Integrated Intelligent Research (IIR)
International Journal of Communication and Networking System
Volume: 05, Issue: 01, June 2016, Pages: 43-46
ISSN: 2278-2427

Survey of Indexing Techniques on Moving Objects in Spatio Temporal Networks for PPF

S. Sathya[1], P. Sugumaran [2]
[1]Research Scholar, Dept. of CSE, Bharathiar University, Coimbatore.
[2] Professor and Head, Dept. of CSE, Pondicherry Engineering College, Puducherry

Abstract - In spatial-temporal applications, moving objects identify the particular locations called location based services and update the locations continuously to the server. Due to the enormous collections of moving objects, many spatial-temporal access techniques are developed to process user queries efficiently. Spatial- temporal access methods are classified into four categories: (1) Indexing the past data, (2) Indexing the current data, (3) Indexing the future data, and (4) Indexing data at all points of time [1]. Spatial objects whose position changes for every seconds. It helps to find the moving objects i.e., vehicles moving on the road networks, finding the position of the aeroplanes or any criminals etc. To achieve this an innovative indexing techniques needed to update data frequently.

Keywords: Spatial-Temporal, Moving Objects, Indexing, Querying.

I. INTRODUCTION

The one of the emerging trends and technologies such as wireless communications and hardware, global positioning systems and personal locator software enable a range of new personal information services to be available. Most of those are related to the user's geo-location. Example services include searching process applications that allow mobile phone users to locate any persons or landmarks. Services may also deliver maps, directions, or traffic reports commonly says environmental monitoring, real-time navigational systems [2]. These new features are supported by location-based services such like traffic management systems, mobile communication systems, sensor-based surveillance systems, etc. The location-based services necessities are growing very rapidly. It could be elaborately categorized into 4 major types: informational, tracking, emergency and employee services. [3] Spatial-temporal databases involves with moving objects that updating locations over time to time. Usually Moving objects catch their locations obtained through location-aware devices to a spatial-temporal database server. The server stores all the particulars about moving objects because of replying the queries of historical (storing information's of past) and current information. So the data stored in the server which is dynamically varied information. The supporting indexing techniques should be dynamic indexing methods. In earlier works it was found only historical data of Static location updation techniques. In recent years, a number of surveys has been published about the spatial-temporal database accessing methods [4]. Collectively, they give a wide-ranging summary about the field and afford a useful classifications of the different area investigated by the spatial database accessing methodology. The area discussed include representation, spatial access methods, system architectures, spatial database languages, query optimization etc., But all these areas does not includes accessing based on particular time slots in the spatial database.

If it includes then it is called as spatial-temporal data accessing approach.This paper is organized as follows. Section II describes about the related works. Section III explains about discussed in detail about spatial-temporal access methods in an entirely different approaches and exhibits weakness in existing approaches. In addition having separate indexing techniques for historical, current and future data for spatial-temporal moving objects. Some of the accessing techniques have been proposed to work with data accessing at Euclidian spaces as well as 3 dimensional spatial environment. Eg. road networks.

II. INDEXING

In all databases speedy accessing of raw data in spatial-temporal databases depends on the physical association of the warehoused information and the availability of the suitable indexing techniques. Many of these indexing techniques may suit for a particular querying methods only. Not for every indexing techniques may be suitable to access and answering all types of querying techniques. Indexing and querying is like in phrasal bread and butter. Without the indexing there is no more effectiveness for querying approaches. Many of the existing indexing techniques are derived from the famous and well known structures like R-tree, Quad trees and Grid file. That is extension of these three basic indexing ideas.

III. RELATED WORKS

Indexing future trajectories increases several problems when compared to the indexing past trajectories. The objective is to capably retrieve objects that will satisfy some spatial condition at a future time given their present motion vectors. Based on some basic index techniques such as R-tree, quad tree, B-tree, and grid file, along with attributes such as three dimensional, temporal (history and prediction) etc., indexes with different structures are experimented for different objectives to look for boosting the performance. In this section can report the detailed performance testing and analysis on a few index methods which is applicable both for past, present and future indexing. And to identify the advantages and disadvantages of these indexes related with certain types of queries.

A. Guttman. et. al [5] proposed R-tree and its extensions R-tree is an extension of B-tree and also height-balanced indexing structure in the multi-dimensional space. Each leaf node of the R-tree indicates a hyper rectangle ie., minimum bounded rectangle (MBR) in d dimensions. It contains objects of any types of moving object networks. Then some other familiar index techniques drawn-out from this are R*-Tree, packed R-Tree, CR-Tree, and R+-tree, etc. [6][7] are designed

Volume: 05, Issue: 01, June 2016, Pages: 43-46
ISSN: 2278-2427

to be the disk-oriented indexing techniques. A best accessing is achieved by selecting the node size to be a multiple of disk page size. R*-tree is the different from R-tree where characterization of overlapped rectangles and splitting algorithm is different [8].

Merits

R and R* trees allows overlapping of traversing while searching objects in a euclidean space as well as polylines. All the existing methods are framed out only by its prominent accessing methodology. Allows inter traversal between leaf nodes. When the nodes become overflow, then it allows to split into themselves. Efficient pruning are used to decrease the number of routes visited by the consequent searches. R* tree automatically updates location objects instantly and also incorporates a combined optimization of area, margin and overlap of each enclosing rectangle in the directory. R* tree outperform existing R-tree variants and efficiently support point and spatial data at the same time. Insertion and deletions can be intermixed with queries, no periodic global reorganization is required.

Demerits

Needs frequent updation of moving objects simultaneously, then only the latest positions are extracted by the querying server. Due to overlapping of multiple rectangles the possibilities are to create duplicate objects are in more than one nodes. These trees plays a vital role in the spatial access methods of at present data sets. R* trees implementation cost is only slightly higher than other R-trees.

Pfoser et al. [10] proposed the FNR and MON tree indexing methods Another approach, known as the Fixed Network R-tree (FNR) [9], separates spatial and temporal modules of the trajectories and indexes with in the ***time intervals*** during which any moving object was on a given network link. It can applied in a three-dimensional fixed network that should convert into two sub divisions of 2D and 1D transformations of the trajectories. The fixed network is a set of connected multiple line segments that is polylines instead of just one line segment. FNR uses 3D R-tree to index static objects. The Moving Object Network (MON-tree) [11] approach further expands the performance of the FNR-tree by revealing each edge by multiple line segments (i.e., polylines) instead of just single line segment ie., set of junctions is nodes and routes is nothing but edges in which routes are non-intersecting polylines. Unlike the FNR tree objects is updated only when it leaves a line segment.

Merits

FNR and MON tree has the ability to retrieve with in a particular time slot. So the accessing objects should be exact without having out dated moving's object datasets. It can work with three dimensional sets of spatial groups in a poly line segments. Both FNR and MON tree indexes the historical or past spatial-temporal data sets. Since historical is not feasible to address feature perception.

Demerits

In this FNR and MON tree, the objects movements are not retrieved in the middle of a line segment and also their speeds or directions cannot be changed in between a line segment. FNR tree cannot be implemented in dynamic moving object trajectories. Both these trees can have two layers each dependent on another bottom layer without the top layer the lower layer has R tree for inserting obects that cannot be addressing the exact location. If the time intervals exceeds simultaneously the bouding rectangle may get elapsed.

Papadias et al. [13] proposed the TPR and TPR tree* TPR tree proficiently indexes the current and predicted future indexes of moving objects network. The main focus of the most recent studies is time-parameterized R-tree (TPR-tree) and its variations are based on R-tree and the *Bx*-tree. These structures use the linear prediction model to support predictive queries and to reduce the number of index updates. However, the assumption of linear movement limits their applicability in a majority of real-life applications especially in traffic network where vehicles change their velocities frequently.

Merits

Both TPR and TPR* can maintain recent update history of continuously moving object in a future trajectories due to time parameterized accessing techniques. It maintains most recent locations and also can review its motion function. The server can easily locates the objects have the same motion functions. These trees are come into the types of time parameterized access methods. Whereas TPR* tree not only consider timing factor it also includes two opposite corner of the bounding recangle in order to avoid the enlargement of d-space rectangles.

Demerits

In the TPR-tree,[12] it is possible that an MBR may contain objects moving in opposite directions, or objects moving at different speeds. As a result, the MBR may expand rapidly, which may create large overlap with other MBRs. Since different objects have varied motion function, so every time exactly should identify the speed of the moving object or else cannot address it. The precise motion function cannot be identified by the server for future positioning. For every time need to mention the entry and exit time as well as corners of rectangle.

Kollios et al. [15] proposed MVR-tree The MVR-tree is optimized from the original multi-version framework, taking into account spatial properties. It is helpful to index historical that is past data retrieval and based on this tree more number of extensions are inherited. MVR tree [16] maintain small fragment of spatiotemporal data at the server at the most recent updates and distribute obsoleted data sets to other sites. It is done by clustering data that are nearest in space and time stamp in a small number of hierarchical nodes. One important feature of multi-version framework is the utilization of version copies to handle node overflows.

Merits

Tree nodes contain expired data sets can be easily removed over the network to other sites. Using this tree can achieve high update rates. MVR tree is better indexing structure with stretched time intervals that are updated infrequently and creates long bounded rectangles for efficient clustering [17].

Demerits

Long bounding boxes are truncated naturally by introducing redundancy. Excessive redundancy is harmful. Space consumption is increased. Compromise interval query performance, as multiple copies of the same object need to be retrieved.

J.D. Chen et al. [14] proposed the ANR tree The adaptive objects avoids this aforementioned problem by grouping objects having similar moving patterns. It is a two level indexing methodology adopted. It comes under the future trajectories of moving objects in a constrained network. At the top level layer has 2D R-tree to indexes the moving objects and at the lower layer another 1D R-tree is used to index adaptive groupings. Each adaptive unit has a collections of moving objects with time parameter of similar direction and speed

Merits

In this ANR tree has greatly reduced the storage space for pointing individual objects. If we concentrate on individual objects swallow more time consumption. Instead of this disadvantages we group objects of same speed and same direction. So it greatly reduced the updation cost and effective query accessing is possible.

Demerits

At one particular stage the grouping objects has in situation to get come out of the grouping object due to divert the direction, every time the user has to revise the speed and direction of the vehicle or any moving objects. So this type of indexing methods needs very frequent updation with minimal cost consumption of grouping objects is mandatory.

IV. TABLE OF COMPARISON

Qualitative comparison of trees against existing approaches.

Indexing Methods	Features	Advantages	Disadvantages	Performance
R-tree and R* trees.	a) Supports spatial data accessing. b) Suitable for Knn search option.	a)Allow bulk loading and overlapping, b)Reinsertion optimizes	Increases time complexity.	Not suitable for bulk updation. Lazy Node splitting. Support historical indexing methods.
FNR and MON tree.	Allow Spatial and temporal data accessing	Specify stipulated time slots.	Limited to Fixed networking.	Indexing historical data accessing.
TPR and TPR*	Update most recent location	Access different speed and motion of moving object.	Continuously needs updation of objects.	Indexing future position of moving objects.
MVR tree	Implementing multiple branches of Spatial access method	Allow cross reference	More chances for redundancy of data objects.	Imply only spatial access methods for indexing past data.
ANR tree	a) Grouping similar motions of moving objects. b) Support two level indexing layer.	a) Accept bulk updation. b) Reduce storage space due to grouping objects.	More expensive Cost implementation.	Indexing future position of moving objects.
AMLR* tree	Supports the above mentioned features in a single indexing mode.	Allow bulk updation, insertion and deletion with stipulated time slot.	Above mentioned inconvenience eluded.	Indexing past, current and future data retrieval.

V. RESEARCH DIRECTIONS

As per the above tabulation each mentioned indexing techniques has its own drawbacks and merits. This survey was analyzed from the year 2003 to 2012 which is having so many indexing techniques. Among all we have taken the efficient methods which is applicable for past, present and future indexing in

Moving Objects. A dynamic data structure in spatial indexing for multiple varieties of network. The combinations of Multi versioned R-tree with Adaptive R*-tree to neighboring objects using clustering algorithm with similar movement patterns with time parameterized attribute. Finally comparing the metrics of querying with this afore said indexing techniques give out performed results in spatio-temporal networks.

VI. CONCLUSION

This survey classifies and gives an overviews existing spatio-temporal indexing techniques during the past decades to till now for both past, present and future indexing. This paper also addresses solution the problem of efficient indexing of moving objects in spatial networks for assisting heavy updates of data sets. To achieve the limitations of the network and the stochastic (random probability pattern that may not predicted precisely) performance of the real traffic to achieve both high update rates even in cost wise as well as query efficiency.

References

[1] M.Mokbel, T.Ghanem, and W.G.Aref. Spatio-temporal access methods. IEEE Data Eng. Bull. 26(2): 40-49, 2003.
[2] Speicys L, Jensen C S, Kligys A. (2003) Computational data modeling for network-constrained moving objects, Proc. of GIS'03, Louisiana, USA, 2003
[3] Graham Chen. (2003) A tutorial on location based services. http://www.cs.umn.edu/Research/shashi-group/CS8715/MO2_Tutorial-GrahamChen.pdf
[4] D.J. Abel. "What's Special about Spatial?" Proc. Of the 7th Australian Database Conference. Melbourne, Australia, 72-81, 1996.
[5] A.Guttman. R-trees: A dynamic index structure for spatial searching. In SIGMOD, pages 47-57, June 1984.
[6] K. Kim, S. K. Cha, and K. Kwon. (2000) Optimizing multidimensional index trees for main memory access. In Proc. of SIGMOD 2000.
[7] Dmitri V. Kalashnikov, Sunil Prabhakar, Susanne Hambrusch, and Walid Aref.(2002) Efficient evaluation of continuous range queries on moving objects. In DEXA, 2002.

Volume: 05, Issue: 01, June 2016, Pages: 43-46
ISSN: 2278-2427

[8] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger,(1990) "The R*-tree: An efficient and robust access method for points and rectangles," Proceedings of ACM SIGMOD Int'l. Conf. on Management of Data, pp. 322-331, 1990.
[9] E.Frentzos, Indexing objects moving on fixed networks. In SSTD, pages 289-305, 2003.
[10] Pfoser D.Jensen CS (2003) Indexing of Network Constrained Moving Objects. In: Proceedings of the 11th ACM International Symposium on Advances in GIS 2003. Pp 25-32.
[11] Almeida VT, Guting RH (2004) Indexing the Trajectories of MO in Networks. Geoinformatica 9(1):33-60.
[12] S.Saltenis, C. Jensen, S. Leutenegger, and M. Lopez, Indexing the positions of continuously moving objects. In SIGMOD, pages 331-342, 2000.
[13] Y.Tao, D.Papadias, and J.Sun. The TPR* -tree: An optimized Spatio-temporal access method for predictive queries. In VLDB 2003.
[14] J.D. Chen and F.Meng. Indexing future trajectories of MO in a constrained network. Journal of CS and Tech. 22(2):245-251, 2007.
[15] G.Kollios, V.J. Tsotras, D.Gunopulos, A. Delis, and M. Hadjieleftheriou. Indexing animated objects using spatiotemporal access methods. IEEE Transactions on Knowledge and Data Engineering,13(5):758-777, 2001.
[16] Y.Tao, D.Papadias, MV3R-tree: A Spatio-temporal access method for timestamp and interval queries. In the VLDB Journal, pages 431-440.
[17] M. Hadjieleftheriou, V.Kariakov,G.Kollios: Spatio-temporal data services in a shared nothing environment. June 2004, IEEE Computer Society