Journal of Al-Qadisiyah for Computer Science and Mathematics Vol.11(3) 2019 , pp Comp.94–104

University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Using space-filling curves to improve the quad tree for spatial indexing

Rehab Flaih Hassan [a] , *Ali Abid Hussan Azeez* [b]

[a] *Department of Computer Science, University of Technology, Baghdad, Iraq,email: 110019@uotechnology.edu.iq.*

[b] *Department of Computer Science, University of Technology, Baghdad, Iraq,email: a.u.h.altalby2018@gmail.com.*

ARTICLE INFO

Article history:
Received: 16 / 07 /2019
Rrevised form: 00 /00 /0000
Accepted : 22/07 /2019
Available online: 02/ 09/2019

Keywords:

Space filling curve, quad trees, Z-order curves, Hilbert curve (H-curve), multi-dimensional, data structure.

ABSTRACT

Spatial indexes, like the ones that are based on Quad Tree, are important in spatial data-bases for the effective implementation of queries with spatial constraints, particularly in the case where queries include spatial links. The quad 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. But it is not without the disadvantages because the search process in the quartile suffers from the problem of repetition when reaching the terminal node and return to the behaviour of another way in the search and lead to the absorption of large amounts of time and storage. A database management system can handle data very easily if the object is one-dimension (sequential).

In this paper, improve the quad tree by combining one of the space filling curve types, including the Hilbert curve and the Z-ordering curve with a quad tree. It will convert from two-dimensional to one-dimensional and sequentially search and end the problem of repetition whenever it reaches a terminal node Ordinary quad tree. Resulting in reduced storage space requirements and improved implementation time.

MSC.

1. Introduction

Spatial indexing is based on physically grouping indexed items. For instance, countries may be grouped according to the continent. With spatial indexing, there is even a possibility of providing virtual reality (VR) fly through (or over, or under) for the sake of helping users in maintaining their search context while users narrow or expand their search domain.

Spatial indexing has richness which parallels the indexing with an author or a subject. Spatial indexing concept is quite powerful, that for the management of records, they are found according to their correlation with a place. Numerous records are strongly coupled to a place. Like other indexing forms, geographical indexing can be combined with other indices. There is a wide range of cases when we need to find a subset of spatial information for various Purposes. For example, when we talk about things that happened in the past, we try to link them with the years or our ages, and in this way we treat the events as a time sequence and index the events with time. The power of indexing lies in a commonly used strategy humans have used for thousands of years: divide and conquer.

Corresponding author Ali Abid Hussan Azeez
Email addresses: a.u.h.altalby2018@gmail.com
Communicated by Qusuay Hatim Egaar

Quad trees are a very straightforward spatial indexing technique; effectively we divide the whole search area into four quarters and ignore three quarters when our target is definitely not there. We keep on dividing the search space into four quarter until we find what we need [7].

A space filling curve (space filling curves) is a continuous path that passes through each one of the points in a space, which gives a one-to-one correlation between point coordinates and the 1-D-sequenced numbers of the points on the curve. Space-filling curves were commonly utilized in math and for the transformation of multi-dimensional tasks to 1-D forms. For scientific applications, ordering calculation or data along space-filling curves may be utilized to exploit locality in the case of partitioning to parallel systems or in the case of restructuring for exploiting memory hierarchy [3].

The presented paper has been organized in the following: Section 2 discusses related works. Section 3 includes a description about quad trees, section 4 gives a brief survey about Space filling curves. Section 5 explains method for quad tree building, Section 6 gives experimental results. Section 7 includes the paper conclusions.

2. Related work

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

1. Julian Hirt (2010) Implemented a method for path finding on a 3m * 3m board for the mobile robot, combining the A* algorithm with the quad tree, resulting in an optimal path and shorter time than using the A* algorithm with the regular Grid [2].
2. Amitava Chakraborty (2011) balanced the quad tree with the use of point pattern analysis; because the effectiveness of the search approach is dependent on the tree height, arbitrarily inserting point features could result in the tree being unbalanced and result in longer searching time. Post implementing this algorithm, it has been noticed that its performance enhanced [8].
3. A. A. El-Harby (2012) presented a new Quad tree-based Color Image Compression Algorithm By dividing the color of the image into RGB components and then dividing the blocks on the way of the quad tree The ratios of compression of the second were in the range from 0.25 to 0.80 at a value of threshold equal to 0.1, and from 0.78 to 0.94 at value of threshold which is equal to 0.2 [4].
4. Phyo Wai (2014) Provide a proposed compression system for GIS by Using a quad tree way that is an integral part of Cluster encoded method. This system was produced to be more convenient to pressure more than other pressure methods. It is effective in sending data over the current network [1].
5. Ashwaq Talib Hashim (2016) presented a color image compression approach for increasing the ratio of compression with no impact on the original scene by distortion or noise. With the use of low lossy rate quad-tree compression approach for increasing the association amongst pixels with the use of Quantizing and encoders of entropy like shift encoding and run length encoding will additionally compress the image. the results of CR (i.e. Compression Ratio) of the presented approach on average to be about 1:29 of the original image size, higher ratio of compression may be accomplished with the increase of the levels of compression, this high ratio of compression has been considered an optimal ratio in comparison to the achieved PSNR (Peak Signal to Noise Ratio) of decompressed-compressed image [6].
6. Satori Tsuzuki (2019) a method was proposed to create a central graph of the geometrically complex road map using a quad tree pyramid to simulate cellular automation and trace the distant leaves of the tree that used Morton's curve. The leaves were chosen to maintain a certain distance between the previously selected sheets as a graph. Each specific node searches the adjacent nodes and stores them as glyphs. The shortest path of the resulting graph was produced using the Dextra algorithm. It is quite possible that done develops a new technology to create a suitable network diagram for vehicle simulation studies in transport research [11].

3. Quad Tree Basics

A quad tree is a tree data structure where every one of the internal nodes has precisely 4 children. Quad trees are the 2-D analog of octrees and are usually utilized for partitioning a 2-D space via the recursive sub-division of that space to 4 areas or quadrants. The data which is related to a leaf cell differs according to the application, but the leaf cell denotes a "unit of interesting spatial information".

The sub-divided areas could be rectangular or square shaped, or they could be of a random shape. This data structure has been referred to as a quad tree by R. Finke and J.L. Bentley in 1974. An analogous partitioning is referred to as Q-tree as well. All quad trees forms have some features in common:

1. They decompose space into adaptable cells
2. Each cell (or bucket) has a maximum capacity. When maximum capacity is reached, the bucket splits
3. The tree directory follows the spatial decomposition of the quad tree.

Quad trees could be categorized based on the data type that they represent, and that includes points, areas, curves, and lines. Quad trees could as well be categorized according to whether the tree shape is not dependent on the order of data processing (fig 1) and (fig 2) Represent points and region by quad tree [7]

.

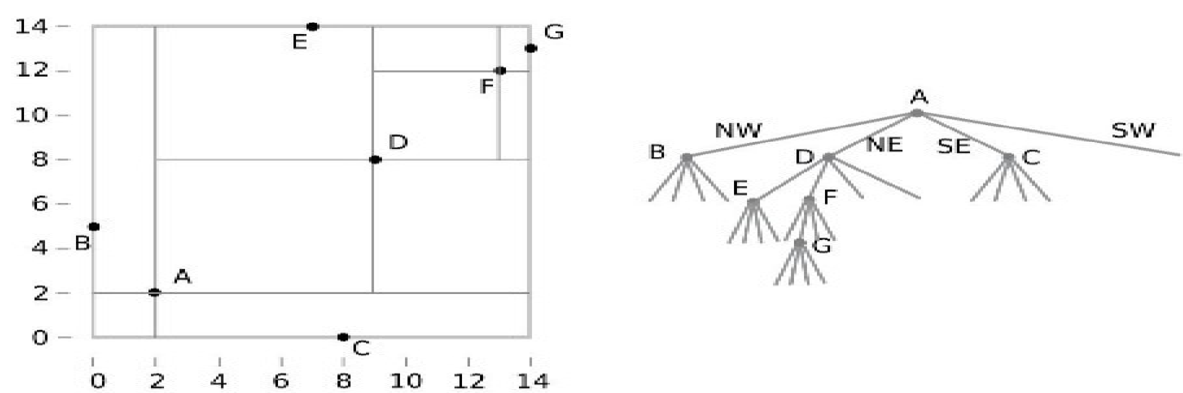

Fig. (1) Points quad tree

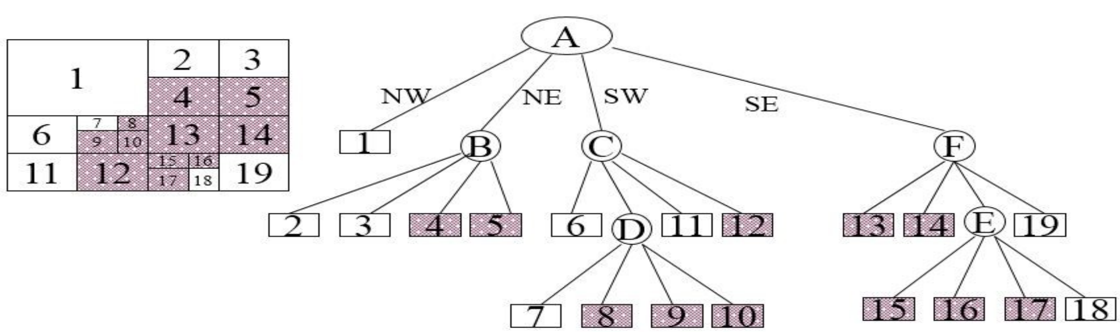

Fig. (2) Region quad tree

4. Space filling curves (SFC)

An SFC is a continuous trajectory visiting each one of the points in space precisely one time in a specific order thereby resulting a one-to-one correlation between points' coordinates and the one-dimensional sequence numbers of points on the curve. Which gives a way for linear ordering of a grid points. The aim is conserving locality, thus the points that are near in space have to be stored close to one another in a linear ordering [2].

The concept of multi-dimensional space mapping to the 1-D space has a significant impact on applications which include multi-dimensional data. Multi-media data-bases, Geographic Information Systems (GISs), and Image processing are examples of multi-dimensional applications. By using a mapping scheme, a point in the D-dimensional space can be represented with one integer reflecting the different dimensions of the original space. Space filling curve can be used for mapping from multi-dimensional space to 1-D space.

There are various space-filling curve types. They differ in the way they map into the 1-D space. Choosing the suitable curve for any application based on knowing the mapping approach which is given by every space-filling curve. Some space filling curve examples are the Hilbert curve, and the Z-order curve [1].

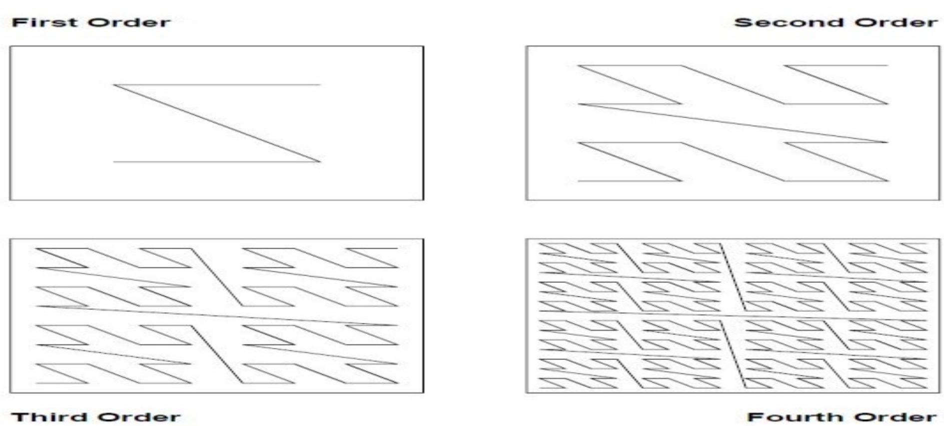

Fig. (3): Z-order curves

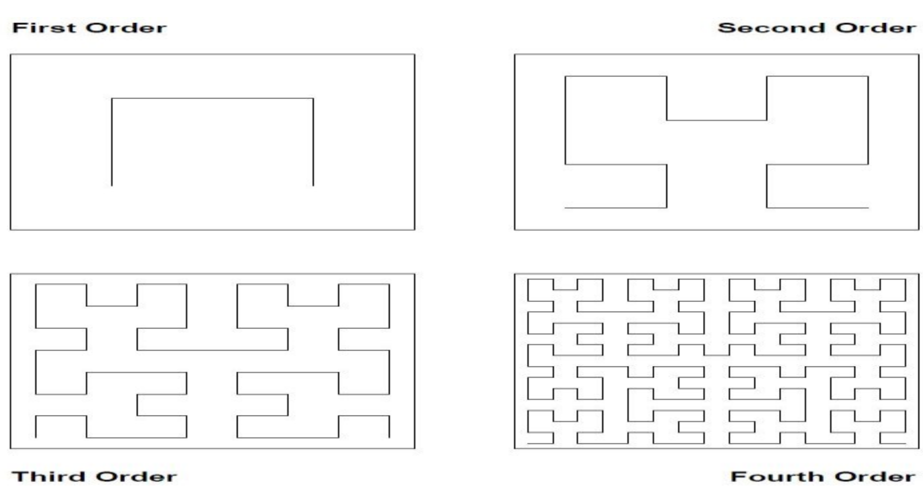

Fig (4): Hilbert order curves

5. Improved Method for quad tree building

The space filling curves may be utilized to create a quad tree structure and construct relevant data of higher dimensions. The order of how these four sub-regions are closely related to the area fill curve is related.

If a multidimensional application applies a Z-order or H-order curve, a quad tree in SW, SE, NW, and NE will be created more efficiently to access the data.

Work has been done on the map of the province of Baghdad, the capital of Iraq, which contains points representing intersections and each point has a two-dimensional events as shown in the fig (5):

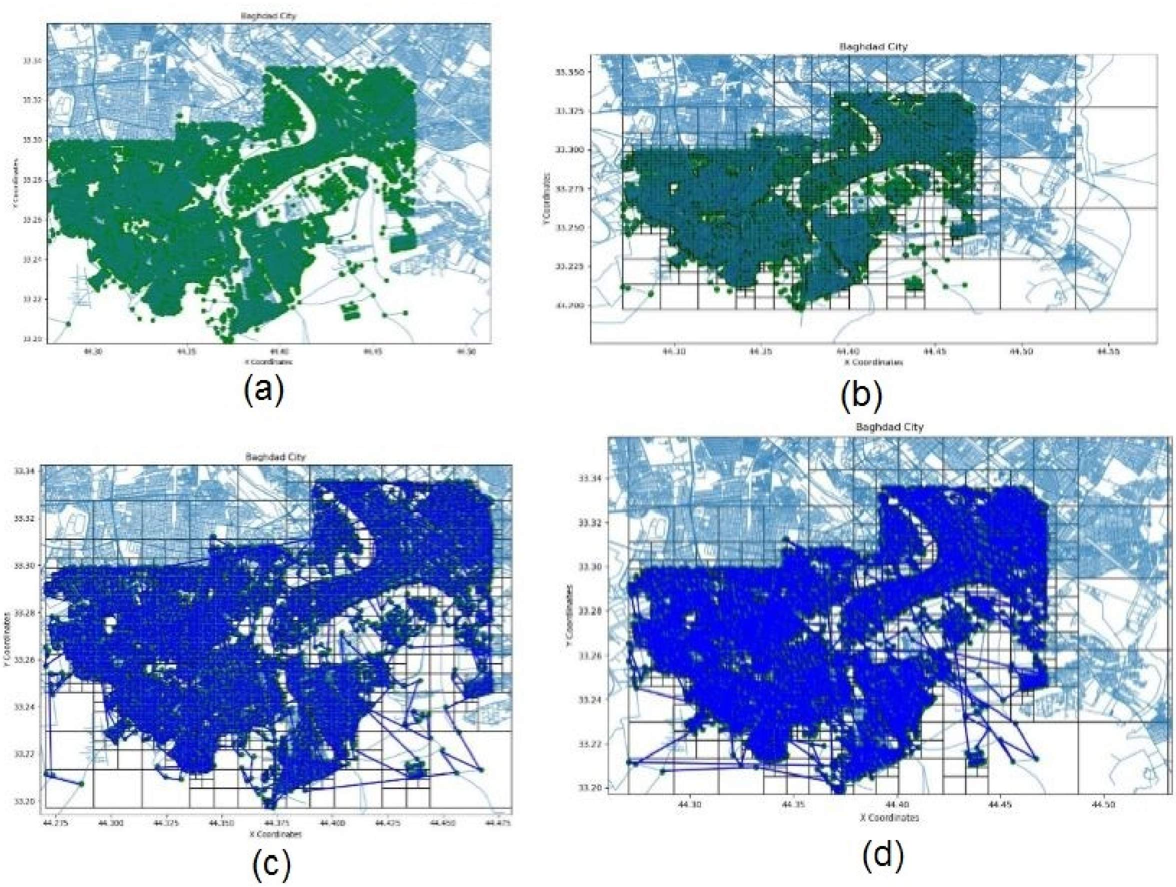

Fig. (5): Stages of representation of the map points (a): map points (b): only quad tree (c): H-curve with quad tree, (d): Z-curve with quad tree

a) Quad tree building algorithm

Input: Points, K // where Points are the coordinates of points in the map, and K is the threshold that uses to divide the quad trees

Output: Quad Tree structure

Procedure:

1- Initialize root that holds all points in one node

Root = Node (Points)

2- Call recursive_subdivide(root, K)
3- Call Nodes = find_children(root)
4- Representing quarters and nodes on the map.

****

b) Z-order & Quad tree building algorithm

Input: Points, K

Output: SequencePointsZOrder

Procedure:

1. Tree = QuadTree(Points,K)
2. Initialize Queue
3. Add Children of Root of Tree to Queue
4. While True
5. If Queue is not empty Then
6. Children = queue.pop()
7. For each Node in children
8. Add Node.Children to Queue
9. For each Point in Node.Points
10. Code = Morton(Z-order)Encoding(Point.x, Point.y)
11. Add Code to SequencePointsZOrder
12. next
13. next
14. Else
15. End
16. endif
17. next

****

c) Quad tree building & H-order algorithm

Input: Points, K

Output: SequencePointsHOrder

Procedure:

1. Tree = QuadTree(Points,K)
2. Initialize Queue
3. Add Children of Root of Tree to Queue
4. While True
5. If Queue is not empty Then
6. Children = queue.pop()
7. For each Node in children
8. For each Point in Node.Points
9. Code = HilbertDistanceFromCoordinates(Point.x, Point.y, 32)
10. Add Code to SequencePointsZOrder

11. next
12. next
13. Else
14. End
15. endif
16. next

6. Performance results

In the process of spatial indexing, The hybrid structure of quad tree and Hilbert curve or Z-order mapping may benefit of each of the retrieval (time) and storage (size). The objects have been split and represented, Determine the minimum number of points at which the division of the quartile is divided into quarters is k = 10 (This means that each quarter contains 10 points or less not divided into four quarters) , Work has been applied to intersections in Baghdad's two-dimensional map, where each point has coordinates (x, y) on the map.

After quad tree is applied to the points, the required amount of storage was calculated in bytes and then the hybrid structure of quad tree and Z-curves, quad tree and H-curves was applied. This resulted in a significant reduction in the amount of storage required, as the points in the hybrid state were serialized.

Note that the hybrid structure is equal in terms of storage for quad tree with Z-curves and quad tree with H-curves for each number of points. The experiment was carried out each time as shown in (table 1).

Table (1): The storage size taken for each case, with number of k = 10

size			
Number of points	**Quad tree (byte)**	**Z-curve with quad tree (byte)**	**H-curve with quad tree (byte)**
3000	**16724**	**12000**	**12000**
6000	**32148**	**24000**	**24000**
9000	**47892**	**36000**	**36000**
12000	**63044**	**48000**	**48000**
15000	**78596**	**60000**	**60000**
45000	**238404**	**180000**	**180000**

When the above storage results are represented by the graph, the difference is clear. The black line represents quad tree, the red line represents Z-curves with quad tree, and the blue line represents H-curves with quad tree, as shown in the (fig 6).

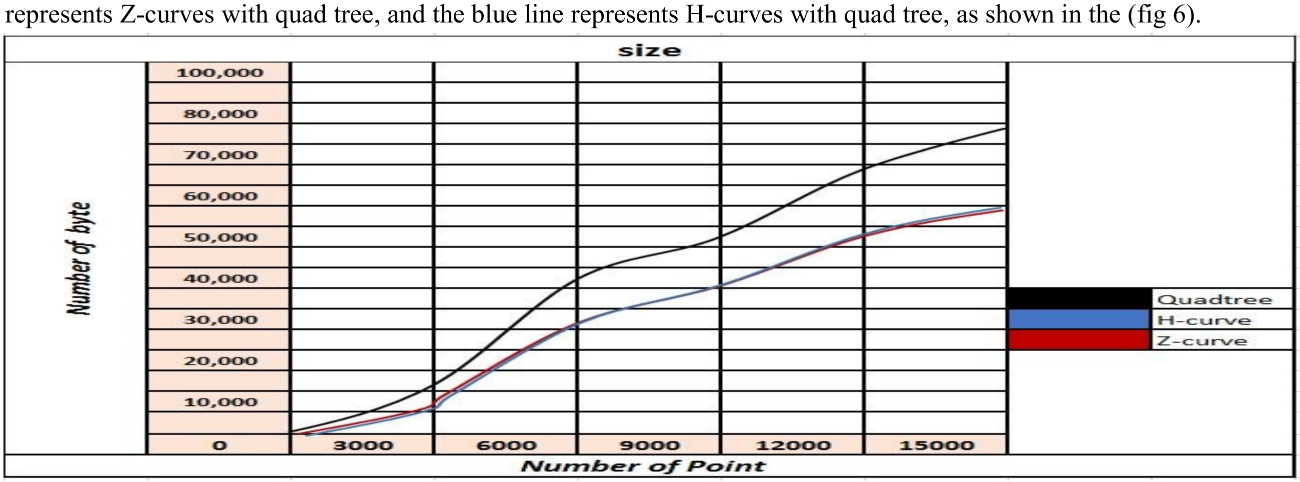

Fig (6): As the required storage for each case

In the same way, when calculating the amount of time it takes, it will not retrieve the point information which is (X: 44.42574332115971, Y: 33.30565709558853) looking for each time for a different number of points. The time taken when applying quad tree alone is much greater than applying the hybrid state of Z-curves with quad tree and H-curves with quad tree, as shown in (table 2) .

Table (2): The Response Time taken for each case, with number of k = 10.

Time			
Number of points	**Quad tree (second)**	**Z-curve with quad tree (second)**	**H-curve with quad tree (second)**
3000	**0.043**	**0.007**	**0.006**
6000	**0.103**	**0.006**	**0.006**
9000	**0.144**	**0.006**	**0.006**
12000	**0.153**	**0.007**	**0.006**
15000	**0.226**	**0.006**	**0.007**
45000	**0.669**	**0.032**	**0.031**

When the time results are represented by the graph (fig 7) , Observed that the time taken has significantly improved in the hybrid case where the quad tree.

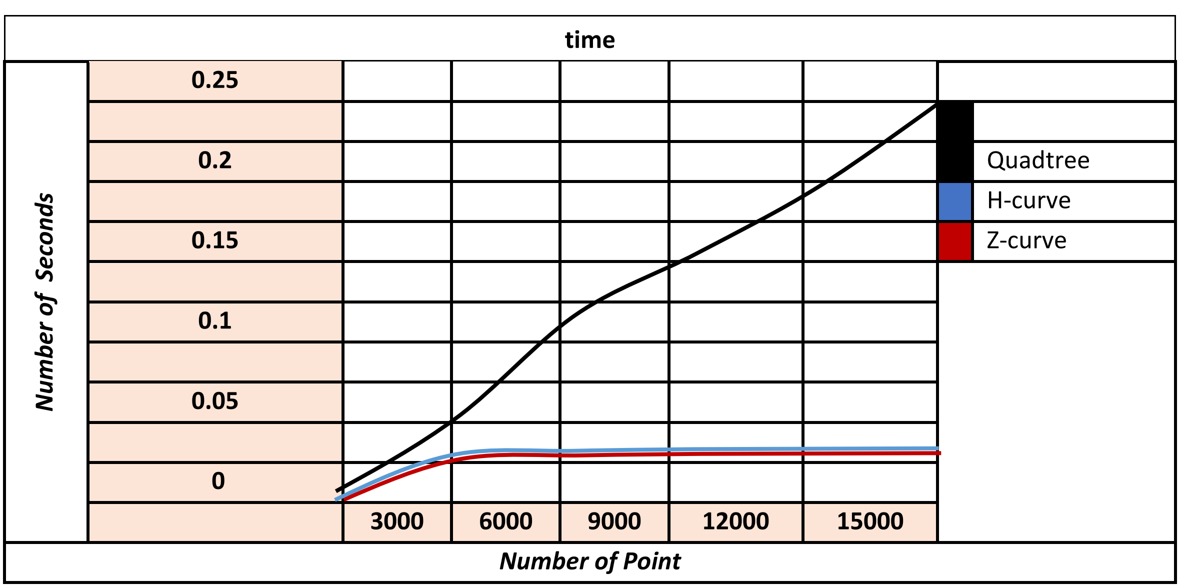

Fig. (7): Average Response Time

In previous experiments the focusing was on a certain point (constant) for a different number of points each time. Now, different points have been taken from the map and work on it and at all intersections points in the Baghdad map of 16039 points (fixed). Where x and y are the coordinates for each point that are searched each time, as shown in (table 3) of size.

Table (3): The required storage for a number of point.

size(byte)					
no.of. points in map =16039 and k=10					
	Points coordinates		Quad tree	Z-curve	H-curve
1	x	44.41766018608943	84208	64156	64156
	y	33.27467162030946			
2	x	44.441372963220765			
	y	33.28128363820795			
3	x	44.37628338581957			
	y	33.21815039705076			
4	x	44.328338898606724			
	y	33.23086648950427			
5	x	44.46671797633286			
	y	33.27449448275181			

When the results are represented by the graph below, we notice the difference and how quad tree has improved significantly.

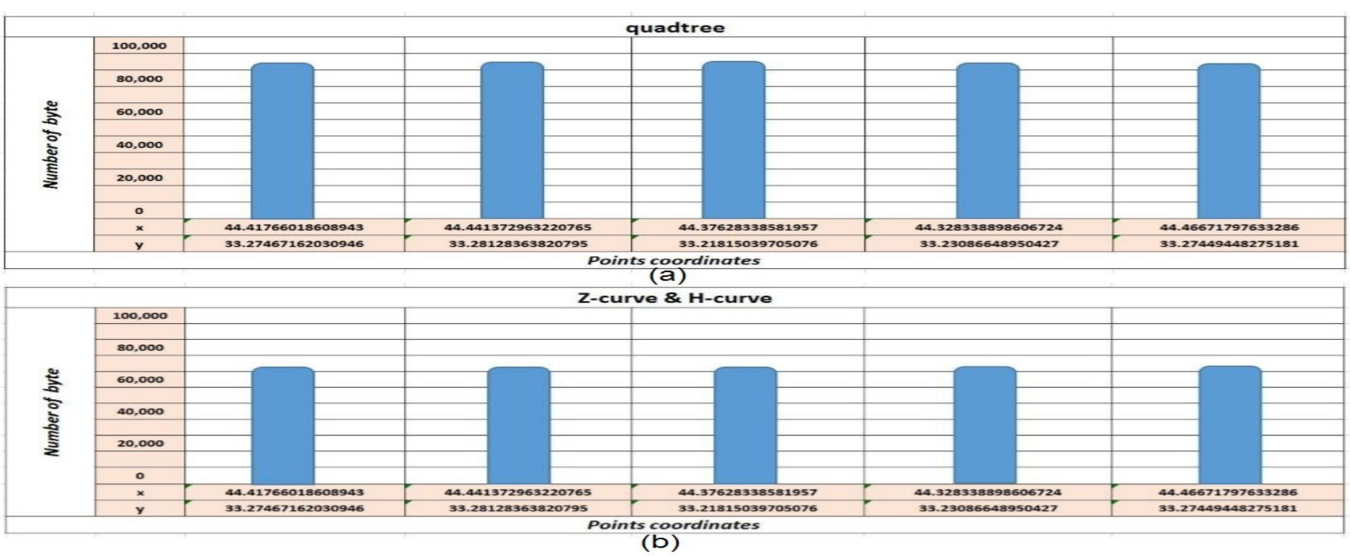

Fig. (8): Number of bytes for each points (a): quad tree, (b): Z-curve and H-curve

In the same calculation method above, the time taken to search for each of the points below was calculated on the map and for the same total number of points. A significant improvement was also found in the quad tree when combined with space filling curves.

Table (4): The Response Time for a number of point

Time (second)						
no.of. points in map=16039 , k=10						
	Points coordinates		Quad tree	Z-curve	H-curve
1	x	44.41766018608943	0.168	0.043	0.037
	y	33.27467162030946			
2	x	44.441372963220765	0.227	0.029	0.017
	y	33.28128363820795			
3	x	44.37628338581957	0.162	0.023	0.021
	y	33.21815039705076			
4	x	44.328338898606724	0.188	0.021	0.014
	y	33.23086648950427			
5	x	44.46671797633286	0.219	0.038	0.021
	y	33.27449448275181			

Each point is represented in the graph (fig 9) to show the difference between points on the one hand and between the normal quad tree and hybrid quad tree, where the improvement is very clear.

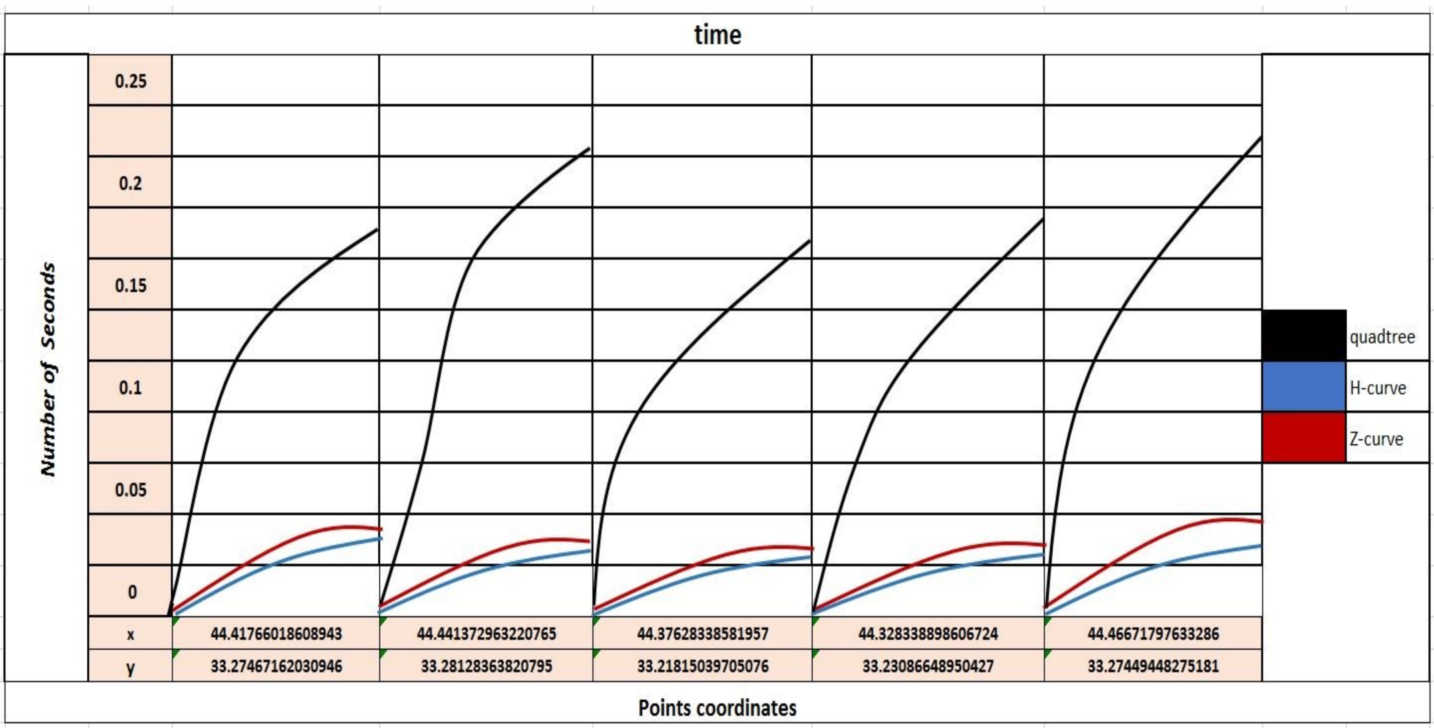

Fig. (9): the Average Response Time for each point

7. Conclusions

Most quad tree algorithm improvements depend on the speed of access and the amount of storage needed to represent and index data spatially, and most multidimensional applications are focused on sequential reading. Storage and searching for objects will be carried out with high efficiency. In this paper, the quad tree algorithm was improved through hybridization with some types of space fill curves, namely Morton (z-order) curve and Hilbert curve. They arrange the points sequentially and convert them from two dimensions to one dimension, so that repetition is eliminated in the normal search process quad tree. When a specific node in the quad tree is searched and reached to the terminal node in the tree, it returns to repeat the contract again to take another path, take a lot of time and storage, the method was applied to the intersection points in the

Baghdad map and used for q and q of normal or compressed hybrid And found a significant improvement in storage time required. Resulted Storage was reduced to approximately 25%, as well as reducing the time taken to more than 80% of the time taken by the regular quad tree.

Reference:

[1] Amitava Chakraborty, Sudip Kumar De, and Ranjan Dasgupta,'' Balancing of Quad Tree using Point Pattern Analysis", International Journal of Computer and Information Engineering
Vol:5, No:4, (2011)

[2] Ashwaq Talib Hashim1* and Zaid Mundher Radeef2,'' Correlated Block Quad-Tree Segmented and DCT based Scheme for Color Image Compression,'' Indian Journal of Science and Technology, Vol 9(26), DOI: 10.17485/ijst/v9i26/92433, July 20/(2016)

[3] Fang Hou, Chenghui Huang, Jiyuan Lu."A Multi-dimensional Data Storage Using Quad-tree and Z-Ordering". International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).

[4] Michael Bader,'' Space-Filling Curves'', ISSN 1611-0994, ISBN 978-3-642-31046-1 (eBook), ISBN 978-3-642-31045-4,Springer Heidelberg New York Dordrecht London,(2013)

[5] A. A. El-Harby & G. M. Behery,'' Novel Color Image Compression Algorithm Based on Quad tree'', 0975-4172 & Print ISSN: 0975-4350 , Year (2012)

[6] Julian Hirt, Dominik Gauggel, Jens Hensler, Michael Blaich, and Oliver Bittel," Using Quadtrees for Realtime Path_nding in Indoor Environments", University of Applied Sciences Konstanz, Germany Brauneggerstr. 55 D-78462 Konstanz (2010)

[7] Ningchuan Xiao. 'GIS Algorithms, Theory and Applications for Geographic Information Science & Technology ". ISBN 978-1-4462-7432-3, December 2016

[8] Phyo Phyo Wai, and Nyein Aye,'' GIS Data Compression Based On Block-Encoding and Quadtree Approach'', International Conference on Advances in Engineering and Technology (ICAET'2014) March 29-30, (2014)

[9] Capt. Dr. S. Santhosh 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, pg. 810-816, April- 2014

[10] G.Shravan Kumar, D.Naresh 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, pp. 642–648, July 2017.

[11] Satori Tsuzukia, Daichi Yanagisawaa, Katsuhiro Nishinaria,'' Auto-generation of a centerline graph from the geometrically complex roadmap of real-world tra_c systems using a hierarchical quadtree for cellular automata simulation'', The University of Tokyo, 4-6-1, Komaba, Meguro-ku, Tokyo 153-8904, Japan (2019).

[12] X.G. Zhou1, H.S. Wang,"a quad tree spatial index method with inclusion relation for the incremental updating of vector landcover database ". The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences``, Volume XLII-4, 2018.