IJournals: International Journal of Software & Hardware Research in Engineering
ISSN-2347-4890
Volume 4 Issue 3 March, 2016

Comparative Study and Analysis of Spatial Access Method using Fractal and L-system Approach: A Critical Review

Author: Rojali Mjahi[1]; Dr.Sasmita Mishra[2]

Department of Computer Science, Engg& Applications, Indira Gandhi Institute of Technology, Sarang, Dhenkanal, Odisha (India)-759146
Email: rojalimajhi@gmail.com,sasmita.mishra.csea@gmail.com

ABSTRACT

This paper attempts to give a broad outlook to the potential researchers in the field of spatial access methods. Many journal papers have been critically reviewed to establish the true research direction in this field. This paper is attempted to give a comparative analysis of existing multidimensional access methods based upon fractals and L-System.

Keywords: **B–Tree, R–Tree, Hilbert curve, Query type, Data type, Comparative Study, L–System.**

1. INTRODUCTION

Spatial data also known as geospatial data or geographic information, it is the data or information that identifies the geographic location of features and boundaries on Earth, such as natural or constructed features, oceans, and more. There are two characteristics of spatial datasets that they are frequently large and that the data are quite often distributed in an irregular manner. Spatial data is usually stored as coordinates and topology. Spatial access methods are used to support efficient selection of objects based on spatial properties. There are two characteristics of spatial datasets that they are frequently large and that the data are quite often distributed in an irregular manner.

A spatial access method regards both spatial indexing and clustering techniques. Without use a spatial index, every object in the database has to be checked to see whether it meets the spatial selection criterion, a full table scan in a relational database. As spatial datasets are usually very large so full table scan is unacceptable [1]. Without looking at every object a spatial index is necessary, in order to find the required object efficiently. Most spatial datasets are large; example Geographic maps typically occupy several gigabyte of storage so that they cannot reside in the main memory of the computer and most be stored in secondary memory [2]. Clustering is required to group those objects which are often requested together [1].

Every important class of geometric operators that needs spatial support at the physical level is the class of spatial search operators. Value of certain alphanumeric attributes and the spatial location of a data object are required for retrieval and update of spatial data [3].A retrieval query on spatial database requires a geometric operation such as a point and region query. Both operations in the database require fast access to those objects that occupy a given location in space [3]. Multidimensional data with explicit knowledge about object, their extent and their position in space is contained by spatial database [1]. The main problem for using multidimensional access method is that there exists no total ordering among spatial objects that preserve spatial proximity. This makes difficult to design of efficient access methods in the spatial domain than in traditional databases. In B tree approach sorting of the data forms the basis of efficient searching as in traditional database system [1].

2. DIFFERENT TYPES OF ACCESS METHODS

Spatial data is 2-dimensional or 3-dimensional (or even higher dimensioned) but computer memory is 1-dimensional, and must be organized somehow in memory [1].Followings are the different access methods:

2.1 B-Tree:

B- Tree is used for 1- dimensional access method. One-dimensional access methods act an important foundation for almost all multidimensional access

IJournals: International Journal of Software & Hardware Research in Engineering
ISSN-2347-4890
Volume 4 Issue 3 March, 2016

methods. A collection of index element which are pairs (x,α) of fixed size physically data items, namely a key x and some associated information α by an index. The key identifies a unique element in the index and the associated information is typically a pointer to a record and collection of records in a random access file [2].

B-tree contains variable number of child node with predefined range. Because of fixed range each time node is inserted or deleted internal node may join and split [4].

Let k a natural number, h $\geq$ 0 be an integer. It is a directed tree with in the class τ(k,h)of B trees if T is either empty (h=0) or has the following properties:

(i) Every path from the root to any leaf has the same length h, also called the height of T, i.e. h= number of nodes in path.

(ii) Every node accepts the root and the leaf has at last k+1 sons. Either the root is a leaf or has at last two sons.

(iii) Every node has at most 2k+1 sons.

In B tree best case height is log (mn) and worst case height is log (m/2n).

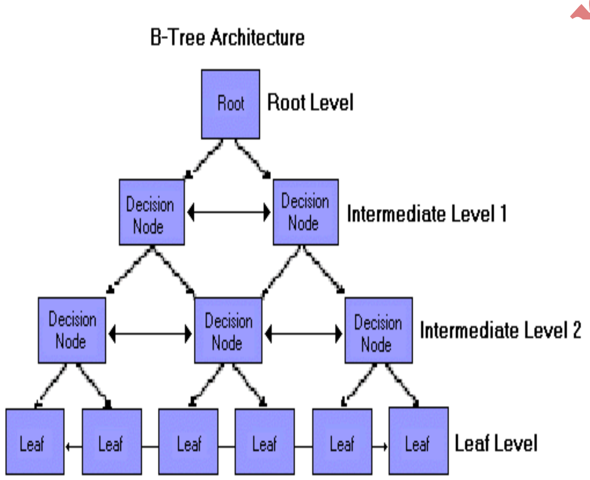


Figure1: B-tree

B tree do not work well with spatial data because search space is multidimensional. Spatial data covers space in multidimensional not presented properly by one dimensional index structure [4].

2.2 R-Tree

Antonin Guttman developed R tree in 1982. It is dynamic index structure use for the spatial searching. Several dimension uses to represent data object. It is height balance tree just like B tree. Operation like insertion and deletion can be intermixed with searching due to index structure is dynamic [4].

Let M is the maximum number of entries to R tree in one node and minimum number of entries in a

node is m $\leq$ M/2. R-tree has the following properties:-

(i) Between m and M each leaf node (unless it is root) has index record.

(ii) Each index records contains (I, tuple - identifier) in a leaf node where I is smallest rectangle represented by the indicated tuple and contains the n-dimensional data object.

(iii) Each non leaf node has children between m and M.

(iv) Non leaf node (I, tuple-identifiers) each entry where I contain the rectangle in the child node is the smallest rectangle.

(v) The root node having at last two children.

(vi) All leaves appear on the same level.

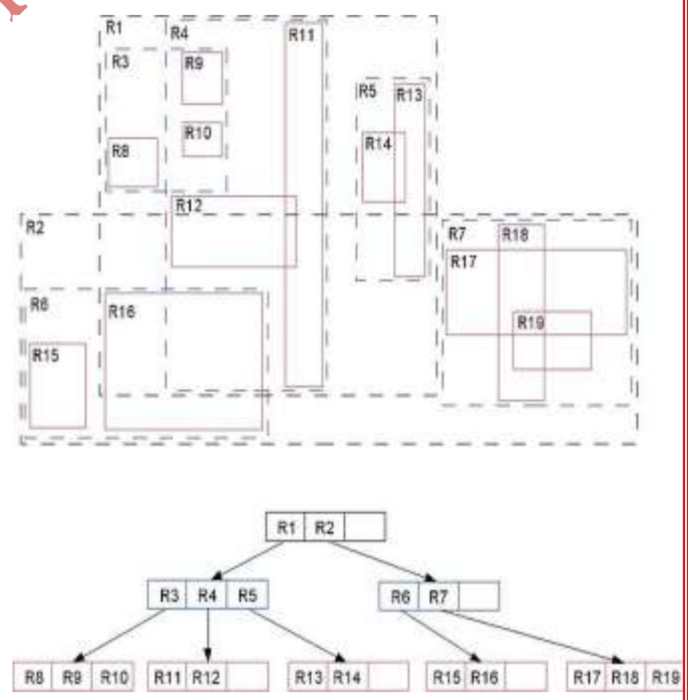


Figure2: R-Tree

An extension of the B tree is the R tree for multidimensional objects. The main innovation in the R tree is those fathers nodes are allow overlapping during split. So the R tree can guarantee at last 50% space utilization and remain balance [5]. If a node overflows, some of its children are carefully chosen; they are deleted and re-inserted, usually resulting in an R tree with better structure.

3. VARIANTS OF B TREE AND R TREE

3.1 Variants of B tree

B+ tree is similar is B tree the only difference is all records are stored at leaf level and keys stored in non-leaf nodes. B+ tree, B* tree and many other improved variants of B-tree is also proposed for particular data types. B-tree is effective for the point query but not for range query and multi-dimensional data. Index node and leaf nodes may have different formats or even different size. Leaf nodes are usually linked to gather left-to-right. B+-tree understands the implications of having an independent index and sequence set. In B+-tree searching proceeds from the root of the index to a leaf. All keys reside in the leaves, it does not matter what values are encountered as the search processes as long as path leads to the correct leaf [6].

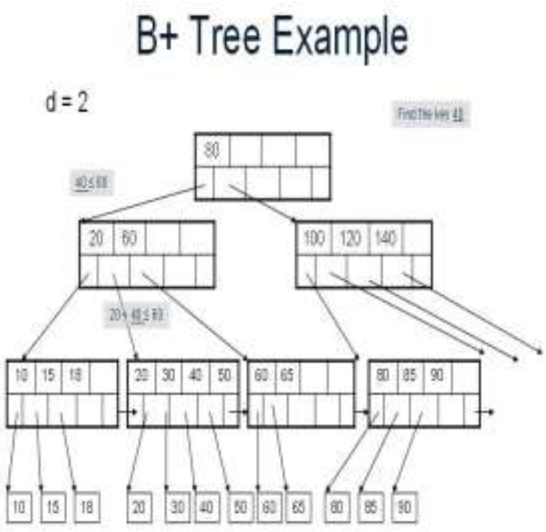


Figure3: B+ Tree

In the time of deletion the ability to leave non-key values in the index part as separators simplifies processing. It removal is simple because the key to delete must always reside in a leaf. Insertion and searching operation in a B+ tree are processed almost identically to insert and searching operation in a B tree [6].

B*-tree has frequently been applied to another, very popular variation of B-tree. A B*-tree to be a B tree in which each node is at last 2/3 full (instead of just ½ full). In B*-tree insertion a local redistribution scheme is applied to delay splitting until 2 sibling nodes are full. Then the 2 nodes are divided into 3 nodes, each2/3 full. This redistribution scheme guarantees that storage utilization is at last 66%. Science the height of resulting tree is smaller, that increasing storage utilization has the side effect of speeding up the search [6].

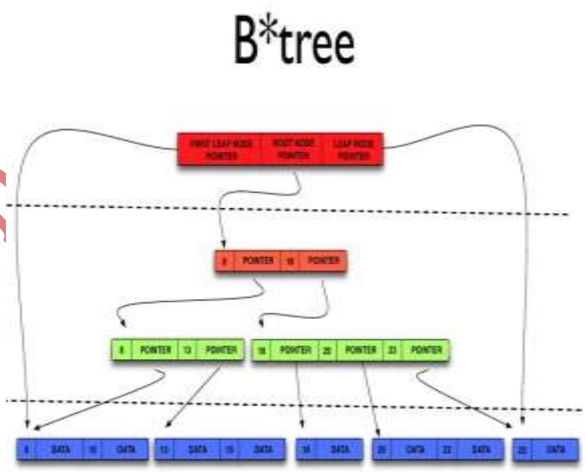


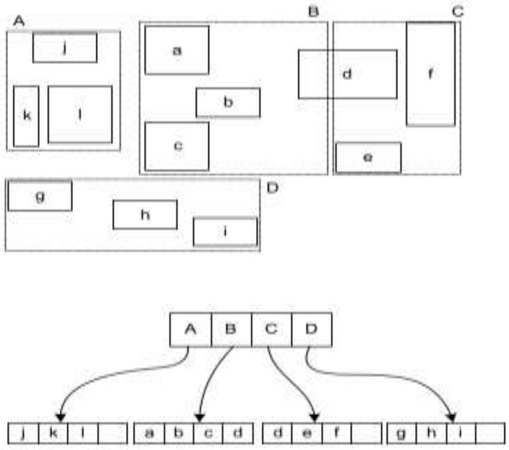
Figure 4: B*Tree

3.2. Variants of R tree:-

Main disadvantage of R tree is space wastage. So variants of R tree were proposed R+-tree, R*-tree, priority R tree, Hilbert tree etc. R+-tree is different from R tree because of the following point:-

1. Nodes are not guaranteed to fill at last half.
2. Each entries of internal node do not overlap.
3. More than one leaf node may use to store object id.

R+ tree can be larger than R tree because data are duplicated over many leaf node structures.



. An R+-tree example.

Figure 5: R* Tree

Another variants of R tree is R* tree, it tested parameter area, margin and overlap in different combination. To split of node R* tree first sort lower value of the rectangles then two group are determined. R* tree is very robust in compare to other data distribution. Three moral value and different approaches using them in different combination are tested; first margin-value, second area-value, third overlap-value. It reduced the split operation but it's one of costly operation is reinsertion. Storage utilization is higher than variants of R tree. R* tree is build by repeated insertion in index structures. There is little overlap in this tree "figure 6", resulting in good query performance. Red and blue MBRs are index pages, green MBRs are leaf nodes.

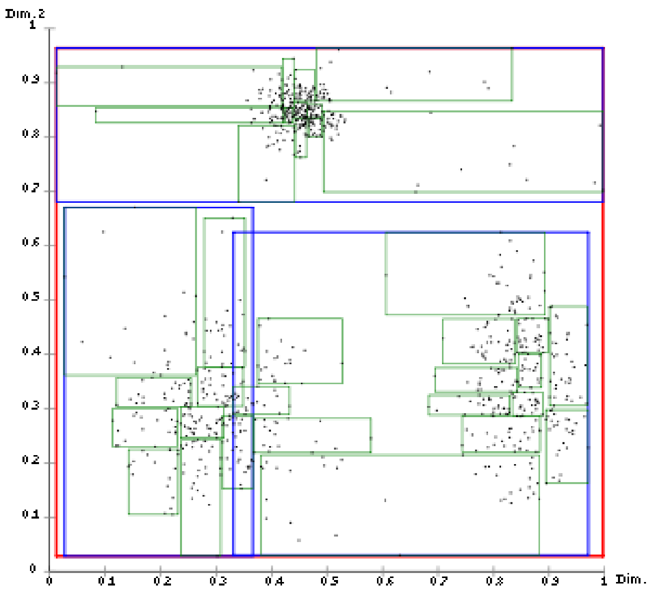


Figure 6: R*Tree

BR- tree (Bloom filter base R-tree) is basically R tree structure for supporting dynamic index. Within it each node maintains range index to indicate attributes of existing items. It store item and range of it together with the help of range query and cover query support. Using several independent hash function an index can represent a set of item as a bit array.

BR tree is a lode balance tree. R tree node and Bloom filter together known as BR tree node. It is not ordinary because BR tree maintains advantage of Bloom filter and R trees both. After point query result is positive it mixes the queries like bound query and range query. BR tree keep uniformity between queried data and the attribute bound in an integrated structure so that fast point query and accurate bound query possible [4].

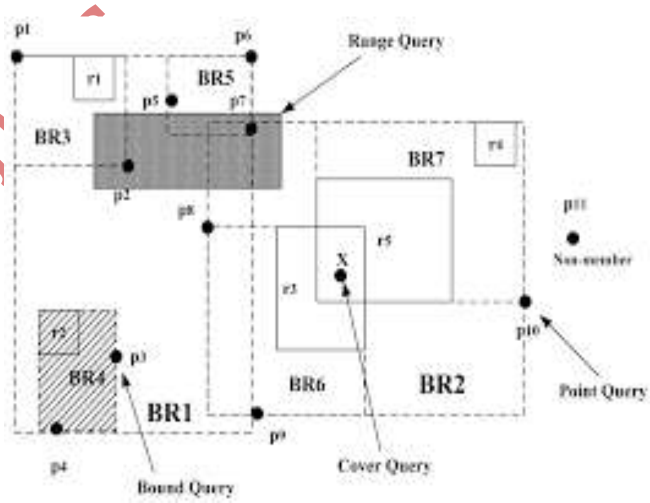


Figure 7: BR Tree

4. COMPARISON BETWEEN ACCESS METHODS

4.1 Query type

There are 4 types of query i.s Point query, range query, bound query and cover query.

4.2 Data type

There are two types of data linear and multidimensional. 3-D objects, curves, rectangles are represented by multidimensional data. Spatial data are also part of multidimensional data.

4.3 Complexity

Table1. Comparison between access methods

Index Structure	Query Type	Data Type	Complexity	Application
B- tree	Point query	Linear data	O(log n)	Some Linux file systems, Microsoft's NTFS etc.
B+- tree	Point query	Linear data	O(log n)	Database management system IBM DB2, oracle 8 etc.
B*- tree	Point query	Linear data	O(log n)use more space than B+ tree	HSF system
R-tree	Range query	Multidimensional data	Not utilize space more efficiently and not have worst time complexity	Navigation system in real world application etc.
R+- tree	Range query	Multidimensional data	Utilize space by non overlapping efficiently than R-tree	Multidimensional data object.
R*-tree	Point query, range query	Spatial data	Implementation cost is more than R tree	Application with data having points and rectangles.
BR- tree	Range query	Multidimensional data	O(n)	High dimensional data

Every data structure has complexity in term of space and time.

4.4 Application

Different methods are used for the different application.

5. HILBERT CURVE

Space filling curve transforms a 2- dimensional problem into a 1- dimensional one, so it can be used in combination with a well-known data structure for 1- dimensional storage and retrieval, such as B tree [1]. Space filling curves have been used in computer science for many applications due to their good clustering properties [7]. In space filling a curve visits all the points in k dimensional grid exactly once and never crosses itself. Hilbert curve is example of space filling curve. Propose for using space filling curve (or fractals), and specifically the Hilbert curve to impose a linear ordering on the data rectangle [5].

A 2*2 grid Hilbert curve is shown in Figure 8. Each vertex of the basic curve is replaced by the curve of order i-1, to derive a curve of order I, which may be appropriately reflected. Figure 8 also shows the Hilbert curve of order 2 and 3 [5]. In 4*4 grids the point (0, 0) on the Hilbert value of 0, while the point (1, 1) has a Hilbert value of 2.

That way Hilbert curve visits every point in a square grid with a size 2*2, 4*4,8*8,16*16 or any other power of 2 [5]. Hilbert curve can be generated by using L- system concept. To represent these curves (figure 8) in an easy way, apply the following terminology.

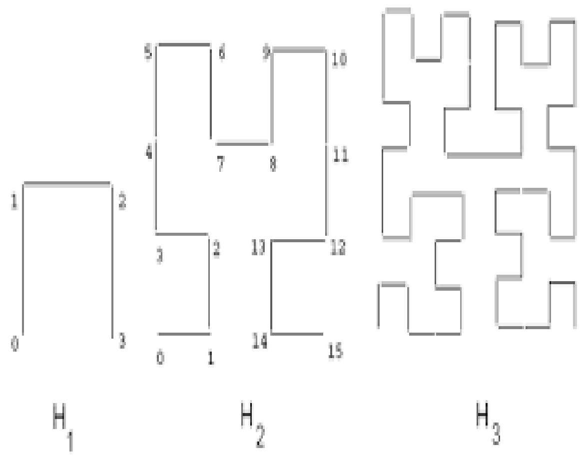


Figure 8. Hilbert curve

A horizontal line of one unit to the right is indicated by a 1, one unit horizontal to the left by -1 and the same for 2 in the vertical direction: 2 represents one unit up and -2 one unit down. So Hilbert curves are the representations 2, 1, -2 for first curve and 1, 2, -1, 2, 2, 1, -2, 1, 2, 1, -2, -2, -1, -2, 1 for second curve.

Before generating an L-system from Hilbert curve first introduce L- system.

5. L-SYSTEM

1. An L-system consists of V, an *alphabet* (of symbols)

$V^* = V\cup(V*V)\cup(V*V*V)\cup\ldots\ldots=\sum V^k$ for $k\geq1$.The set of all *strings* (words) of V (with *being the Kleene star) and *production rules*, a mapping $P:V\rightarrow V^*$.

2. Now p can be extended to $p:V^*\rightarrow V^*$, where the small latter p is use for extended mapping, by $p(x_1,x_2,x_3,\ldots\ldots)=$ $p(x_1)$, $p(x_2)$, $p(x_3)$….. where $x_i\in V(i\geq1)$ and the strings are concatenated.
3. $\{x\in V \mid P(x) = x\}$ are constant and $\{x \in V \mid P(x) \neq x\}$ are variables. For a special $s\in V^*$ all generation is $\{P^n(s) \mid n\geq 0\}$

Example. The 2-dimensional Hilbert curves are generated by the L-system with V= {0, 1, -1, 2, -2, 3, -3, 4, -4}, 0 being the start element and the production rules being

$$P=\begin{cases}0\rightarrow 0,3,4,-1\\ 1\rightarrow 2,3,\ 4,-1\\ 2\rightarrow 1,\ 3,4,-1\\ 3\rightarrow 4,2,1,-4\\ 4\rightarrow 3,2,1,-4\end{cases}\quad \text{and } P(-x)=-P(x)$$

In transforming a string like 0, 3, 4, –1, 4, 2, 1, –4, 3, 2, 1, –4, –2, –3, –4, 1 to a graph the following prescription is adopted: 0 makes the graph start in the origin, 3/–3 and 4/–4 are replaced by 1/–1 and 2/–2 respectively and a 1 means a horizontal line of one unit to the right.–1 one unit horizontal to the left and the same for 2 in the vertical direction: 2 equal one unit up and –2 equals one unit down. So this string gives the second (rotated) Hilbert curve as one can easily check. So the string which represents the 2nd Hilbert curve is 1, 2, –1, 2, 2, 1, –2, 1, 2, 1, –2, –2, –1, –2, and 1. The ultimate Hilbert string is not (yet) in the OEIS [10].

6. CONCLUSION

Some access methods have less space complexity; some have less time complexity and support different data types. Main objective of this paper is to compare various multidimensional access methods with their performance and make some observations using fractal and L-system.

7. REFERENCES

[1] P VAN OOSTEROM, "Spatial access methods" pp. 385–400, 1995.
[2] R.Bayer and E. McCreight, "Organization and maintenance of large order indices", Mathematical and Information Sciences laboratories, July 1970.
[3] VolkerGaede and Oliver Gunther , "Multidimensional Access Methods", By the ESPRIT working group.
[4] Parth Patel and Deepak Garg, "Comparison of advance tree data structures", International Journal of Computer Applications, Volume 41–No.2, March 2012.

[5] Ibrahim Kamel and Christos Faloutsos, ”Hilbert R-tree: An improved R-tree using fractals ”, Proceeding of the 20th VLDB Conference, 1994.

[6] Computer science Department, “The Ubiquitous B-tree” , Computing Surveys Vol 11, No 2, June 1979.

[7] IBM Research, “A note on Space filling visualizations and Space –Filling curve”.

[8] ArieBos, “Hilbert curve in 2- Dimensions generated by L- system”, vol. 33, no. 4, 2000.

[9] Sasmita Mishra, An Intuitive Method for Hilbert Curve Coding, International Journal of Computing and Corporate Research, Published in IJCCR, Volume 1 Issue 3, November 2011.

[10] S.N.Mishra,et al., L-System Fractals, MATHEMATICS IN SCIENCE AND ENGINEERING Volume 209 Series,Elsevier, 2007.

	Miss.RojaliMajhi is currently pursuing her MTech degree at Indira Gandhi Institute of Technology, Sarang, Dhenkanal, Odisha. She is doing her Mtechdisseratation under the able guidance of Prof.(Dr). Sasmita Mishra. Her research area is characteristic study of spatial data and L-System.
	Dr.(Mrs.) Sasmita Mishra is currently working as Associate Professor in the Department of Computer Science, Engg. & Applications at Indira Gandhi Institute of Technology, Sarang, Dhenkanal, Odisha, India. She has published more than 80 papers in International Journals and National Journals of repute. Her research area focuses on multidimensional database, spatial data analysis and fractal & its applications. She has more than 25 years of teaching and research experiences. She is guiding many PhD and MTech scholars for their research work.