



## DTNH Indexing Method: Past Present and Future Data Prediction for Spatio-Temporal Data

John Ayeelyan<sup>1\*</sup>, Sugumarn Muthukumarasamy<sup>2</sup>, Rengan Sivagurunathan Rajesh<sup>2</sup>

<sup>1 2\*</sup>Department of Computer Science and Engineering,  
Manonmanium Sundaranar University, Thirunelveli, Tamilnadu, 627012, India.

<sup>2</sup>Department of Computer Science and Engineering,  
Pondicherry Engineering College, Puducherry, 605014, India

\* Corresponding author's Email: johnmtech@gmail.com

---

**Abstract:** Indexing methods are developed to effectively process user queries in many real-time and moving object management applications. The existing spatial data updating indexing methods are based on the Integrated binary Tree, R-Tree, R\*-Tree, Oct-Tree, Quad-Tree, Grid-Tree and Hex-Tree. The depth of these trees is unbalanced and overlapping, hence the performance is reduced in the multi-structure indexing methods. D-Tree (Decompose - Tree) based multi-structure spatio-temporal index method is proposed to find the present, past and future data. The new multi-structural model called DTNH-Tree used to find the present, past and future data. It consists of D-Tree, TB\*-Tree, NT-Tree and hash table. The D-Tree indexing is used to get the spatial data and manage the moving objects in the road network. A set of TB\*-Tree is used to index the history of moving object on road networks. A set of NT -Trees is used to manage the current position of the recently updated data and find present data of the moving objects. NT-Tree indexes the present and future information of the moving objects. Finally the set of hash tables is used for updating the data continuously. The proposed multi-structure indexing method supports different types of query processing compared to the existing indexing methods. Experimental results exhibits better updation and query performance compared to the MSMON-Tree and PPF\*-Tree.

**Keywords:** Spatio-temporal data, Indexing methods, Query processing.

---

### 1. Introduction

Nowadays query processing and indexing methods have an essential one for data retrieval in spatial - temporal network. Query processing is a collection of data arranged for ease and speed of search and retrieval of data. Indexing is a key data structure used to retrieve data from database tables and improve the speed of data retrieval in movement database. The indexing methods and query processing is inter-related one, based on the indexing method different types of query processing types are supported to the particular indexing method. Nowadays developing efficient indexing and query processing methods is an important research in Moving object database. The traditional moving objects database (MOD) is not well equipped to

handle and predict moving objects database system. The past, present and future data Predictions in moving objects data is complex one in spatial (space) - temporal (time) environments because it changes the space and time in each and every seconds. The numerous research effect taken to solve the issues of continuous moving objects databases in indexing and query processing. The spatio-temporal indexing methods are categorized into three types [1, 2] first method deal about spatial index access methods , the second method is spatio-temporal index access methods , the third is indexing the temporal dimensions, while in this method spatial data is considered as the second priority. The third method is used to predict all the point of the data and the data prediction based on the temporal and based on spatial is very complex because this type of indexing are multi-structural and multi-model.it consists of more

than two indexing methods is used to processing all the point of the data. The some of the main spatial indexing are RT-TREE [3], 3D-R-Tree [4], STR-Tree [5], MTSB-Tree [6], FNR [7], MON [8]. The spatio-temporal index methods MR-Tree [3], HR-Tree[9], HR+-Tree[10], MV3R-Tree[11]. The Temporal dimension with spatial priority indexing methods are TB-Tree[12], SETI [13], SEB-Tree [14], PMR-Quad tree[15], TPR-Tree [16], TPR\*-Tree [17], LU Grid[18], RUM-Tree [19], IMORS [20] and present research direction indexing methods  $R^{ppf}$ -Tree [21], PCFI+-Index [22],  $BB^x$ -Index [23], UTR-Tree [24], STCB+-Tree [25], PPF-Index [26]. The temporal based spatial indexing is used index all the points of indexing methods that is index the past, present and future data prediction of indexing method. The moving objects can be grouped into three movement scenarios [27], namely constrained, unconstrained, and movement in transportation networks. The movement scenarios are considered as different space than compared to first two scenarios. This paper is based on movement in transportation networks and third method in spatio-temporal indexing method.

The Most of the spatial data updating indexing methods [28] are based on the integrated binary tree, R-tree, R\*-Tree, Oct tree, Quad tree, Grid tree and Hex tree. Those derived from R-tree employ a minimum bounding rectangle (MBR) or a time-parameterized minimum-bounding rectangle (TPMBR) to represent a moving object. The range of the bounding rectangle is always larger than the range of the moving object itself. This is the fundamental reason why they cannot resolve the overlapping problem in high dimensions. The depth of these trees is unbalanced and overlapping, hence the performance is reduced in the multi-structure indexing methods. To Reduce these problem new indexing method is D-Tree (Decompose-Tree) based multi-structure spatio-temporal index method is proposed to find the present, past and future data. The proposed multi-structure indexing method supports different types of query processing compared to the existing indexing methods. Experimental results exhibits better updation and query performance compared to the  $R^{PPF}$ -Tree, STCB+-Tree, MSMON-Tree and PPF\* -Tree.

The organization of this paper is as follows. Section 2 is related works and motivation and section 3 describes data model and data structure of the new hybrid index method. Section 4 reports the performance evaluation, and conclusion is given in section 5.

## 2. Related work

In this section, gives the brief literature about various spatio-temporal access methods for past, present and future indexing methods. These indexing methods come under the categories of spatio-temporal access method or indexing at all the point of time. Past-Current-Future+-Index (PCFI+-Index) [22] indexes the past, current and future information of the dynamic objects and the finds an object's position at every point of time. The PCFI+-Index constructed based on the PCFI-Index, which was based on SETI-tree and TPR\*-tree. The PCFI+-Index having memory parts with the name frontline, and disk based part. The spatial access is indexed using region partitioned into the non-overlapping cell. TPR\*-tree index the moving belong the on overlapping cells. The hash table is indexed large updating and find the current position of the moving objects. The PCFI+-Index support most of the queries efficiently and provide only uniform solutions for the trajectory query.  $BB^x$  Index Method [23] used to find queries about the past, present and future position of moving objects.  $BB^x$  index structure is used to find the position of moving objects in linear function of time.  $BB^x$  index inherits the ability of finding present and future position of the  $BB^x$ -Tree. Three factors are considered to the stage efficiency are Object position is represented as a linear function. It is done by TPR-tree family (spatial representation), Object is based on indexing  $BB^x$ , B+-Tree used for updation.  $R^{PPF}$ -Tree is used to [21] obtain all the online position information from the moving e-service users. The proposed work offers an  $R^{PPF}$ -Tree technique capable of capturing moving objects at all positions in time. This tree supports only valid transaction time to monitor the applications and it supports time related query and position related query. VPMR-Tree [27] indexing is used to provide the solution to approximate future information for moving objects and indexing the road network moving objects. This indexing method is a variation of PMR Quad tree and real time data prediction purpose dynamic array is used. Past, Present and Future Index method (PPFI) [26] is used to predict the current and near future position of moving objects on a fixed road network with an efficient update mechanism. The proposed work is PPF-Index and which consists of a  $2DR^*$ -Tree,  $1DR^*$ -Tree and hash table for updating.  $2DR^*$ -Tree is a static part built on managing the fixed network.  $1DR^*$ -Tree is a dynamic part built on indexing object movements. This indexing method is used R Tree and R\* Tree so data overlapping problems occur. Final Future Trajectory Prediction Algorithm (FFTPA) [29] method is based on the R-Tree and its support

related to past position and trajectories. It has a dynamic road network and efficient update mechanism.

The FT-Tree Index method is used to predict past, present and future prediction of the moving objects [30]. The moving object index for full temporal query, proposes a moving object index structure FT Tree supports for temporal query, it is based on TPR-Tree. In this FT-Tree is used TPR so it is based on the R-Tree and R Tree based indexing data have overlapping problems are occurring. *HTPR\*-Tree* Indexing [31] is the foundation for the past, present and future and used to predictive queries, but also partial history ones involved from the most recent update. This index structure, named History Time-Parameterized R-tree (*HTPR\*-tree*), which takes into account moving object creation time or update time in the leaf node entry, and supports partial history query. This indexing method also based on the R-Tree and R\*-Tree. *MSMON -Tree* [32] is used to retrieve the past, present and future position of network constrained moving objects. *FNR & MON* tree only deal with history of data and cannot support the query of moving objects in the current and future positions. The structure of the hybrid tree is support multi-Grid-R\*-Tree used to index the road network. R\*-Tree is used in a route in the road network and also find trajectories of the moving objects in the road. *PPFN\*-Tree* [33] Indexing of moving objects grouped into two categories are deal with past information retrieval and future data prediction. It is hybrid structure which consists of *2DR\** managing road networks and set of *TB\*-tree* indexing the object movement history trajectories along the polyline. *HTPR\** indexing the position of the moving object after recent updates. Update improvement purpose hash table is used with *2DR\*-Tree*. The previous indexing methods of spatial data updating indexing methods are based on the integrated binary Tree, R-Tree, R\*-Tree, Oct-Tree, Quad-Tree, Grid-Tree and Hex-Tree. The depth of these trees is unbalanced and overlapping, hence the performance is reduced in the multi-structure indexing methods. The various types of indexing and the query process sing are mentioned in [38] and further consideration in spatio-temporal is security mentioned in [34-37] these papers are mentioned various advanced security features. The *PPFI* [26], *MSMON* [32], *PPFN\**[33] and *D-Tree* [28] are the main past, present and future data prediction indexing methods and these methods are considered as comparison purpose.

### 3. Proposed Method

#### 3.1 Problem analysis and data model

The past, present and future indexing hybrid methods are based on the integrated binary tree, R-tree, R\*-Tree, Oct tree, Quad tree, Grid tree and Hex tree. The main problem is having dead spaces and overlapping problems are occurring in the moving objects updation. The data overlapping produce massive data updation, so the data retrieval time is more and it produces worse performance. The *MSMON- Tree* [32] addressed this problem, in this model also used R\* -Tree and TPR-Tree is stored spatial data information. The structure of the R-Tree and R\*- Tree shown in the Figs. 1 and 2.

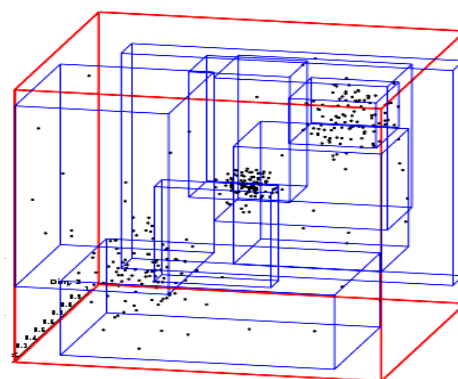


Figure.1 R-Tree

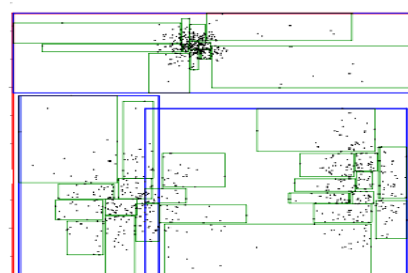


Figure.2 R\*-Tree

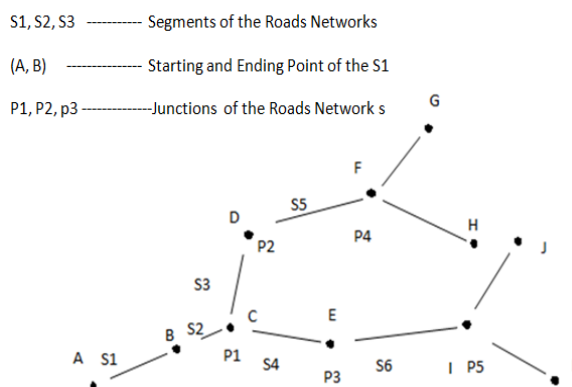


Figure.3 Road Segment Representation

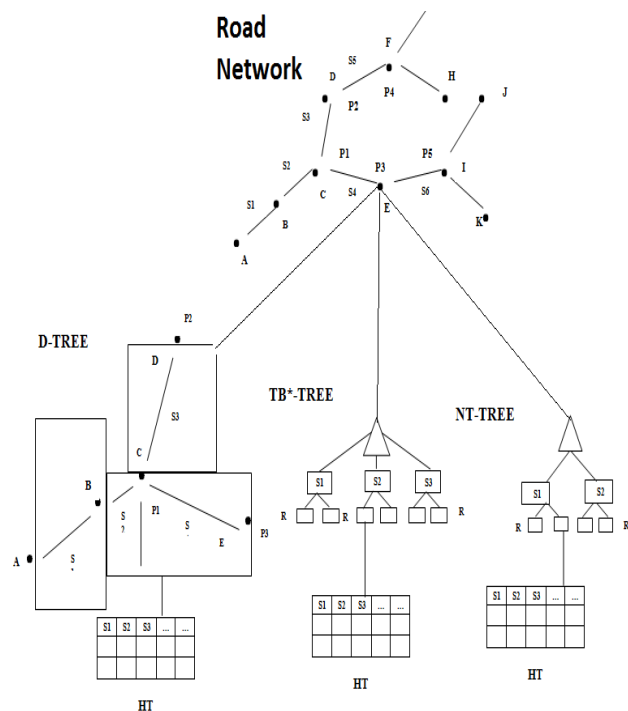


Figure.4 Proposed index method

In the moving object data representation, network model is necessarily one. The proposed network model represented as in term of routes, junctions and roads (i.e) network  $G = (P, R)$ , where  $P = \{P_1, \dots, P_n\}$  is a set of junction,  $R$  is the route and  $A, B, C, D$  is the points and  $S_1, S_2, S_3$  is the segment between two points. The  $A$  and  $B$  is the starting and ending point between the junctions. The sequence of road segment in the continuous network has shown in the Figure 3.

### 3.2 Structure

The proposed multi- model structure is called DTNH –Tree shown in the figure 4. This is a multi model structure consists of D-Tree (D-Compose Tree), TB\*-Tree (Trajectories Bundle Tree), NT-Tree (Networks -Limited TPR-Tree), and hash table. D-Tree is used to answer the spatial information such as geometric information (i.e) Road information, data transformation, actual time and spatial location of moving objects noted directly. In D-Tree spatial information noted directly so time complexity is reduced, so performance automatically increased. TB\*-Tree used to find trajectories of moving objects and the NT-Tree is used to find the present and future position of moving objects. Hash Tables are used for updation purpose. The Hash tables are connected each tree node so it is easily updates the data.

The proposed DTNH-Tree indexing has three parts. The first part is Decompose Tree (D-Tree). The D-Tree is a space partitioning method for accessing moving object data. In this part all the spatial points

are noted with time attributes. The main attributes are  $[ID, T, S, A]$ , where  $ID$  is a unique object identifier,  $T$  is a continuous time stamp,  $S$  is a spatial Co-ordinates,  $A$  is a time attributes. Using this D-Tree recursively decompose spatial data with time attributes. So easily get all the records in the, spatial coordinate wise and time wise moving object data is update in the table. This D-Tree is based on partitioning, so data overlapping is avoided, easily access the data and time complexity is reduced. Finally The D-Tree is linked to a hash table for continuous updations. The Second part is TB\*-tree indexing having moving object history trajectory along with position of moving objects. This TB\*-Tree is linked with hash table and D-Tree. TB\*-Tree manage both spatial and Temporal information’s. The third part is NT-Tree is designed and manages moving object information. Queries about future position information are retrieved from NT-Tree. This tree also updated with hash table for current updation.

### 3.3 Insertion

Insertions of the moving objects are happening with the help of Hash Tables (HT).  $P, R, S$  and  $O$  information are located and then inserted to hash table structure. The spatial information and temporal data  $[ID, T, S, A]$  are inserted into the leaf node of the hash table structure. The coming objected are inserted into TB\*-Tree with the help of hash table structure. The present moving objects are moving from a hash table to NT-Tree. All this hash table and NT-Tree and TB\*-Tree are implemented with the help of the Pointer. The after insertion of hash table from the NT-Tree and TB\*-Tree data are moved to D-Compose Tree. With the help of D-Tree data overlapping are avoided.

#### ALGORITHM 1. Insert $[ID, T, S, A]$

*/\* input:  $P, R, S$  and  $O$  information inserted into the hash table and  $S_1$  and  $S_1$  starting and ending point of Road segments \*/*

1. Get Route ( $R$ ) information and moving Object ( $O$ ) are inserted
  2. Invoke hash table
  3. If (hash table, Object > 0) objects are inserted
  4. Else Hash table is empty
  5. Check time duration then
  6. If any objects are find then insert
  7. Otherwise
- END Insert.

### 3.4 Updation

The moving objects, updating are updated with the help of hash table, D-Tree and TB\*. The hash table is used for dynamic updation and objects are moving next indexing and the hash table is resize the objects when updations are change. D-Tree is used to update the road information. TB\*-Tree update the Road trajectories and moving update moving objects updations in hash tables. First moving objects are updated in a hash table and then spatial data are moved to D-Tree and TB\*-Tree. The present data updated in NT Tree and here all the updations are updated directly from hash table to D-Tree and TB\* and NT-Tree so performance is increased because time complexity is reduced. The main updation parameters are Moving Objects (*MO*), Moving Object Identification (*MO.id*) Object Velocity (*V*), Time (*T*), Segment (*S*), Position (*P*). Three Situation is considered for updation of temporal data. First Velocity (*V*) is considered for updation. Second Time threshold (*T*) is considered and Third is moving objects are moving in opposite direction is also considered for updation. The main spatial updation is new place or position find, new entry (*N*) is considered and this new entry updated in D-Tree. The updation algorithm is as follows.

Algorithm 2: Updation

Input: /\*moving object information (*MO*, *MO.id*), Velocity (*V*), Time (*T*) and spatial information (*P*, *S*, *N*)\*/  
 1. If hash table (*Mo*, *MO.id*)!=0;

2. *MO* position (*P*) is moved to D-Tree
3. Update hash table node;
4. If Objects opposite directions
5. Delete old node trajectory and new Entry (*N*)

into TB\*-Tree

6. Hash table node is moved into NT-Tree
7. Else delete from hash table
8. Search and insert new into TB\*-Tree and NT-Tree

9. Hash table (*Mo*, *MO.id*) point to store moving object information  
 End Update.

### 3.5 Search procedure

DTNH -Tree indexing support to different types of search procedure compared to other indexing method because D-Tree is support to most of the all spatial queries and remain two trees are supported for temporal queries. And it also supports to spatio-temporal queries also. The main Query types are point query, Time slice query, spatial slice query, window query, moving query, trajectory query,

Topological query and predictive query. TB\* is supports both spatial- temporal queries and NT-Tree supports temporal queries.

#### Point Query

Point query is used to find exact information, in particular spatial portion *P* at a time interval *T*. This query retrieves all the nearest spatial information. The representation of point query is  $Q = (P, T)$ .

#### Time Slice Query

The time slice query retrieves spatial data in term of time slice *T* and spatial rectangle *R* at the time instance. B-Tree, R-Tree and R\*-Tree are also used to find spatial information, but the main drawback is overlapping of rectangle and data. The proposed work has rectified this problem because D-Compose tree rectified this problem.

#### Spatial Slice Query

Spatial Position (*P*) information's are retrieved particular time slice interval [*Ts*, *Te*]. The nearest information is retrieved in particular time. The representation of spatial Slice query  $Q = (P, Ts, Te)$ .

#### Window Query

Multi-dimensional spatial rectangle [*R*] area data is valid during a particular time interval [*Ts*, *Te*] with the help of D-Tree is overlapping is avoided. Query representation is  $Q = (X_1, Y_1, X_2, Y_2, Ts, Te)$ .

#### Temporal Query

This query is used to find time related information which is retrieved with time interval. For example, "what is the object position of 10.30 am?" This query information is retrieved with the help of velocity and time. The Query representation is  $Q = V*T$ .

#### Spatio-Temporal Range Query

This query is used to find all the objects have with certain Range  $R = (X_1, Y_1, X_2, Y_2)$  during time interval  $T = (Ts, Te)$ .

#### Topological Query

This query is used to find all the objects are in enter, leave, cross a given position  $R = (X_1, Y_1, X_2, Y_2)$  with time interval  $T = (Ts, Te)$ .

### Continuous updation Query

Continuous updation query is used to update all the information. The updation is based on velocity ( $V$ ), Time ( $T$ ) and position ( $P$ ). The updation query is all the events are updated with instance of time. The query representation is  $Q = (V, T, \text{ and } P)$

### Predictive Query

DTNH indexing method supports to predictive query. The predictive query predicts the data based on recent update time Interval ( $T_1$ ), update position ( $P_1$ ) and velocity ( $V$ ). Based on this currently what information available and next time interval ( $T_2$ ), it predicts further position ( $P_2$ ).

## 4. Experimental setting and details

The evaluation performance of the DTNH -Tree is compared with MS-MON-Tree [32] and PPFN\*-Tree [33]. The Experimental purpose randomly generated, distributed network based moving object are used. On real implementation is based on real road network data is used as input. The position of the object and the velocity of the objects are considered for experimental and data prediction purpose. The position in X-Y coordination and moving objects are considered in the form of (Object id, Position ( $P$ )), where an object id denotes where the objects are available and relative position of the objects are denoted. After identifying the object position and object attributes, the number of segments also noted.

### 4.1 Performance analysis

#### Update Cost Comparison

The figure 5 shown the compare with the update cost of DTNH-Tree with MSMON-Tree and PPFN\*-Tree. The CPU time is Increase when the increase the number of updates. The main advantages of the updation are node access performance is increased because the node overlapping is avoided with the help of D-Tree. The comparison purpose data are generated from 0 to 1000 and updation is checked at different times and different query ranges.

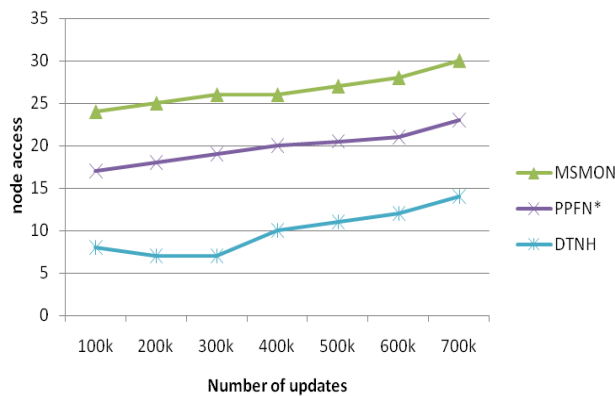


Figure.5 Update Cost Comparison

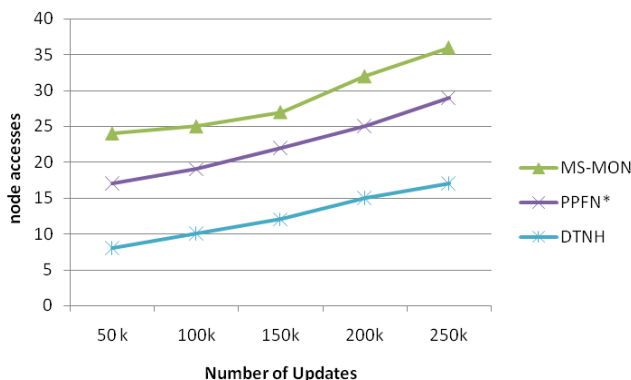


Figure.6 Query Efficiency Comparison

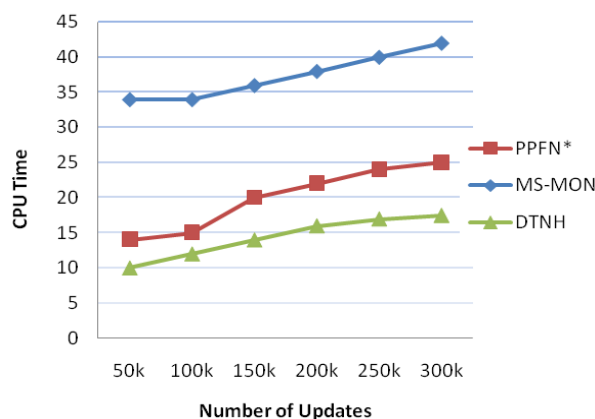


Figure.7 Query Cost Comparison

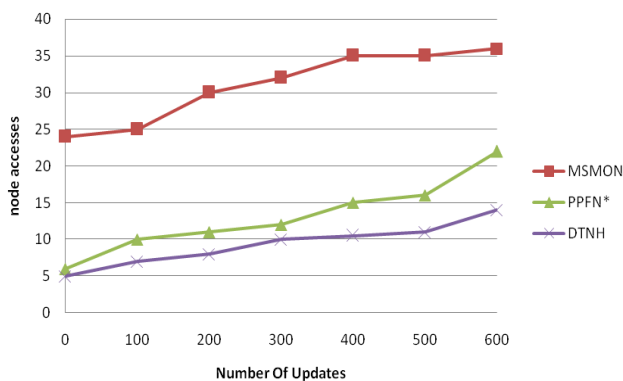


Figure.8 Trajectory Query Cost Comparison

## Query Efficiency

Data are generated 0 to 1000 and different query ranges are taken for checking query efficiency and cost comparison. The query efficiency and cost comparisons are compared with previous indexing methods Shown in the Figure 6 and Figure 7. The query performance is increased because R-Tree is avoided and overlapping also avoided. The query range is divided into segment wise so performance is increased because within time search particular segments. The Query range segment are often very large the search time is also increased in MSMON-Tree [32] and PPFN\*-Tree [33]. Here it is avoided because no overlapping. Trajectory data are updated with help of TB\*- Tree and the query performance is increased because each segment spatial data are updated with the help of the D-Tree and the trajectory query performance shown in the Figure 8.

## 5. Conclusion

DTNH structure was proposed to index the moving objects prediction in the past, present and future positions. The existing spatial data updating indexing methods are based on the integrated binary Tree, R-Tree, R\*-Tree, Oct-Tree, Quad-Tree, Grid-Tree and Hex-Tree. The depth of these trees is unbalanced and overlapping, hence the performance is reduced in the multi-structure indexing methods. DTNH multi-structure spatio-temporal index method is proposed to find the present, past and future data. DTNH index method manages past, present and future data and it predicts the data in different position and time. This proposed multi-structure indexing method supports different types of query processing compared to the existing indexing methods. Experimental results exhibits better updation and query performance compared to the MSMON-Tree and PPFN\*-Tree. DTNH index method supports better query range in compared to MSMON and PPFN\*. The DTHN Tree base Tree Having D - Tree it is used to avoid overlapping of data, so performance of the index method is increased and time complexity also decreased. The performance of the query is compared to the previous indexing methods it shown better performance in number of queries supporting over previous indexing methods. The further research includes different direction of moving objects updation and predictions of data in different time interval are considered.

## References

- [1] M. Mokbel, T. Ghanem, and W. G. Aref, "Spatio-temporal access methods", *IEEE Data Eng. Bull.*, Vol.26, No.2, pp. 40–49, 2003.
- [2] L.V. Nguyen-Dinh, W. G. Aref, and M. F. Mokbel, "Spatio-temporal access methods: Part 2 (2003 - 2010)", *IEEE Data Eng. Bull.*, Vol.33, No.2, pp. 46-55, 2010.
- [3] X. Xu, J. Han, and W. Lu, "RT-tree: an improved r-tree indexing structure for temporal spatial databases", In: *Proc. of the International Symposium on Spatial Data Handling, SDH*, pp.1040–1049, 1990.
- [4] Y. Theodoridis, M. Vazirgiannis, and T. Sellis, "Spatio-temporal indexing for large multimedia applications", In: *Proc. of the IEEE Conference on Multimedia Computing and Systems, ICMCS*, 1996.
- [5] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel approaches in query processing for moving object trajectories", In: *Proc. of the International Conf. on Very Large Data Bases, VLDB*, pp.395–406, 2000.
- [6] P. Zhou, D. Zhang, B. Salzberg, G. Cooperman, and G. Kollios, "Close pair queries in moving object databases", In: *Proc. of the GIS*, pp.2–11, 2005.
- [7] E. Frentzos, "Indexing objects moving on fixed networks", In: *Proc. of the SSTD*, pp.289–305, 2003.
- [8] V. T. De Almeida and R. H. Gunting, "Indexing the trajectories of moving objects in networks", *Geoinformatica*, Vol. 9, No.1, pp.33–60, 2005.
- [9] M. Nascimento and J. Suva, "Towards historical rtrees", In: *Proc. of the ACM Symposium on Applied Computing (ACM-SAC)*, pp. 235-240, 1998.
- [10] Y. Tao and D. Papadias, "Efficient historical r-trees," In: *Proc. of the 13th IEEE Conference on Scientific and Statistical Database Management (SSDBM)*, pp. 223 -232, 2001.
- [11] Y. Tao and D. Papadias, "MV3R-tree: a spatio-temporal access method for timestamp and interval queries", In: *Proc. of the 27th International Conference on Very Large Data Bases*, pp.431-440, 2001.
- [12] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel approaches in query processing for moving object trajectories", In: *Proc. of the 26th International Conf on Very Large Data Bases*, pp.395-406, 2000.
- [13] V. P. Chakka, A. Everspaugh, and J. M. Patel, "Indexing large trajectory data sets with SETI",

- In: *Proc. of the Conf on Innovative Data Systems Research, CIDR*, 2003.
- [14] Z. Song and N. Roussopoulos, "SEB-tree: an approach to index continuously moving objects", *Mobile Data Management, MDM*, pp. 340–344, 2003.
- [15] J. Tayeb, O. Ulusoy, and O. Wolfson, "A quad tree-based dynamic attribute indexing method", *The Computer Journal*, Vol. 41, No. 3, pp. 185–200, 1998.
- [16] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez, "Indexing the positions of continuously moving objects", In: *Proc. of the ACM International Conf. on Management of Data, SIGMOD*, pp. 331–342, 2000.
- [17] Y. Tao, D. Papadias, and J. Sun, "The TPR\*-tree: an optimized spatio-temporal access method for predictive queries", In: *Proc. of the International Conf. On Very Large Data Bases, VLDB*, 2003.
- [18] X. Xiong, M. Mokbel, and W. G. Aref, "LUGrid: update-tolerant grid-based indexing for moving objects", In: *Proc. of the MDM*, pp. 13, 2006.
- [19] Y. N. Silva, X. Xiong, and W. G. Aref, "The RUM-tree: supporting frequent updates in r-trees using memos", *VLDB J.*, Vol. 18, No. 3, pp. 719–738, 2009.
- [20] K.S. Kim, S.W. Kim, T.W. Kim, and K.-J. Li, "Fast indexing and updating method for moving objects on road networks", *Web Information Systems Engineering Workshops*, pp. 34–42, 2003.
- [21] M. Pelanis, S. Šaltenis, C.S. Jensen, "Indexing the past, present, and anticipated future positions of moving objects", *ACM Transactions on Database Systems (TODS)*, Vol. 31, No. 1, pp. 255-298, 2006.
- [22] Z.H. Liu, X.L. Liu, J.W. Ge, and H.Y. Bae, "Indexing large moving objects from past to future with PCFI+-index", In: *Proc. of the COMAD*, pp. 131–137, 2005.
- [23] D. Lin, C. Jensen, B. Ooi, and S. Saltenis, "Efficient indexing of the historical, present, and future positions of moving objects", In: *Proc. of the MDM*, pp. 59–66, 2005.
- [24] Z. Ding, "UTR-tree: an index structure for the full uncertain trajectories of network-constrained moving objects", In: *Proc. of the MDM*, pp. 33–40, 2008.
- [25] H. Lin, "Indexing the trajectories of moving objects", In: *Proc. of the International Multi Conf. of Engrs. and Computer Scientists*, pp. 732–737, 2009.
- [26] Y. Fang, J. Cao, Y. Peng, and L. Wang, "Indexing the past, present and future positions of moving objects on fixed networks", In: *Proc. of the International Conf on Computer Science and Software Engineering*, pp. 524-527, 2008.
- [27] D. Pfoser, "Indexing the trajectories of moving objects", *IEEE Data Engineering Bulletin* Vol. 25, No. 2, pp. 2–9, 2002.
- [28] Z. He, C. Wu, G. Liu, Z. Zheng, and Y. Tian, "Decomposition tree: a spatio-temporal indexing method for movement big data", *Journal Cluster Computing*, Vol. 18, No. 4, pp.1481 -1492, 2015.
- [29] I. Sasikala, M. Ganesan, and A. John, "Uncertain data prediction on dynamic road network", In: *Proc. International Conf on Information Communication and Embedded Systems (ICICES)*, 2014.
- [30] Y. Zheng, "A fast index method for moving objects on full temporal query", In: *Proc. of the 3rd International Conf on computer Research and Development (ICCRD)*, 2011.
- [31] Y. Fang, J. Cao, J. Wang, Y. Peng, and W. Song, "HTPR\*-tree: an efficient index for moving objects to support predictive query and partial history query", *Web-Age Information Management Lecture Notes in Computer Science*, Vol. 7142, pp. 26-39, 2012.
- [32] Z. Zhu, Q. Yang, and D. Pi, "Past, current and future positions index of moving objects in networks future", *Control and Automation Lecture Notes in Electrical Engineering*, Vol. 172, pp. 335-342, 2012.
- [33] Y. Fang, J. Cao, Y. Peng, N. Chen, and L. Liu, "Efficient indexing of the past, present and future positions of moving objects on road network", *Web-Age Information Management Lecture Notes in Computer Science*, Vol. 7901, pp. 223-235, 2013.
- [34] T.C. Segar and R. Vijayaragavan, "Pell's RSA key generation and its security analysis", In: *Proc. of the Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1-5, 2013.
- [35] C. Thirumalai, "Physicians drug encoding system using an efficient and secured linear public key cryptosystem (ESLPKC)", *International journal of pharmacy and technology*, Vol. 8, No. 3, pp. 16296-16303, 2016.
- [36] C. Thirumalai, and S. Shanmugam, "Multi-key distribution scheme using Diophantine form for secure IoT communications", In: *Proc. of the IEEE IPACT*, 2017.



- [37] C. Thirumalai, and P. Viswanathan, “Diophantine based asymmetric cryptomata for cloud confidentiality and blind signature applications”, *JISA, Elsevier*, 2017.
- [38] A. John, M. Sugumaran, and R.S. Rajesh “Indexing and query processing techniques in spatio-temporal data”, *Ictact Journal on Soft Computing*, Vol. 6, pp.1198-1217, 2016.