Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.810 – 816

RESEARCH ARTICLE

Efficient Data Storage and Searching for Location Based Services using Quadtrees and H-ordering

Capt. Dr. S. Santhosh Baboo[1], V.Narmadha[2]

[1]Associate Professor, Department of Computer Science, D.G. Vaishnav College, Chennai, India

[2]Research Scholar, Research and Development Center, Bharathiar University, Coimbatore, Tamil Nadu, India

santhos2001@sify.com; vnarmadha@yahoo.co.in

Abstract—***Data is always stored in secondary storage Devices due to its volume. When it is stored in secondary storage devices, it is required that the access time is as minimal as possible. Most of the queries issued for providing location based services, result in accessing nearest point. This paper provides an algorithm to store the data efficiently so that accessing time is reduced for answering such queries using a space filling curve called Hilbert Curves and quadtrees.***

Keywords— space filling curve, data, storage, quadtrees, Hilbert curves

I. Introduction

Spatial objects representation and storage is a difficult task in Geographical Information system. It has been a research topic in GIS always. Database Management system can handle data very easily if object is one dimensional in nature. Object to be stored is more than one dimension; the method proposed in this paper can be used. The method is efficient in terms of storage and access time.

The main idea in the proposed method is to use two consecutive transformations, to map spatial objects into one dimensional points. Good distance-preserving mappings are essential for the performance nearest neighbor queries. The structures used for storing is quadtrees. We examine space filling curves, also known as "fractals" [10]. These curves, such as the Peano curve and the Hilbert curve, define a path that traverses the points in a N ×N square grid. These curves can be generalized for higher dimensionality spaces.

The paper is organized as follows: Section 2 gives a brief survey about Hilbert curve. Section 3 describes about quad trees. Section 4 describes the proposed approach distance preserving mappings. Section 5 presents the conclusions and future research.

II. Hilbert Curve

A space filling curve is a continuous, surjective mapping from R to R^d. It was not always clear that such a mapping would exist for d > 1. In 1878 Cantor exhibited a one to one map from the unit interval I=[0, 1] onto unit square S = [0, 1] X [0, 1] and thus proving that I and S have same cardinality.

Later in the late 19th century Peano[1] showed that it is possible to find a continuous map which is not one to one but onto for d = 2 and d = 3. An example given later by Lebesgue makes use of the standard Cantor set I. Any number t in has a expansion given by

t = $\sum_{i=0}^{\infty}$ ai/3i, where each ai takes one of the values 0, 1, or 2.

The Cantor set C is defined as

C= { $t: t = \sum_{i=0}^{\infty}$ ai/3i with ai = 0 or 2 }, C is known as fractal.

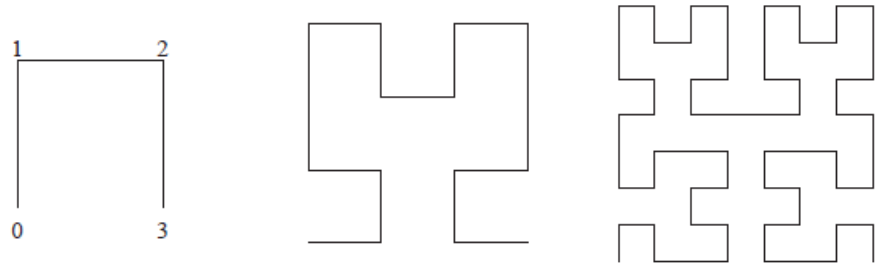
Another (geometric) way to arrive at *C* is the following: from *I*, first remove the open middle one-third interval G_i = (1/3,2/3) and call what remains as F_1. Thus F_1= [0,1/3] U [2/3, 1]. From each of the two intervals in F_1 , remove their open middle one-third intervals (1/9,2/9) and (7/9,8/9) and call what is left as F_2. From each of the four intervals in F_2 , remove their open middle one third intervals and call what remains as F_3 and so on. What finally remains is the Cantor set C = $\cap$ *Fn*. Lebesgue's construction can now be given as follows:

for t ϵ C with t = $\sum_{i=0}^{\infty}$ ai/3i,

Define

x(t) = $\sum_{i=1}^{\infty} \frac{b2i-1}{2i}$ and y(t) = $\sum_{i=1}^{\infty} \frac{b2i}{2i}$ where $b_i = a_i / 2$.

The popularity of space filling curve is due to the geometric construction given by the Germen mathematician David Hilbert. His basic idea was that if the unit interval should fill the whole of S, then $1/4^{th}$ of I will fill a corresponding subsquare of S of area ¼ with continuity in neighbouring squares. Next I and S can be replaced by an interval of length ¼ and subsquare area of ¼ respectively and the process can be repeated. Hence for each n>1, I and S are subdivided into 4^n closed intervals and 4^n closed subsquares. The first three stages are shown in Figure 1. At each stage centres of the subsquare are joined by consecutive straight lines in the shown in Figure1. This procedure defines a sequence of continuous functions from I to S. Since the length of the sides of the square tends to 0, the sequence converges to a limit function which is therefore continuous. This limit function is called Hilbert Curve[.



(a) (b) (c)

Since then, quite a number of space filling curves have appeared in the literature. During the early days space filling curves were primarily seen as a mathematical curiosity. Today however, space filling curves are applied in areas as diverse as load balancing for grid computing, colour space dimension reduction, small antenna design, I/O-efficient computations on massive matrices, and the creation of spatial data indexes. In this paper, we focus on the application of space filling curves to the creation of query-efficient spatial data indexes using Hilbert curve.

In 1890, Italian mathematician G. Peano present a family of curves which pass through all points in a space. Among the space filling curves such a Piano, Sierpenski, Moore and Hilbert curves, Hilbert curve has the property of preserving the locality. Let us make use of this property and use it for solving the problem.

III. Quadtrees

Quadtrees are a very straightforward spatial indexing technique. In a Quadtree, each node represents a bounding box covering some part of the space being indexed, with the root node covering the entire area. Each node is either a leaf node - in which case it contains one or more indexed points, and no children, or it is an internal node, in which case it has exactly four children, one for each quadrant obtained by dividing the area covered in half along both axes.

To query a quadtree, starting at the root, examine each child node, and check if it intersects the area being queried for. If it does, recurse into that child node. Whenever you encounter a leaf node, examine each entry to see if it intersects with the query area, and return it if it does.

A quadtree is a regular trie, since the values of the tree nodes do not depend on the data being inserted. A consequence of this is that we can uniquely number our nodes in a straightforward manner: Simply number each quadrant in binary (00 for the top left, 10 for the top right, and so forth), and the number for a node is the concatenation of the quadrant numbers for each of its ancestors, starting at the root. Using this system, the bottom right node in the sample image would be numbered 11 01.

If we define a maximum depth for our tree, then, we can calculate a point's node number without reference to the tree - simply normalize the node's coordinates to an appropriate integer range (for example, 32 bits each), and then interleave the bits from the x and y coordinates -each pair of bits specifies a quadrant in the hypothetical quadtree. This is well explained in Figure 2.

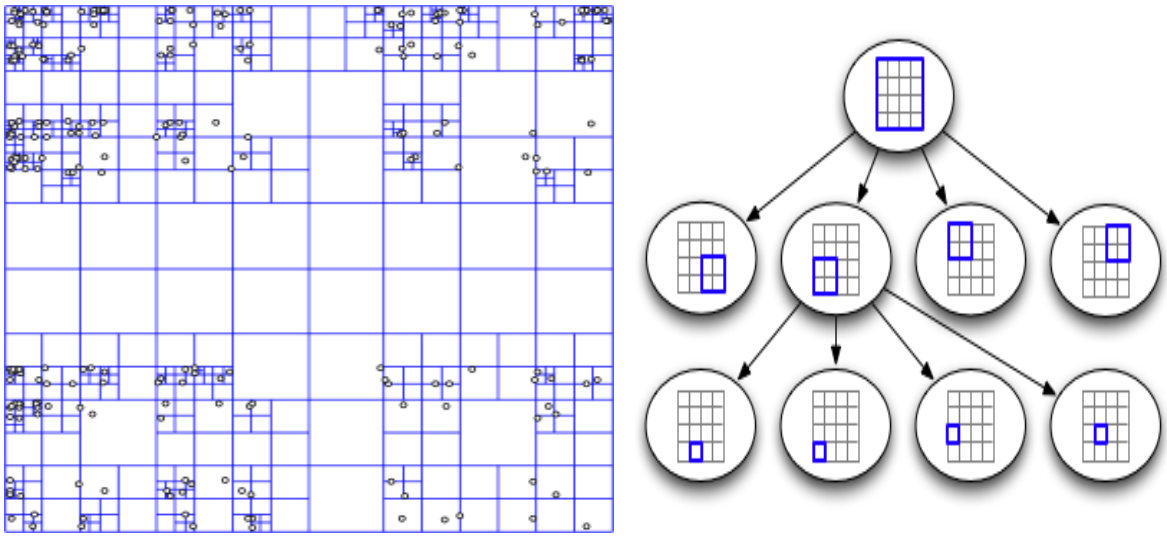


Figure 2: Representation of points in 2-d using quadtrees.

IV. Our contribution

Both quadtrees and Hilbert ordering can be used store the data. The problem faced when using quadtree is lot of unused slot when division is made. So the height of the tree increases. When Hilbert ordering is used to store the data, the order increases, h-number length also increases. Increase in the h-number beyond the Maximum integer is not preferable because most of the coding and decoding of Hilbert labeling algorithm operates on bit manipulation.

Divide the entire map of the city 4 equal squares. Divide each subsquare into 4 subquares. keep on dividing this till each subsquare contains required number of points that can be is approximately equal to N= 2^{32} x 2^{32} · L et us call this subsquare as region. Now divide each regions into small subsquar e till each one them can hold the number of points that a page can contain. Each node in the quadtree contain the data of the form(x,y, width>. This is well explained in figure 2. Now this information has to be stored. Now we label each square using Hilbert curve. Store data point(subsquare) in the order of Hilbert label. The resulting method enjoys the best of both worlds, avoiding the drawbacks of its individual transformation.
The method works as follows (see Figure 3 & 4):

Step 1. Divide the region into four quadrants. Sub divide each quadrant into four sub quadrants. Continue this process until each quadrant contains maximum M points. Let us call each quadrant as cell. This transformation is called first transformation.
Step 2. A distance preserving mapping [11] is used to map the points present in each cell to a point in a 1-dimensional space. This transformation is called second transformation.

Algorithm for transformation:
The Quad tree each node contains <x,y, width, NE,NW,SE,SW> where x,y is the position of top left corner of the square width is the length of the side. NE,NW,SE,SW point to subsquare of the parent square.

1. Convert the point to the region coordinate called (p,q)
   Current node = root
   Found=false
2. repeat
   If p < x+width and q<y+width
   If curr node is a leaf
   Found = true
   Else
   If (p < x->NW+width and q<y+N+width)
   Currnode = currnode->NW
   Else If (p < x->NE+width and q<y+NE+width)
   Currnode = currnode->NE
   ELSEIF
   If (p < x->SW+width and q<y+SW+width)
   Currnode = currnode->SW
   ELSE

   Currnode = currnode->SE
   ENDIF
   Until found
3. map (x,y) to hilbert space called P
4. Locate P in the corresponding region which is labelled according to H-label as shown in Figure 4. Ordering of the will depend upon the number points present in the region.

The currentnode in step 3 identifies the in which the point is present.

Analysis:

The hybrid structure of quadtree and Hilbert curve mapping can benefits of both storage and retrieval. It is based on splitting the area without considering the objects inside. Hilbert mapping indexes depending on the objects inside the region. This is well explained with Figure3(a), Figure3(b) and Figure(c). Assuming m=2, we have split the objects and represented. Now objet present in the each quadrant is labeled using Hilbert curve. The nodes that are shaded are leaves contain zero objects. Other leaves point to the corresponding Hilbert list. Hilbert list consists of number and the corresponding page address of the secondary storage device.

Time required to search for accessing the region is equal to the height of the tree. Number regions created step1 will be N/M where N is the total number of points. M is the highest Hilbert number that can be handled easily for encoding and decoding. Inserting a point in Hilbert curve tree structure is equal to the order of the curve called H.

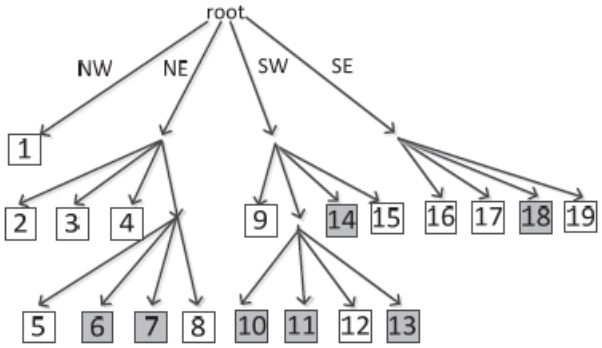


Figure 3(a)

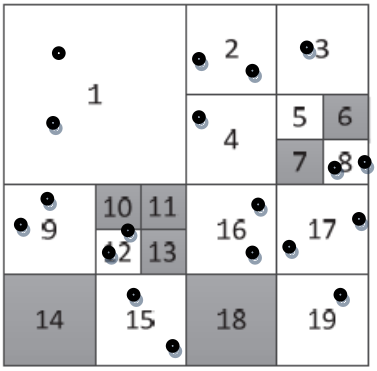


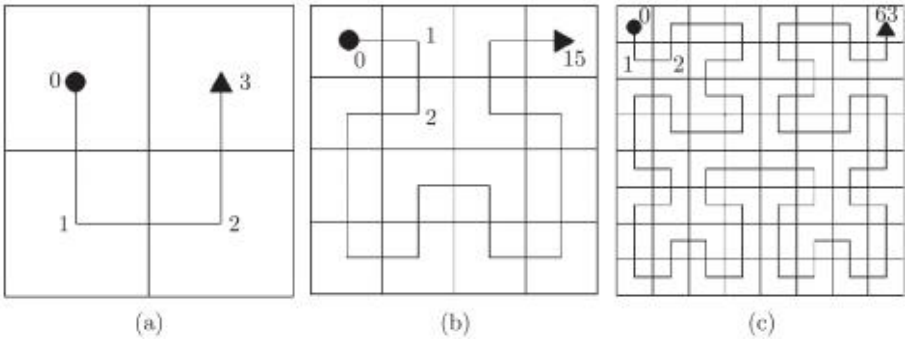
Figure (b)

Figure 4: Hibert Ordering to store Each region in Quadtree

Hibert Index structre for each quadrant is as follows.

`

Pointer from the quadtree leaf	Position of the quadrant in the entire space	Hilbert labeling of the objects



V. Conclusion

Most of tree structure algorithm optimizations are application oriented. At the same time, most of multi dimensional applications focus on sequential reading. Objects searching operations will be performed more frequently than objects insertions or deletions; the H-ordering-Quad tree is an appropriate choice for data structure. The organization of data nodes depends on which kind of data access is needed.

References

[1] J Alsina, The Peano curve of Schoenberg is nowhere differentiable, Approx. Theory, Vol. 33, pp. 28-42, 1981.
[2] M F Barnsley, Fractals everywhere, Academic Press, Second Edition, 1993.
[3] B B Mandelbrot, Fractals: Form, Chance and Dimension, W H Freeman and Co., San Francisco, 1977.
[4] L Velho and J Gomes de Miranda, Digital half toning with space-filling curves, Computer Graphics, Vol. 25, pp. 81-90, 1991.
[5] H-O Peitgen, H Jurgens and D Saupe, Chaos and fractals, New frontiers of Science, Springer-Verlag, p.l02, 1992.
[6] N Wirth,Algorithms + Data Structures =Programs, Prentice-Hall, Eaglewood Cliffs, New Jersey, 1970.
[7] L M Goldschlager, Short algorithms for space-filling curves, Software 10] R. A. Finkel,J. L. Bentley. Quad trees a data structure for retrieval on composite keys[J]. Acta Informatica. 1974
[8] H SAMET. The quadtree and related hierarchical data structure ACM Computer Survey. 1984, 16(2): 187-260.
[9] Ibrahim Kamel,Christos Faloutsos. Hilbert R-Tree: An Improved R-Tree Using Fractals. ISR; TR 1993-19.
[10] Finkel, R. A. and J. L. Bentley (1974). "Quad trees a data structure for retrieval on composite keys." Acta Informatica 4(1): 1-9.
[11] L. Balmelli, J Kovacevic,M. Vetterli. Quadtrees for embedded surface visualization: constraints and efficient data structures, 1999 International Conference on Image Processing, ICIP 99, KOBE, JAPAN, IEEE Society. 1999:487-491, vol.482
[12] Matthew T. Dougherty, Michael J. Folk, Erez Zadok, et al. Unifying biological image formats with HDF5[J]. Communications of ACM. 2009, 52(10): 42-47

