

# STUDY AND ANALYSIS OF MULTI-DIMENSIONAL HILBERT SPACE FILLING CURVE AND ITS APPLICATIONS – A SURVEY

Swapna Panda<sup>1</sup>, Sasmita Mishra<sup>2</sup> and Sarojananda Mishra<sup>3</sup>

<sup>1,2,3</sup> Department CSEA, IGIT Sarang

## ***Abstract***

*A map has to be designed to show the directions and the objects present in a specific area. So, it is necessary to visit each and every points of that area. For that a Space Filling Curve can be used. SFC can visit all the points present in a multi-dimensional data base. A spatial query can select geographical features based on location or spatial relationship, and a Nearest Neighbor search can be used to find the nearest object of a query object.*

## ***Keywords***

*Space filling curves, Hilbert space filling curves, spatial query, k-nearest neighbor.*

## **1. INTRODUCTION**

The multidimensional representation is very important, because it is used to represent a object in a multidimensional space. Space filling curve Maps a multidimensional data into one dimensional data. [1] For which every element present in a specific area can be viewed in a one dimensional figure. At that situation it will be easy to recognize which element is at which position, or nearer to which element or object. A space filling curve visits each and every points present in a specific area. So that it will be easy to know that which elements are present in that area? It visits each point only once so there will be no chance of repetition. Each thing will appear only once. In 1891 a German mathematician David Hilbert described about a space filling curve. [2] After him that was named as Hilbert Space Filling Curve (HSFC). Nearest neighbor search, as the name specifies means that to search the nearest neighbor of an element. For finding out the closest point of a query point, the nearest neighbor is an optimization method. [3][4] Nearest neighbor search is also known as proximity search, similarity search or closest point search. Query points are the spatial query points. [3][4]

Spatial query points are the particular type of data-base query points. These spatial query points are supported by the geographical databases or the spatial database. This describes spatial relationships between the geometrics like points lines and polygons.

## 2. ORGANIZATION OF THE PAPER

Section-3 is all about a survey on different space filling curves. Section-4 describes about some space Filling Curves. Sub-section-4.1 is the description about Hilbert Space filling Curve that is one among the Space filling curves. Section-5 is the study on spatial queries, and its sub-points are describing about some basic proximity queries or spatial queries, those are: Nearest Neighbor, k- Nearest Neighbor , Range query, spatial join , approximate Nearest Neighbor, point query, window query. Next section Describes more about the k-Nearest Neighbor search, and its sub-section describes the random shift operation and an algorithm for finding the Nearest neighbor by using the Hilbert Space filling curve. In the next section there is an experimental graph on the effect of visualization of close points and the key points of Hilbert curve in 1D and 2D. The section-8 is about the comparison about the approximate and the exact nearest Neighbor Search. Finally the last section is the conclusion.

## 3. A SURVEY

In the year 1877, the mapping technique was developed by Georg Cantor for mapping the unit square  $[0,1]^2$  from the unit interval  $[0,1]$  . [2] After that in 1979, it was proved that the mapping technique was not continuous, which was introduced by Eugen Netto. Then in 1886 the first definition of curves was introduced by Camille Jordan. [2] and the first space filling curve was constructed by Giuseppe Peano in 1890. The graphical representation for this Peano curve was presented by David Hilbert in the Year 1890. [2] and that curve was denoted as Hilbert Curve. Later on many more Space Filling curves were introduced by the researchers such as, Lebesgue Curve was introduced in 1905 and Sierpinski curve was developed in 1912. [2]

## 4. SPACE FILLING CURVES

Space filling curve is an interesting topic for the mathematicians. In their analysis a space filling is a curve whose range contains the entire 2D unit space. In the year 1890 the first space filling curve was discovered by Giuseppe Peano in a 2-dimensional plane, which was called Peano Curve [2] . There are many types of Space Filling Curves as shown in fig:1 .

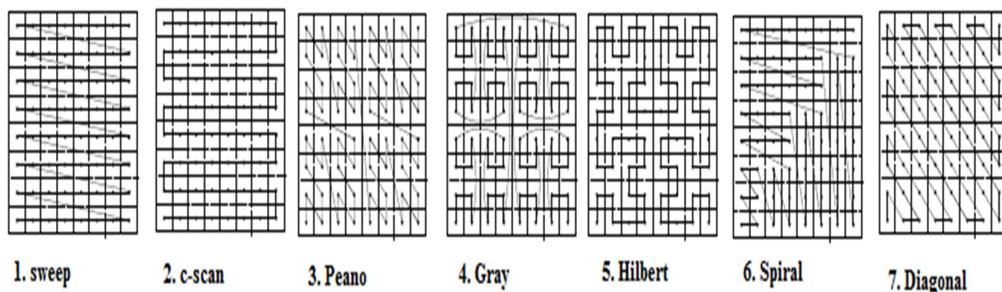


Figure 1. Different Space Filling Curves [9]

A Space filling is basically used for mapping a multi-dimensional space into one dimensional space [1] . A SFC goes through all points present in a space. It goes to every points only once[2] . So that there is no any chance of repetition. By using this the location of an object can be specified.

## 5.HILBERT SPACE FILLING CURVE

Space filling curves were the topic of interest for the leading mathematician of 19<sup>th</sup> century [5]. The graphical representation for this space filling curve was given by David Hilbert in the year 1891. [5][6]

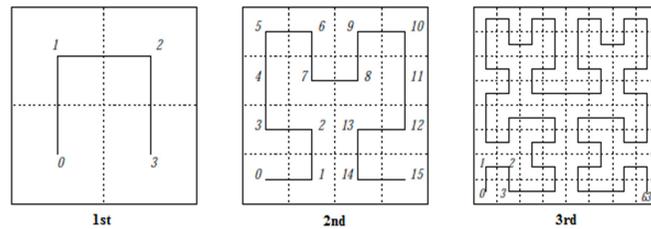


Figure 2. Hilbert Space Filling curves [7][8]

Fig2 shows the three steps of an infinite process in case of 2D. In the diagram 1<sup>st</sup> a square is divided into four quadrants. This is called a first order curve which is drawn through their center points . The quadrants are ordered in such a way that two adjacent will share a common edge. In the diagram 2<sup>nd</sup> each quadrants of the diagram 1<sup>st</sup> are further divided into four quadrants. This is called as the second order curve. Here also the center of adjacent quadrants are joined. Then the diagram 3<sup>rd</sup> represents the third order curve.

The Hilbert curve is a fractal space filling curve and is known to keep the original 2D relationship to a certain degree. It creates a bidirectional 1D-2D mapping [10][15]. The Hilbert curve can be computed (recursively) from the previous one by replacing each corner by a rotated and reflected version of the basic U-shaped pattern. [10]. An important property of Hilbert Space Filling Curve is that consecutively ordered points are adjacent in Space [10].

## 6. SPATIAL QUERY

Spatial query often called proximity problem[11]. Some basic proximity queries are :

### 6.1. Nearest Neighbor

Retrieve the closest object to q in X.  $\{p \in X / \forall v \in X, d(q, p) \leq d(q, v)\}$  . [11]

## 6.2. k-Nearest Neighbor Search

Retrieve  $k$  closest object to  $q$  in  $X$ , which returns a set. Here  $k$  is not the number of dimensions. [11]

## 6.3. Range query

Retrieve all objects that are within distance  $r$  to  $q$ .  $\{p \in X / d(q, p) \leq r\}$ . Range queries effectively define a spherical region. Here, the term sphere could refer to a circle, sphere or hyper- sphere, depending on the number of dimensions. A hyper-sphere is a sphere with more than three dimensions. [11]

## 6.4. Spatial Join Query

Given two sets of points, retrieve all pairs of points, one from each set, such that the distance between the points is less than or equal to  $r$ .  $\{\forall a \in A, \forall b \in B, d(a, b) \leq r\}$ . [11] For example, find libraries within 10 kilo-meters of schools. The special case where both datasets are identical can be termed self-spatial join [12], for instance, find all schools within 5 kilo-meters of another school. Spatial join is also known as  $\epsilon$ -join in some papers, and is often used in database application.

## 6.5. Approximate Nearest Neighbor Query

Retrieve all objects within distance  $1+\epsilon$  of the true nearest neighbor, where  $\epsilon \geq 0$ . Notice that this requires the execution of the nearest neighbor query and then a range query. [11]

## 6.6. Point Query

A point is specified and any points with identical coordinates are retrieved. This is equivalent to a range query with  $r=0$ . Also known as exact query. [11]

## 6.7. Window Query

Retrieve all objects within a specified rectangular region in data space. A hyper-rectangle is a rectangular prism with more than three dimensions and can be defined by two bounds on each coordinate, and is therefore always parallel to the axis window. [11] In this paper rectangle is used to refer to a rectangle, rectangular prism, or hyper-rectangle, depending on the number of dimensions. In vector space, window queries are typically quicker than range queries, so it often makes sense to execute a window query which forms a minimum bounding rectangle around a range query sphere, and then examine all returned elements to check which fall inside the range query. [11]

**Figure 3** illustrates most of the simple queries. In this paper k-Nearest Neighbor will be the focus. In which, from a set of objects  $X$ , and how find  $k$  closest object to  $q$  ( $q$  is the query object).

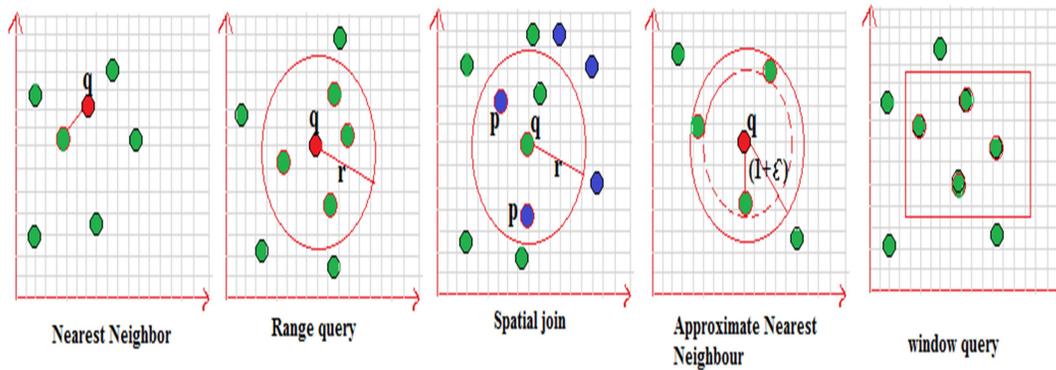


Figure 3. Proximity problems [11]

## 7. K-NEAREST NEIGHBOR SEARCH

The nearest neighbor search continues to prove itself as important among many field of mathematics, science and engineering. The Nearest Neighbor problem multi-Dimensions is stated as follows : Given a set of  $n$  points and a novel query point  $q$  in a  $D$ -dimensional space and a point has to be search, such that its distance from  $q$  is lesser then or equal to the distance of  $q$  from any other point in the set [13].

The distance between the query point and another object can be found by the Euclidian Distance formula. This is the most commonly used formula for finding the distance between two spatial objects. It is denoted as  $E^n$ . Euclidian distance can be thought of as a “real world” distance between two points. Formula for calculating the distance between two object  $P(P_x, P_y, P_z)$  and  $Q(Q_x, Q_y, Q_z)$  is :

$$E^n\{(P_x, P_y, P_z), (Q_x, Q_y, Q_z)\} = \text{sqrt}\{(P_x - Q_x)^2 + (P_y - Q_y)^2 + (P_z - Q_z)^2\} \quad [11]$$

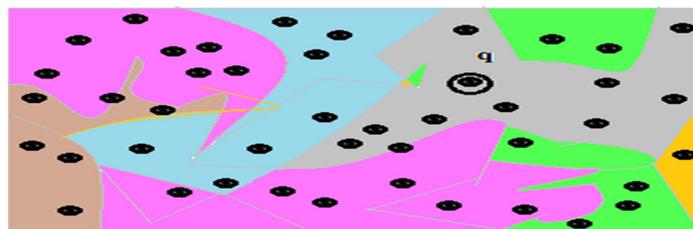


Figure 4. Nearest Neighbors [13]

The model for  $k$ NN is the entire training dataset. [11] When a prediction is required for a unseen data instance, the  $k$ NN will search through the training dataset for the  $k$ -most similar instances. [11] The prediction attributes of most similar instances is summarized and returned as the prediction for the unseen instance. [11] The similarity measure is dependent on the type of data. For real-valued data, the Euclidean distance can be used. Other types of data such as categorical or binary data, Hamming distance can be used. [11]

Nearest neighbor searches in the image plan are among the most frequent problems in a variety of computer vision and image processing tasks [10]. They can be used to replace missing values in image filtering, or to group closest objects in image segmentation, or to access neighboring points of interest in feature extension[10].

## 7.1 The k-NN Join Process by Random Shift

### 7.1.1 Random Shift

To map a point from multi dimension to one dimension, the Hilbert space filling curve can be used. The Hilbert curve can usually conserve the spatial locality. This is actually always not possible; sometimes it may happen that object that is close in a multi-dimensional space may not be close along the Hilbert curve. For the input dataset, we have to generate some number of independent vectors, which must be belongs to  $[2, \alpha]$  and should also belongs to the input dataset. In the random shift operation data points in the input dataset by a random vector, and this operation is simply the combination of dataset and the independent vector. Then the input data set achieves the total no of independent vector. Randomly shifted copies are the data set. For the randomly shifted version of data set, the search process can be repeated. In this way a better result can be obtained. [14]

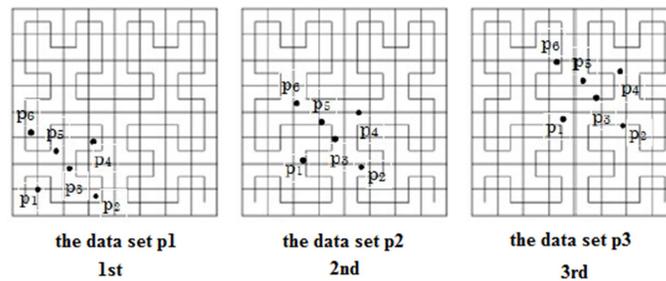


Figure 5. The Data shift Effect [14]

The above figure 5 shows the data shift effect. In the data set there points  $\{ p_1, p_2, p_3, p_4, p_5, p_6 \}$  on the Hilbert curve of 1<sup>st</sup> figure. Here  $p_5$  is the true nearest neighbor of  $p_3$ . But in the 1<sup>st</sup> figure both predecessor  $p_2$  and successor  $p_4$  of  $p_3$  are closer to  $p_3$  in the linear Hilbert order. So if someone will check the immediate neighbor of  $p_3$  then that person cannot find the true nearest neighbor. This above example tells that, this curve recursively divides the space into compartments each of which is thoroughly traversed before proceeding to the next. This is intrinsic to other space filling curves and is inferior only in high dimension. After this we have to shift the data set and the query points by random shift, through which we can get the data set  $p_2$  and the Hilbert order of the  $\{ p_1, p_3, p_5, p_6, p_4, p_2 \}$  in figure 2<sup>nd</sup>. In this situation the true nearest neighbor are now adjacent in Hilbert order. Like this the Hilbert order for  $p_3$  can also be obtained and the new Hilbert order will be  $\{ p_1, p_6, p_3, p_4, p_5, p_2 \}$  in figure 3<sup>rd</sup>. by looking at the predecessors and successors of the query we can find the nearest neighbor by shifting the data

several time. By examining more than one predecessors and successors only improves the quality of approximation [14].

## 7.2 An Algorithm for Finding Nearest Neighbor by Using Hilbert Space Filling Curve

This algorithm is about finding the nearest neighbor of the query object by using the nearest neighbor search [10].

1. Computation of mapping Between 1D and 2D coordinates
2. Store all key points in an array at their Hilbert Index position
3. Find the left and right neighbor of each array entry and assign the nearest neighbor in a two passes walkthrough.

For the k-NN problem the set of key points must be stored in a priority queue [10]. The neighbor of the query point can then be found by using the successor function of the queue [10].

### EXPERIMENT

Diagrams on the relationship between corresponding 1D and 2D distances for randomly selected pairs of close points once for the 1D Hilbert curve and then for the 2D image space visualize : [10]

- Close points in 1D are close in 2D as well
- Close 2D points are not necessarily close in 1D
- On average 2D distance is the square root of 1D-distance
- Even in exact approximated results are still good due to the good 1D to 2D correspondence.

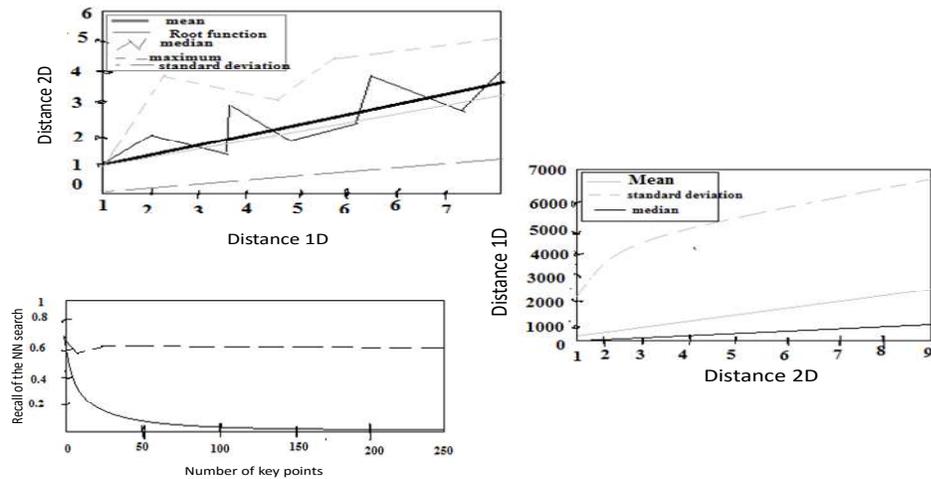


Figure 5. Visualization of close points and key points of Hilbert curve in 1D and 2D [10]

comparing the approximated results to the exact result for a varying number of key points show a recall of 50%. [10]

## APPROXIMATE VS. EXACT NEAREST NEIGHBOR SEARCH

When 'C' is the number of Hilbert indices and 'n' is the number of key points, then : [10]

- Pre-computation time  $O(C)^*$
- All Nearest Neighbor Approximation  $(C+n)^*$
- K-Nearest Neighbor approximation  $O(n \lg \lg C)^*$
- 50% of the query are exact result
- Real time capable for high resolution image / video

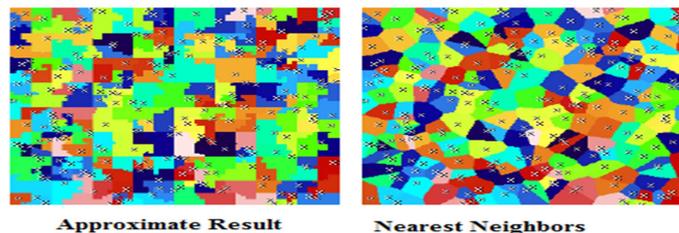


Figure . Approximate and Exact Nearest Neighbors [10]

All Nearest Neighbor assignment using the proposed approximate method ( left image ) and an exact search (right image ) for 240 key points (marked by crosses ). The resulting cells are colored randomly but consistent over both diagrams [10].

## 8. CONCLUSION

Hilbert space filling curve is used to locate the spatial points in a multi-dimensional database. Similar types of spatial query are represented through this Hilbert Curve and this curve is used to find out the nearest neighbor of a query point in 1D and 2D. Comparing the representation of spatial query in Hilbert Curve, the total numbers of key points are calculated and the close points in 1D which are close in 2D are shown; but the close points in 2D are not necessarily close in 1D. Again by comparing the approximate nearest neighbor search and exact nearest neighbor search, it is concluded that the exact nearest neighbor is more appropriate as more accurate result is generated than in approximate nearest neighbor search.

## 9. REFERENCES

- [1] Mokbel M. F., Aref W. G., 2001. Irregularity in Space Filling Curves with application in Multimedia Database. Department of Computer Science, Purdue University, West Lafayette 47907-1398, *ACM*. (2001).
- [2] Bader M., 2008. Space Filling Curve and Their Applications in Scientific Computing, Technische University, Munchen, *ACOPhys- State University of St. Petersburg* ., (1-5. Sep. 2008)
- [3] Shamos M. I., Hoey D., 1975 Closest point problem, In proc. 16<sup>th</sup> annual *IEEE symposium on Foundation of computer science (FOCS)*., 151-162., DOI: 10.1109/SFCS.1975.8., (1975).
- [4] Boyd, Lieven, Stephen, Vandenberghe., 2004. Convex Optimization ., *Cambridge University Press* .p.129., ISBN 978-0-521-83378-3., (2004).
- [5] Lawder J.K., King P. J. H. 2001. , Birkbeck College, University Of London. , 2001

- [6] Bongki Moon, H.V. Jagadish, Christos Faloutsos , Joel H. Saltz, 2001 Analysis of the clustering properties of the Hilbert Space Filling Curve. , *IEEE , trans.* Vol-13. , No-1. , (jan/feb 2001).
- [7] J. K. Lawer , P. N. J. King Using Space Filling Curve for multi dimensional indexing., Birkbeck College, University of London., WC1E 7HX
- [8] Valgaerts L., Space Filling Curve An Introduction., 2005., *paper accompanying the presentation for joint advance student school in st. peterberg.*, ( 2<sup>nd</sup> April 2005)
- [9] Mokbel M. F., Aref W. G., Kamel I., Department of Computer Science, Purdue University, West Lafayette., 47907-1398 .
- [10] Martin Stommel, Stefan Edelkamp, Thimo Wiedemeyer and Michael Beetz., Fractal Approximate Nearest Neighbor search in Log-Log., Institute of Artificial Intelligence., *University of Bremen AM Fallturn 1.*, 28359 Bremen, Germany.
- [11] Andrew Noske., Dynamic Range Queries in Vector Space., *IT Department, James University, Cairns Campus.*
- [12] C. Faloutsos, B. Seeger, A. Traina and C. Traina, 2000 Spatial join selectively using power laws, ACM SIGMOID Record , Proceeding of the 2000 ACM SIGMOID international conference on Management of data 29(2), p.177-188 (may 2000).
- [13] Jason Brownlee., Tutorial to Implement K-Nearest Neighbors in Python from Scratch, (12 sept. 2014).
- [14] Qinsheng Du., Xiongfie Li., Xiaoli Zhang., Efficient K-Nearest Neighbor Join Algorithm with the Hilbert curve., *Journal of Computational Information System .*, Department of computer science and Technology ., Changchun University, Changchun 130022, China.
- [15] S.N.Mishra, et al., L-System Fractals, MATHEMATICS IN SCIENCE AND ENGINEERING Volume 209 Series, Elsevier, 2007.
- [16] 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.

## Authors

Ms. Swapna panda is currently pursuing her MTech Degree at Indira Gandhi Institute of Technology, Sarang Dhenkanal, Odisha . She is doing her MTech dissertation under the able guidance of Prof. Sasmita Mishra. Her research Area is study and analysis of multi-dimensional Hilbert Space Filling Curves and its applications.



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.



Dr. Sarojananda Mishra is currently working as Professor & Head of the department of Computer Science, Engg. & Applications at Indira Gandhi Institute of Technology (IGIT), Sarang, Dhenkanal, Odisha, India. He has published more than 75 papers in International Journals and National Journals of repute. His research area focuses on fractal graphics, fractal geometry and internet data analysis. He has more than 25 years of teaching and research experiences. Many students obtained PhD degree and are continuing their PhD and MTech research work under his guidance.

