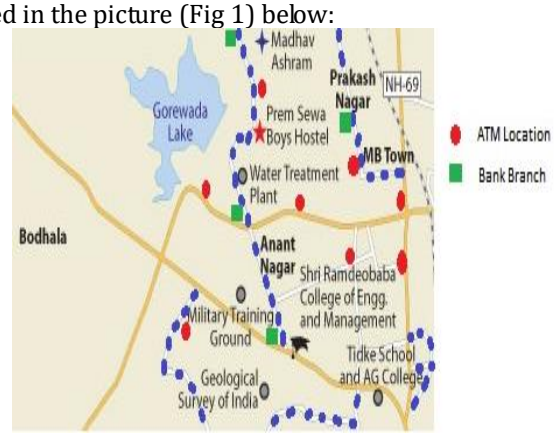


Fig 1. GIS Point data showing Location of ATM & Bank branch

B. Line Data

Line segment is a distance between two distinct points. It has beginning point and ending point. Line features have one dimension i.e. length. Line data is used to represent collinear points; some of the common examples are national highways, rivers and canals etc. Line types (solid lines versus dashed lines) and combinations can be

representing using colors, line type and line thicknesses **[1, 2]**. In the below example roads are distinguished from the internal roads designating roads as a dashed blue line and the national highways as solid yellow line.

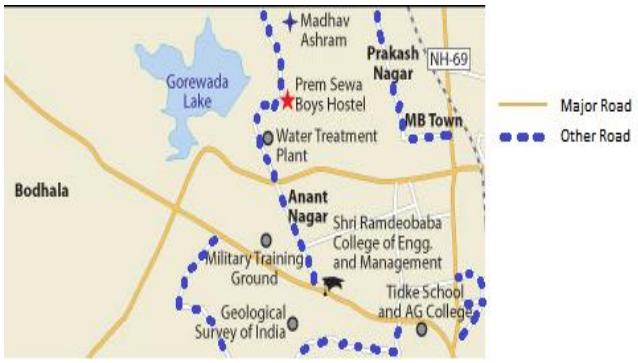


Fig 2. Internal roads are shown as dashed blue lines and National Highways major roads as solid yellow lines.

The data of both the line and point entities represent polygon data on a much smaller scale. It helps reduce confusion by simplifying data positions. As the functions approach, the location of a school point is more realistically represented by a series of building footprints that show the physical location of the campus. The line features of a line file of a road represent only the physical location of the road. If a greater degree of spatial resolution is required, a file of the width of the road curve will be used to show the width of the road and any characteristics such as median and rights of way (or sidewalks).

C. Polygon Data

Polygons are used to represent areas such as the border of a city (on a large-scale map), the lake or the forest. The characteristics of the polygon are two-dimensional and, therefore, can be used to measure the area and the perimeter of a geographical feature. The characteristics of the polygon are most commonly distinguished using a thematic mapping symbology (color schemes), models or, in the case of numerical gradation, a color gradation scheme that could be used.

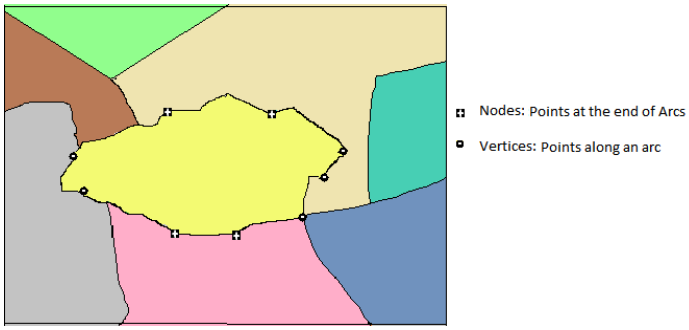


Fig 3. In polygon based dataset, different region areas are shown in color symbology

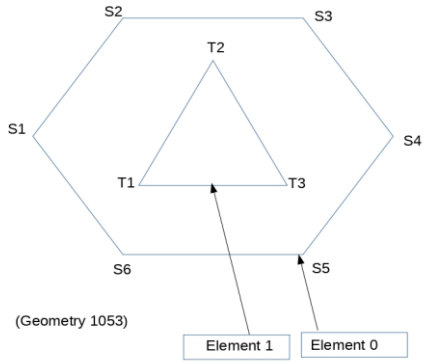


Fig 4. Complex Polygon

II. Spatial Indexing in Hierarchical database

Spatial indices are used by spatial databases (databases which store information related to objects in space) to optimize spatial queries **[3]**. Conventional index types do not efficiently handle spatial queries such as how distant two points differ, or whether points fall within a spatial area of interest. Some of the Common spatial index methods include:

a) HHCode (Helical Hyperspatial Code)

HHCODEs represent a regular binary decomposition of the object space and are used to define an exclusive and exhaustive cover of every element stored in a Spatial Data Option layer. Such HHCODEs are sometimes referred to as "tiles". Spatial Data Option can use either fixed or variable-sized tiles to cover geometry **[4]**.

With the HHCode being an open-source data format, several spatial data and software companies have adopted it in various products targeted at very large corporate data users, namely Helical Systems Inc. and CubeWerx.

b) Grid (Spatial Index)

In the context of a spatial index, a grid is a regular tiling of a multiple or two-dimensional surface that divides it into a series of contiguous cells, to which it is possible to assign and use unique identifiers for spatial indexing purposes. A wide variety of such grids has been proposed or is currently in use, including grids of "square" or "rectangular" cells, triangular grids or meshes, hexagonal grids and grids based on diamond cells.
"Square" or "rectangular" grids are usually the simplest in use, i.e. to translate spatial information expressed in Cartesian coordinates (longitude and latitude) inside and outside the grid system. These grids may or may not be aligned with the latitude and longitude grid lines.

c) **Z-order (curve)**

The Z-order can be used to efficiently construct a quadtree for a set of points **[5].** The main idea is to sort the input set according to the Z-order. Once classified, the points can be stored in a binary search tree and used directly, which is called linear quadtree or can also be used to build a quadtree based on the pointer.

d) **Quadtree**

A quadtree is a tree data structure in which each internal node has exactly four children. It is basically has two-dimensional analog of the most frequently used cotrees to divide a two-dimensional space, subdividing it recursively into four quadrants or regions **[6]**. The data associated with a child cell varies depending on the application, but the child cell represents an "interesting spatial information unit".

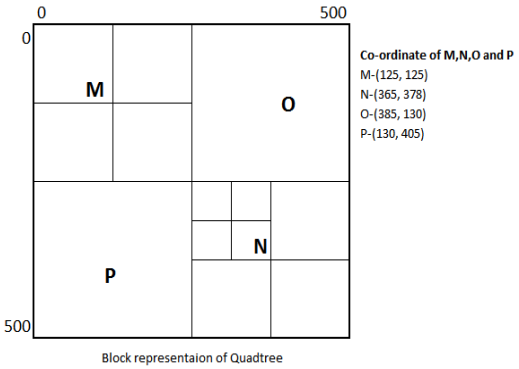
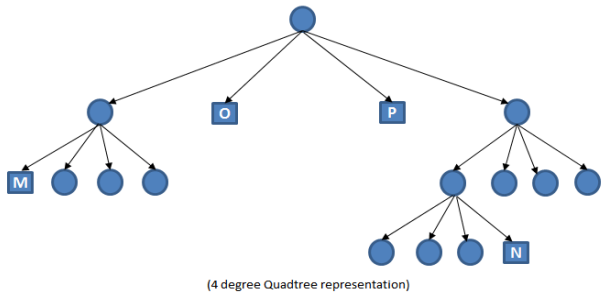


Fig 5. Quadtree image representation



The Quadtrees can be classified according to the type of data they represent, including areas, points, lines and curves etc. The Quadtrees can also be classified according to whether the shape of the tree is independent of the order in which the data is processed. Following are some common types of dials.

- Region Quadtree
- Point Quadtree
- Point-Region Quadtree
- Edge Quadtree
- Compressed Quadtree

e) **Octree**

The Octrees are a natural extension of the concept of quadtree. To describe any object that we wish to represent within a given space, the best approach seems to be an approximate of the position of the object, followed by subsequent refinements that increase the resolution of the objects' details. Such combinations of global and detailed specifications produce a hierarchical manifestation of the object's space.

An octree is a tree data structure in which each internal node has exactly eight children **[7]**. Octrees are often used to divide three-dimensional spaces by recursively dividing them into eight octants. The octrees are the three-dimensional analog of quadtrees. The Octrees are often used in the graphics and in the 3D game engine.

f) **UB-tree**

The UB tree is used to efficiently store and retrieve the multidimensional data proposed by Rudolph Bayer and Volker Mark **[8]**. Basically it is a B + tree (information only in the leaves) with the records memorized according to the Z-order, also called Morton's order. The Z-order is simply calculated by the bit-level interlacing of the keys.

Insertion, deletion, and point query are done as with ordinary B+ trees. To perform range searches in multidimensional point data, however, an algorithm must be provided for calculating, from a point encountered in the data base, the next Z-value which is in the multidimensional search range.

g) **R-tree**

Spatial data is two dimensional, so it is not possible to use B-tree index for spatial data. R-tree is the indexing methods which organize data in a tree shaped structure with grouped objects at nodes. Using R-tree index, search would be more efficient and less time taking because it eliminates the objects which are outside the area of interest and perform the search only on selected are of interest.

h) **R+ tree**

The R+ tree is a tree data structure, a variant of the R tree, used to index spatial information. The R+ tree is a method to search for data using a location, often (P, Q), and often for positions on the surface of the Earth. Searching in a number is a solved problem; observing two or more and asking for close positions in the P and Q directions, requires more advanced algorithms **[9]**.

i) **R* tree**

In data processing, the R* trees are a variant of the R trees used to index spatial information. The R* trees have a slightly higher construction cost than standard R-trees, as the data may need to be reinserted; but the resulting tree will generally have a better query performance. Like the standard R-tree, it can store point and spatial data. It was

proposed by Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider and Bernhard Seeger in 1990 **[10]**.

j) **Hilbert R tree**

Hilbert R-tree, an R-tree variant, is an index for multidimensional objects such as lines, regions, 3D objects or parametric objects based on high-dimensional features. It can be considered an extension of the B + tree for multidimensional objects.

The performance of the R-trees depends on the quality of the algorithm that groups the data rectangles into a node. The Hilbert R-trees use space-filling curves, and in particular the Hilbert curve, to impose a linear order on the data rectangles.

k) **X tree**

X-tree (Extended *node tree*) **[11]** is an index tree structure based on the R-tree used for storing data in many dimensions. It appeared in 1996 and differs from R-trees (1984), R+-trees (1987) and R*-trees (1990) because it emphasizes prevention of overlap in the bounding boxes, which increasingly becomes a problem in high dimensions. In cases where nodes cannot be split without preventing overlap, the node split will be deferred, resulting in super-nodes. In extreme cases, the tree will linearize, which defends against worst-case behaviors observed in some other data structures.

l) **kd tree**

K-d tree (abbreviation of the k-dimensional tree) is a space-partitioning data structure to organize the points in a k-dimensional space. K-d trees are a data structure useful for different applications, such as searches that involve a multidimensional search key (for example, search for intervals and searches for a nearer neighbor). The k-d trees are a special case of binary partition trees.

m) **m tree**

M-tree index can be utilized for the effective determination of similar queries on complex objects as compared using an arbitrary metric.

n) **Binary Space Partitioning tree (BSP-Tree):**

Binary space partition (BSP) is a method of recursively subdividing a space into convex sets using hyperplanes. This subdivision gives rise to a representation of objects in space by means of a data structure of the tree known as the BSP tree **[12]**.

The partition of the binary space has been developed in the context of 3D graphics **[13, 14]** where the structure of a BSP tree allows spatial information about objects in a scene that is useful in representation, such as its order from the front to the back compared to a spectator in a given place, to access quickly. Other applications include the execution of geometric operations with shapes (solid structural geometry) in CAD, 3D video games, ray tracing and other computer applications that involve the management of complex spatial scenes.

III. QUERY MODEL FOR SPATIAL DATA

Spatial and Graph use a two-level query model to resolve spatial queries and spatial junctions. The term is used to indicate that two different operations are performed to resolve queries. The output of the two combined operations produces the exact result set. The two operations are known as primary and secondary filter operations.

The primary filter allows a quick selection of candidate records to be passed to the secondary filter. The primary filter compares geometric approximations to reduce the complexity of the calculation and is considered a lower cost filter. Because the primary filter compares geometric approximations, it returns a superset of the exact result set. The secondary filter applies exact calculations to the geometries resulting from the primary filter. The secondary filter produces a precise response to a spatial query. The secondary filter operation is computationally expensive, but only applies to the primary filter results, not to the entire data set.

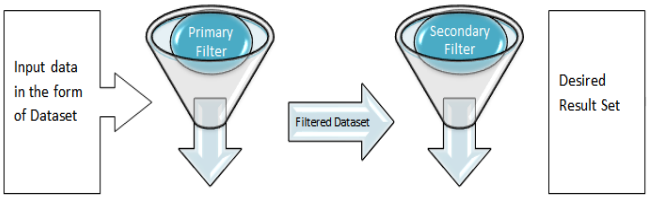


Fig 6. Query Model

IV. Database Structures of Spatial database

The Spatial data is stored in database in the form of layered structure. Mainly it uses four tables which are collectively referred as a layer. A layer is a heterogeneous collection of geometries having the same attribute set. A sample SQL script is mentioned as below to describe the structure **[15]**.

<Layername>: SDOLAYER table or View

SDO_ORDCNT
<number>

<Layername>: SDODIM table or View

SDO_DIMNO	SDO_LB	SDO_UB	SDO_TOLERANCE	SDO_DIMNAME
<number>	<number>	<number>	<number>	<varchar>

<Layername>: SDOGEOM table or View

SDO_GNO	SDO_SEQ	SDO_X1	SDO_Y1	...	SDO_Xn	SDO_Yn
<number>	<number>	<number>	<number>	...	<number>	<number>

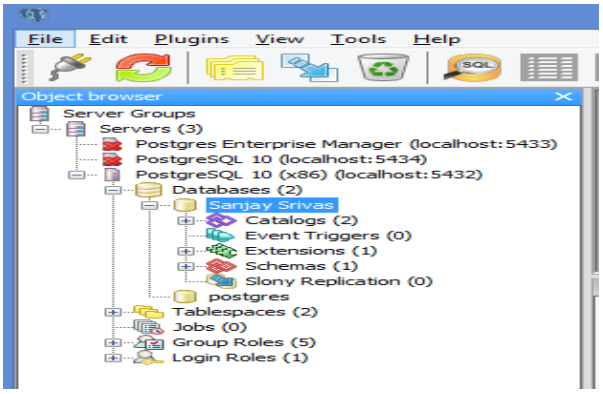
<Layername>: SDOINDEX table

SDO_GNO	SDO_INDEX	SDO_MAXCODE
<number>	<number>	<number>

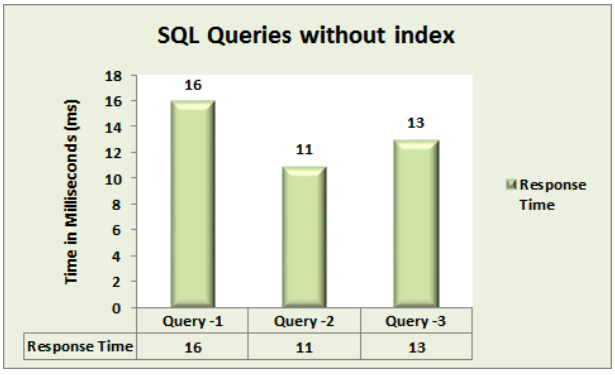
Fig 7. An Example of schema of a Spatial Data Option layer.

V. EXPERIMENT RESULT & DISCUSSION

We installed the PostgreSQL database in our system and used the geographic sample data to analysis the patterns while using the index for querying from the database. PostgreSQL is an Object-Relational Database Management System (ORDBMS). It is an open source which supports the SQL standards and has a lot of advanced features like, foreign key, triggers, transactional integrity, updatable views, complex queries etc. These days many of the research and production applications are implementing in the Postgres SQL.



After creating database in PostgreSQL, sample data populated into the tables. After polluting the data into the tables. We tried to run 3 queries (complex queries using SQL joins) without using any index and captured the execution time of the queries without and with indexes for the experiment purpose.



Now it is the time to create the index of all the tables using R-Tree and R*-Tree indexing structure. Then again ran the same queries using both indexes separately.

Creating Index:
CREATE INDEX ("index_name") on ("table_name") Using rtree ("column_name");

Table 1 shows the time taken by the index namely, R*-Tree and R-Tree to implement in the dataset. The use of indexes clearly increases the processing speeds of the queries.

Index	Table-1	Table-2	Table-3
R*-Tree	3.632	2.035	3.396
R-Tree	1.662	1.06	0.447

Table 1. Response time table

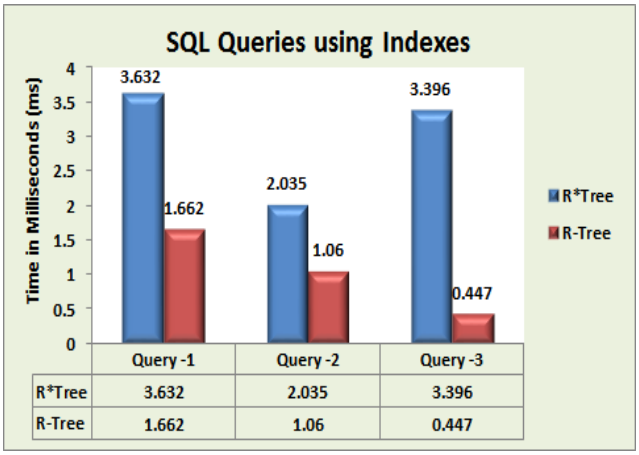


Fig 8. Histogram for comparison of response time using indexes.

We compared the response time of the queries in different tables and it is clearly shown as per above histogram that, R-Tree is the least time consuming indexing structure. However R*-Tree is also an option when there is no arithmetic operations in the queries.

VI. CONCLUSION

Due to high requirement of geographical data in various applications and use of layer data in hierarchical data structure, sometime it is very challenging to retrieve the desired result set within timeline. This paper presented several spatial indexing techniques for hierarchical data structure which can be used for faster data retrieval. Also comparison study shows that some indexing techniques are significantly increasing the query processing speed which result the faster processing from the huge dataset. Mainly the appropriate index selection is required as per the data which is used in the database to enable faster query processing.

VII. ACKNOWLEDGEMENT

I would like to sincerely thank to Dr. Anita Soni, Associate Professor, Department of Computer Application, TIT, Bhopal for constant support & guidance for this study. Her constant support and supervision always encourage me to achieve new dimension in research field.

VIII. REFERENECES

1) Freeman, H.1974. Computer processing of line{drawing images.ACM Computing Surveys,6, 1 (Mar.), 57-97.
2) Hoel, E. G. and Samet, H.1992. A qualitative comparison study of data structures for large line segment databases. InProceedings of the SIGMOD Conference, (San Diego, June), pp. 205-214.
3) Jagadish, H. V.1990. On indexing line segments. InMcLeod, D., Sacks-Davis, R., and Schek, H., Eds.,Proceedings of the Sixteenth International Conference on Very Large Databases (VLDB), (Brisbane, Australia).
4) Varma, H., T. Milne, and G. Kierstead, The Origins of HH Codes, Internal report of Bedford Institute of Oceanography, Canada (1996).
5) Varma, H. et al. A Data Structure for Spatio-Temporal Databases (1990). International Hydrographic Review, Monaco, LXVII(1).
6) Bern, M.; Eppstein, D.; Teng, S.-H. (1999), "Parallel construction of quadtrees and quality triangulations", Int. J. Comp. Geom. & Appl., **9** (6): 517–532.
7) Gargantini, I. (1982), "An effective way to represent quadtrees", Communications of the ACM, 25 (12): 905–910.
8) H. NOBORIO, S. FUKUDA, S. ARIMOTO: Construction of the octree Approximating Three Dimensional Objects by Using Multiple Views, University of Toyonaka, Osaka, Japan, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10 no. 6 , November 1988, pp. 769-781.
9) Rudolfh Bayer, "The universal B-Tree for multidimensional Indexing: General Concepts", World-Wide Computing and its Applications '97 (WWCA '97), 1997.
10) A. Guttman, R-trees: a dynamic index structure for spatial searching, Proceedings of the SIGMOD Conference, Boston, June 1984, 47–57.
11) Beckmann, N.; Kriegel, H. P.; Schneider, R.; Seeger, B. (1990). "The R*-tree: an efficient and robust access method for points and rectangles". Proceedings of the 1990 ACM SIGMOD international conference on Management of data - SIGMOD '90 (PDF). p. 322. ISBN 0897913655.
12) Berchtold, Stefan; Keim, Daniel A.; Kriegel, Hans-Peter (1996). "The X-tree: An Index Structure for High-Dimensional Data". Proceedings of the 22nd VLDB Conference. Mumbai, India: 28–39.
13) Fei-Ching Kuo, Shuang Liu, T. Y. Chen, Testing a binary space partitioning algorithm with metamorphic testing, Pages: 1482-1489.
14) Schumacker, Robert A.; Brand, Brigitta; Gilliland, Maurice G.; Sharp, Werner H (1969). Study for Applying Computer-Generated Images to Visual Simulation (Report). U.S. Air Force Human Resources Laboratory, Page. 142.
15) Fuchs, Henry; Kedem, Zvi. M; Naylor, Bruce F. (1980). "On Visible Surface Generation by A Priori Tree Structures" (PDF). SIGGRAPH '80 Proceedings of the 7th annual conference on Computer graphics and interactive techniques. ACM, New York. pp. 124–133.
16) Brinkhoff, Kriegel, Schneider, R. (1993a), Comparison of Approximations of Complex objects used for Query processing in Spatial Database System, in the proceeding of 9th International conference on data Engineering.