



## Enhancing Quad Tree for Spatial Index Using Space Filling Curves

Ali A. Hussain<sup>a\*</sup>, Rehab F. Hassan<sup>b</sup> 

<sup>a</sup> Department of Computer Science, University of Technology, Baghdad, Iraq.  
[a.u.h.altalby2018@gmail.com](mailto:a.u.h.altalby2018@gmail.com)

<sup>b</sup> Department Computer Science, University of Technology, Baghdad, Iraq. [110019@uotechnology.edu.iq](mailto:110019@uotechnology.edu.iq)

\*Corresponding author.

Submitted: 30/04/2020

Accepted: 19/07/2020

Published: 25/04/2020

### KEY WORDS

Quad trees, Space filling curves, Spatial Index, Z-order curves, Hilbert curves.

### ABSTRACT

*Spatial indexes, such as those based on the Quad Tree, are important in spatial databases for the effective implementation of queries with spatial constraints, especially when queries involve spatial links. The quaternary trees are a very interesting subject, given the fact that they give the ability to solve problems in a way that focuses only on the important areas with the highest density of information. Nevertheless, it is not without the disadvantages because the search process in the quad tree suffers from the problem of repetition when reaching the terminal node and return to the behavior of another way in the search and lead to the absorption of large amounts of time and storage. In this paper, the quad tree was improved by combining it with one of the space filling curve types, resulting in reduced storage space requirements and improved implementation time.*

**How to cite this article:** A.A. Hussai and R.F. Hassan, "Enhancing quad tree for spatial index using space filling curves," *Engineering and Technology Journal*, Vol. 38, Part B, No. 01, pp. 15-25, 2020.

DOI: <https://doi.org/10.30684/etj.v38i1B.264>

This is an open access article under the CC BY 4.0 license <http://creativecommons.org/licenses/by/4.0>.

### 1. Introduction

Spatial indexing is based on physically grouping indexed items. For instance, it is possible to group countries based on their continent. With spatial indexing, there is even a possibility of providing VR (virtual reality) fly through (or over, or under) for helping the users in the maintenance of their search context as they expand or narrow the range of their search. Spatial indexing has a richness, which is capable of paralleling the indexing according to subjects or authors. The spatial indexing concept is quite powerful, that for managing records, records are detected according to how they are associated with a place. A considerable number of records is strongly associated with a place. Such as other indexing forms, geographic indexing may be joined with other forms of indexing [1].

Quad trees are very important, as they solve problems and focus on working in areas where spatial data are more intense than others.

In many places, the volume of work corresponds to the number of lumps combined (Such as tree structure nodes) shorten the actual number of pixels in the original Picture.

This way, they can potentially reduce storage space requirements and enhancing the effectiveness of execution time. Data-base Management system is capable of easily handling data in the case where objects are 1-D in nature. Objects that are to be stored are of more than one dimension, to do this; it will be required to use space filling curves. There is a considerable number of space filling curves that were mentioned in literature. Throughout the early days space filling curves have been mainly viewed as a mathematical curiosity. Nevertheless, nowadays, space-filling curves are being implemented in a wide variety of fields, like, reduction of space dimension, grid computing load balancing, I/O-efficient calculations on massive matrices, design of small antennas, and creating spatial data indexes. In the presented research, the emphasis has been placed on applying space-filling curves in creating query-efficient indexes of spatial data with the use of the Z-order curve [2]. The presented paper has been organized in the following: Section 2 provides related works. Section 3 describes about quad trees .section 4 gives a brief survey about Z-order curve. Section 5 Method for quad tree building, Section 5 experimental results. Section 7 includes a presentation of the conclusions and future researches

## 2. Related Work

This section presents the studies about the spatial index with several published researches related to the goals of this work:

Sardadi extensive study was undertaken on the design of effective data structures to enable rapid spatial research. Commercial database suppliers such as Oracle have begun to implement this spatial indexing to meet the needs of large and diverse GIS. This paper focused on the choice of spatial indexing of the R and quad tree using the spatial Oracle database in the mobile GIS application. The use of quadratic spatial data indexing and R-tree in a single spatial database saved up to 42.5% [3].

Lim presented a method where, classification of packets is an important function for next-generation Internet routers to provide high quality of service. Then the priority area-based quad-tree (PAQT) packet classification algorithm combining priority search and recursive space decomposition is formally described using the framework. Extensive simulation results have demonstrated that PAQT algorithm performs quite well in comparison with other approaches of packet classification concerning the speed of search, the size of the memory, in addition to scalability [4].

Qasem used a Quad Tree to find the constant neighbor on time, four main neighbors are identified for every one of the quadrants. Those designated neighbors will permit accessing every neighbor on every side in the phase of recovery. This access is done at a fixed time for cardinal neighbors independent of their size. The proposed method required a 31% increase in quad tree memory, allowing an average of 69.7% at the time of arrival of neighbors as described by experimental results. The CN-Quad tree will provide opportunities for new improvements in image analysis and processing, robot routing and mobility planning [8].

Kumar based on the rapid motion estimation algorithm and the structure of the hybrid quad tree, these algorithms are suitable in the search for large, small and very small movements respectively, the resultant algorithm is much stronger for complicated sequences of movement, obtained smaller motion projection quality errors, and has consistent computational complexity in the video sequence [5].

Zhou presented a method where, the polygons that intersect with the axes of the quarters were stored in the five bulldozers to hold a different root level, and the polygons were separated into all axes of the quarters stored in the paper nodes. By storing embedding relationships between complicated polygons and their holes in an explicitly inserted table. Index operations (such as insertion, deletion and query) were performed and the improved indicator effectiveness was verified through the incremental land cover data update experiment. Experimental results showed that the sufficiency of incremental updating time has enhanced markedly, and that the enhancement is growing with the volume of data. Compared to other approaches [6].

## 3. Quad Tree Basics

Quad trees are quite a straight forward method of spatial indexing. In a Quad tree, every one of the nodes denotes a bounding box that covers part of the space that is being indexed, with the root node that covers the whole area. Every one of the nodes can either be a leaf node - where it contains one or more points of indexing, and does not include children, or it can be an internal node, where it has precisely 4 children, one for every one of the quadrants that have been obtained via dividing the area

which is covered in half along the two axes. For querying a quad tree, beginning from the root, the first thing to do is examining every one of the child nodes, and checking whether it is intersecting the area which it is being queried for. In case where it does, recourse to that child node. In the case of encountering a leaf node, every entry should be examined in order to see whether it is intersecting with the area of the query, and if so, return it [6].

In Quad tree every one of the nodes includes  $\langle x, y, \text{width}, \text{NE}, \text{NW}, \text{SE}, \text{SW} \rangle$  where  $x, y$  represents the location of top left corner of the square width is the side length. NE, NW, SE, SW point to sub square of the parent square, as shown in the Figure 1.

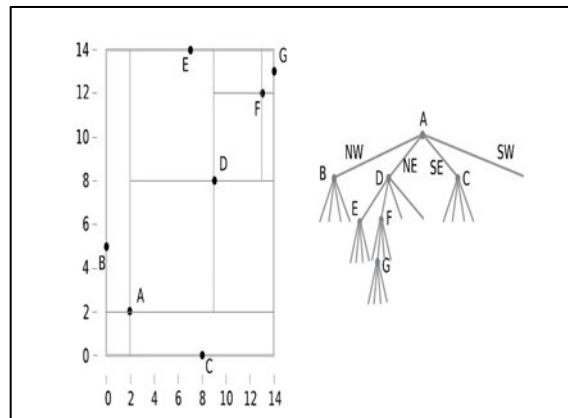


Figure 1: Point quad tree [8]

#### 4. Z-ordering Space Filling Curves

For the sake of mapping multidimensional objects of data to one-dimensional objects, a number of space filling curves were developed. The majority of those curves are continuous self-similar fractal curves. There are two types of curves that are commonly accepted as approaches for spacing filling for the existing multidimensional applications, which are: Z-Ordering Curve and Hilbert Curve, as shown in the Figure 2 [7]. As it can be deduced from its name, this curve has been presented by D. Hilbert in 19-th century. The inner-most black line in Figure 3(a) is a 1st order Hilbert curve, the central black line is a 2nd order Hilbert curve, and the grey line is a 3rd order Hilbert curve. It shows the way a Hilbert curve is capable of recursively exploring a space. By those lines, the rectangles or particles in this two-dimensional space may be traced in a linear manner. Which offers the possibility of mapping two-dimensional objects to one-dimensional objects. Advanced algorithms are available for calculating the Hilbert order that are important for performing operations like search, delete and insert. This Curve, in combination with an R-tree data structure is commonly utilized in multidimensional applications, as shown in the Figure 3 [7].

Morten has proposed Z-ordering curve in the year of 1966. A point's z-value in a multidimensional space is simply computed via interleaving the binary representations of the values of its coordinates, as shown in the Figure 4.

As soon as the data is sorted to this ordering, any 1-D data structure may be utilized like B-trees, binary search trees, skip lists or (with low significant bits truncated) hash tables, as shown in the Figure 5. [7].

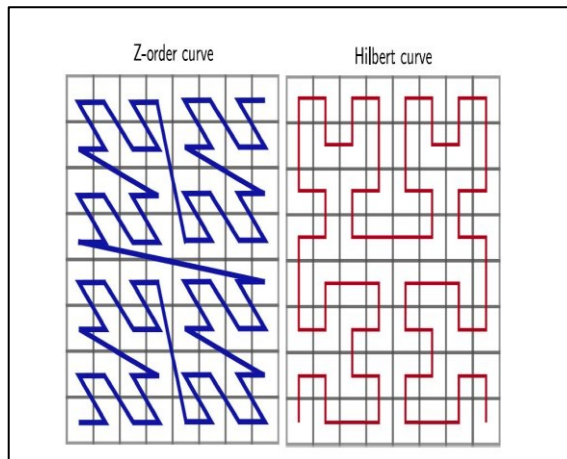


Figure 2: The Z-order curve and Hilbert curve [9]

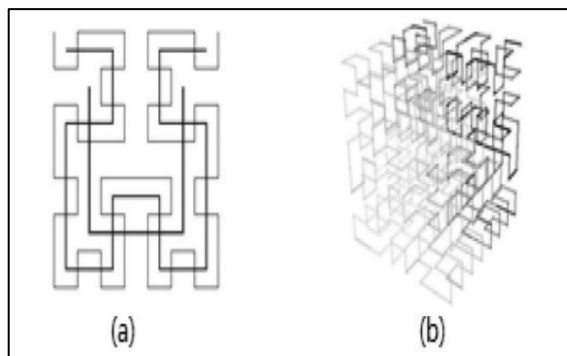


Figure 3: Hilbert Curve in two-dimensional and three-dimensional (a) is a two-dimensional Hilbert Curve (b) is a three-dimensional curve

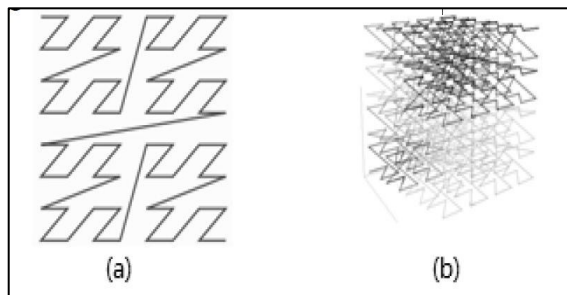


Figure 4: Z-Ordering Curve in a) 2-D b) 3-D

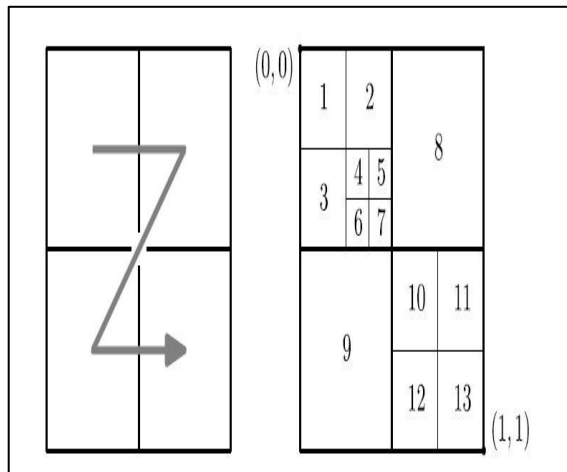


Figure 5: Left: the order where the Z-order curve traverses the 4 canonical square quadrants. Right: a resulting ordering example [10].

## 5. Improved Method for Quad Tree

Z-order may be utilized to create a quad tree structure and associated data structures of higher-dimensions. The order of how these four sub-regions are stored has a close connection to the curve filling the area. If a multidimensional application implements a Z-order curve, creating Quad tree in SE, SW, NE, and NW are going to be of a higher efficiency to access the data. Where the work is in two steps:

The Quad-tree splits the rectangle area to four parts in a recursive way. All parts are equally sized, as shown in the Figure 1.

Step1: Divide the area to 4 quarters. Sub divides every one of the quadrants to 4 quadrants. Repeat this process so that every one of the quadrants has the lowest number of points. This shift is called the first shift.

Step 2: The Preserve Distance parameter is utilized to assign points in every one of the cells to a point in a distance of one dimension. This transformation is called the second shift.

### Quad tree building & Z-order algorithm

**Input:** - Generate random points in a Specific region

**Output:** - one-dimensional sequential points (p1, p2, p3... pn)

1. Converting the point to an are coordinate which is referred to as (p, q)

Current node = root

Found=false

2. Repeat

If  $p < x + \text{width}$  and  $q < y + \text{width}$

If current node is a leaf

Found = true

Else

If ( $p < x - \text{NW} + \text{width}$  and  $q < y + \text{N} + \text{width}$ )

Current node = current node->NW

Else If ( $p < x - \text{NE} + \text{width}$  and  $q < y + \text{NE} + \text{width}$ )

Current node = current node->NE

Else If ( $p < x - \text{SW} + \text{width}$  and  $q < y + \text{SW} + \text{width}$ )

Current node = current node->SW

ELSE

Current node = current node->SE

ENDIF

Until found

3. Map (x, y) to Z-order space or Hilbert order referred to as P.

4. Production sequence of p1, p2... pn as shown in the Figure 6.

The data used for which q is applied is a structure with certain dimensions. Random points were generated within the blue color. The number of random points in each experiment was changed.

A single point of red color was also included in fixed coordinates in that structure to be searched. Each time and for each of the normal and hybrid quad tree cases, as shown in the Figure 7.

After applying the regular quad tree on the above structure, give it a fixed number of minimum points to divide the quad tree into four quadrants which is  $k = 10$ , as shown in the Figure 8.

When experimenting with the hybrid quad tree, quad tree with H-curves resulted in the Figure 9.

Similarly, when experimenting with the hybrid quad tree, quad tree with Z-order resulted in the Figure 10. And when the mechanism of work quad tree on random points and then indicate the stage of integration with one of the types of space filling curves to convert those points sequentially from 2-dimension to 1-dimension, as shown in the Figure 11.

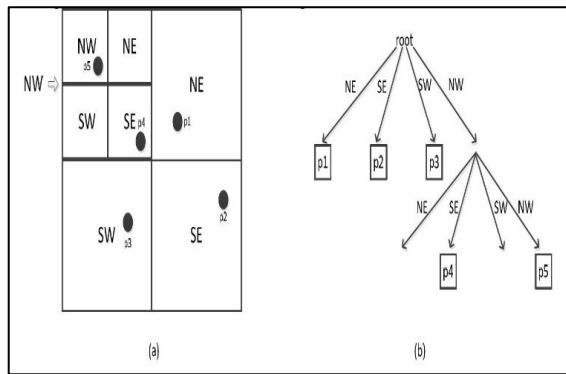


Figure 6: An example of Quad-tree (a) Area Partition (b) Quad-tree representation [7].

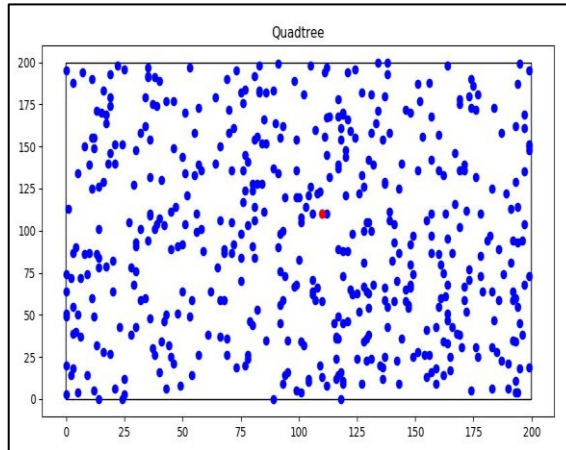


Figure 7: An example is to generate random dots of 500 points and insert a red dot (110,110) point that is searched for each time

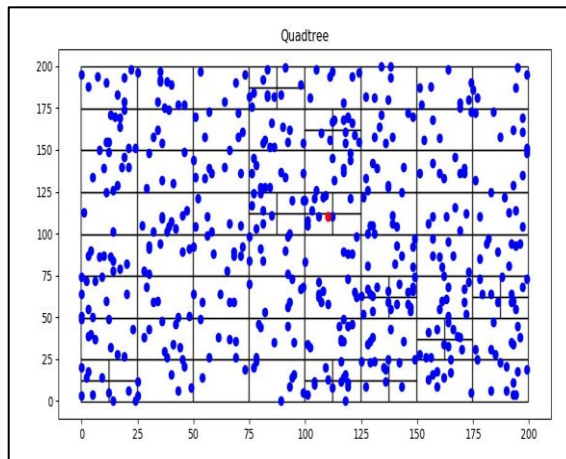
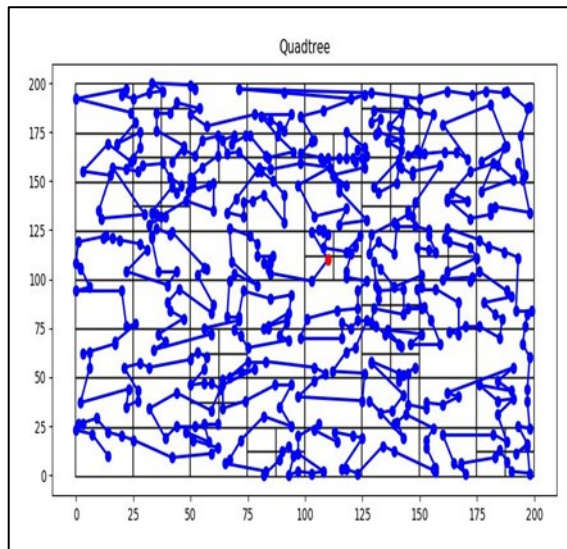
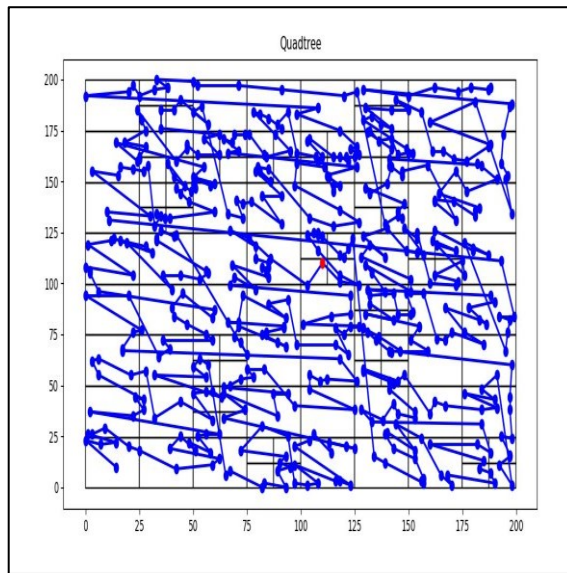


Figure 8: The Example after applying the quad tree



**Figure 9: H-curve with quad tree**



**Figure 10: Z-curve with quad tree**

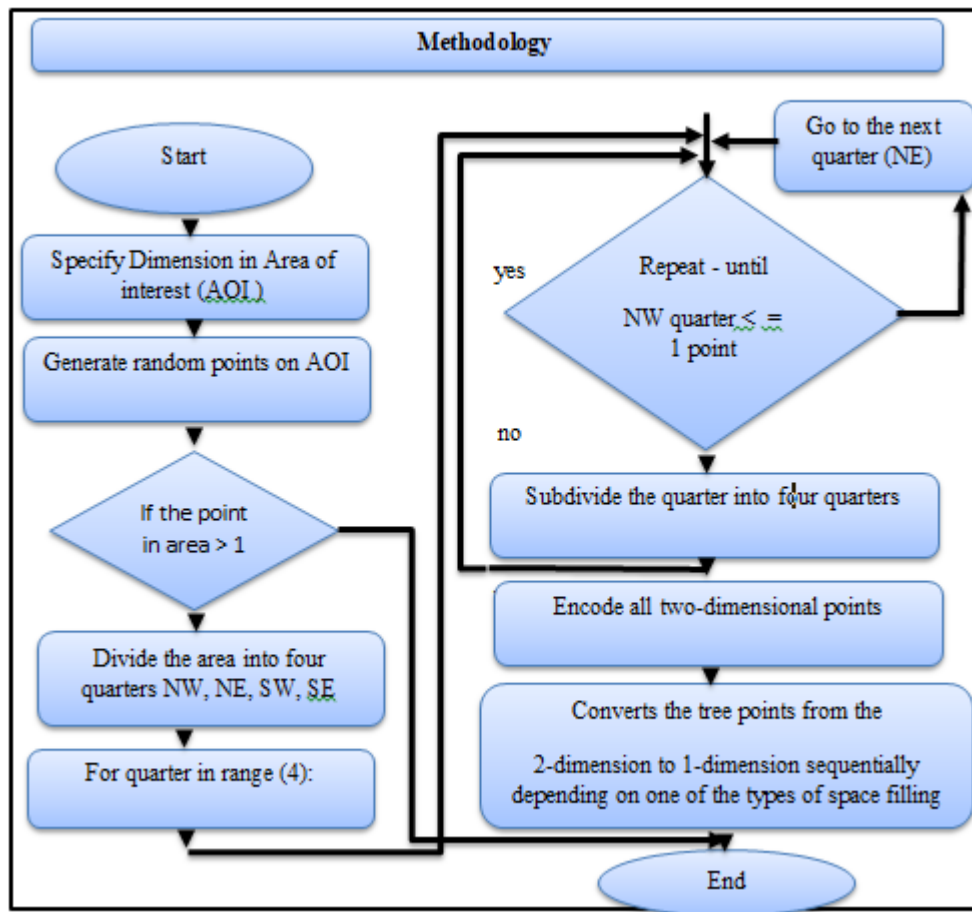


Figure 11: Flowchart of the Quad tree building with Z-order algorithm

## 6. Performance Results

In the spatial indexing process, the hybrid structure of quadruple tree and Z-order maps may benefit from both recovery (time) and storage (size).

The objects are divided and represented, determining the minimum number of points that divide the quarter division into quarters is  $k = 10$ , the work has been applied to random points generated, where each point coordinates  $(x, y)$  on the structure used.

In the same way, when calculating the amount of time it takes, it will retrieve the red dot information that is  $(X: 110, Y: 110)$  which are searched each time for a different number of points. The time taken when applying a quadruple tree alone is much greater than applying the hybrid state of Z-curves with a quad tree and H curves with a quad tree, as shown in the Table 1. When the time results are represented by the graph below, we note that the time taken has greatly improved the hybrid state with the quad tree, as shown in the Figure 12. After quad tree is applied to the points, the required amount of storage was calculated in bytes and the hybrid structure of the quadrature and z-curves were applied. This has led to a significant reduction for storage required, where the sequence of points in the hybrid state has been.

-Note that the hybrid structure is equal in terms of storage for its quad tree with z-curves and H-curves for each number of points. The experiment was conducted each time as shown in the Table 2. When the above storage results are represented by the graph, the difference is obvious. The black line represents a quadruple tree; the red line represents Z curves with a quad tree, while the blue line represents H curves with a quad tree, as shown in the Figure 13. The index size of the quad tree data is reduced by the same amount for both the Hilbert curve and Z-order curve.

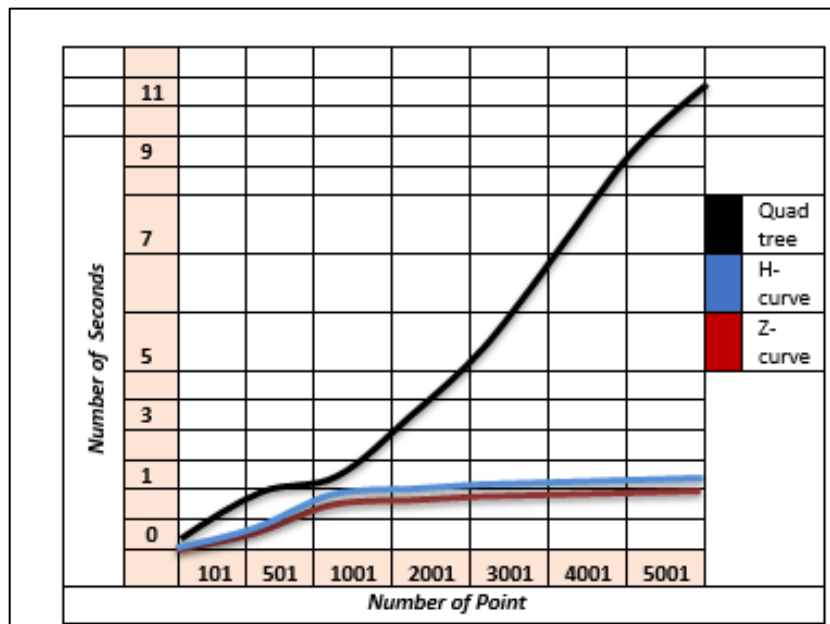


**Table 1: Response time taken for each case, with number of k = 10.**

size			
Number of points	Quad tree (byte)	H-curve & quad tree (seconds)	Z-curve & quad tree (seconds)
101	984	404	404
501	7488	2004	2004
1001	18008	4004	4004
2001	38548	8004	8004
3001	64188	12004	12004
4001	88424	16004	16004
5001	111972	20004	20004
15607	329612	62428	62428

**Table 2: The storage size taken for each case, with number of k = 10.**

Time			
Number of points	Quad tree (seconds)	H-curve & quad tree (seconds)	Z-curve & quad tree (seconds)
101	0.112	0.0	0.0
501	0.941	0.001	0.001
1001	2.225	0.002	0.001
2001	3.996	0.004	0.004
3001	6.984	0.006	0.006
4001	9.651	0.008	0.008
5001	11.990	0.01	0.011
15607	35.899	0.031	0.031



**Figure 12: Average Response Time**

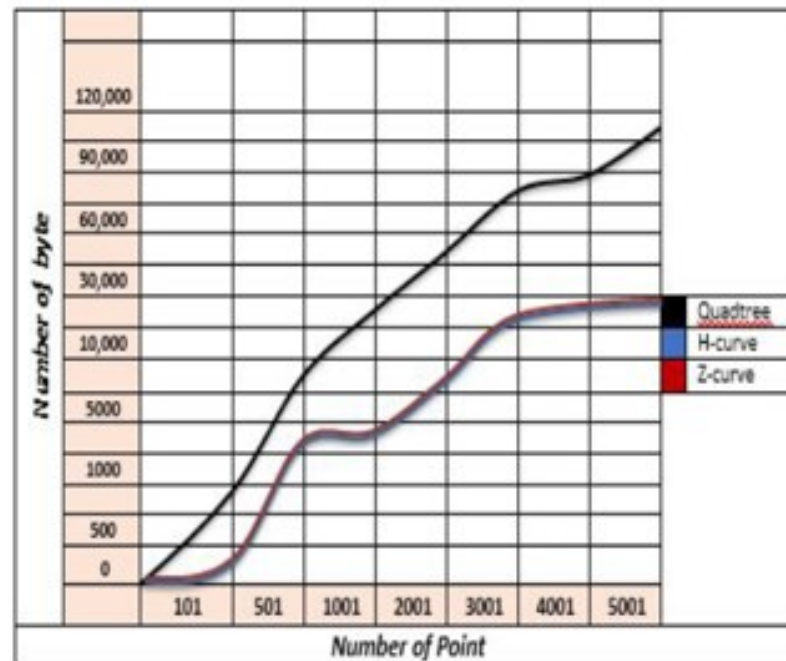


Figure 13: As the required storage for each case

## 7. Conclusion

The majority of quad tree algorithm improvements depend on the speed of access and the amount of storage needed to represent the data and index them spatially, and most multidimensional applications focus on sequential reading. As storing and searching for objects will be performed very efficiently. In this paper, the quad tree algorithm is optimized by hybridizing with some types of space filling curves, which are Z-curve and H-curve. They arrange the points sequentially and transform them from 2-dimension to 1- dimension, so that the occurrence of the recurrence is eliminated in the normal quad tree search process. When a particular node is searched in quad tree and reaches To the terminal node in the tree, it returns to repeat the contract back to take another path, and it takes a lot of time and storage, The method was applied to randomly generated points using normal quad tree and quad tree hybrid and found a significant improvement in time and storage required.

## References

- [1] M. M. Sardadi, M. M. Rahim, Z. Jupri, D. Daman, "Quad tree spatial indexing use to make faster showing geographical map in mobile geographical information system technology using an oracle 10g application server and map viewer applications," *IJCSNS International Journal*, Vol.8, No.10, 81310, 2008.
- [2] S. S. Baboo, V.Narmadha, "Efficient data storage and searching for location based services using quad trees and H-ordering," *International Journal of Computer Science and Mobile Computing*, Vol.3 Issue.4, 810-816, 2014.
- [3] M.M. Sardadi, M.S. Mohd Rahim, Z. Jupri, D. Daman, "Choosing R-tree or quad tree spatial data indexing in one oracle spatial database system to make faster showing geographical map in mobile geographical information system technology," *International Journal of Humanities and Social Sciences*, 2009.
- [4] H. Lim, C. Yim, E. S. lander, "Priority Area-based quad-tree packet classification algorithm and its mathematical framework," *International Journal Applied Mathematics & Information Sciences Appl. Konkuk University, Math. Inf. Sci.* 7, No. 1, 2013.
- [5] G. S. Kumar, D. N. Kumar, A. V. Paramkusam, "A fast motion estimation algorithm based on hybrid quad tree structure," *International Journal of Mechanical Engineering and Technology (IJMET)*, Volume 8, Issue 7, July 2017, 642–648, Article ID: IJMET\_08\_07\_072, 2017.
- [6] X. G. Zhou, H. S. Wang, "A quad tree spatial index method with inclusion relation for the incremental updating of vector land cover Database," *The International Archives of the Photogrammetry, Remote Sensing and Spatial, Information Sciences*, Volume XLII-4, 2018.

- [7] F. Hou, C. Huang, J. Lu, "A multi-dimensional data storage using quad-tree and Z-ordering," International Conference on Computer Science and Electronics Engineering (ICCSEE), 2013.
- [8] N. Xiao, "GIS algorithms theory and applications for geographic information science & technology," California, USA, ISBN 978-1-4462-7432-3, 2016.
- [9] S. SIERANOJA, P. Frenti, "Constructing a high-dimensional kNN-graph using a Z-order curve," International Journal of Mechanical Engineering and Technology (IJMET), Volume 8, Issue 7, Hyderabad, Telangana, India, pp. 642–648, 2018.
- [10] H. Haverkort, S. Thite, L. Toma, "Star-quad trees and guard-quad trees: I/O-efficient indexes for fat triangulations and low-density planar subdivisions," Computational Geometry 43, 493–513, 2010.
- [11] S. W. Qasem, A. A. Tourir, "Cardinal neighbor quad tree: a new quad tree-based structure for constant-time neighbor finding," International Journal of Computer Applications, Vol. 132, No.8, 0975- 8887, 2015.