Moving Object Databases-Indexing Algorithms

İrfan Ayabakan and Perihan Kilimci

Abstract—**In order to effectively use knowledge - one of the most significant elements of our era; it must be managed well and made useful. Until today, classical databases enabled people to share data, reduce unnecessary data and provide data security, etc. more easily. However, they also lacked to use of spatial and temporal data in a dynamic environment efficiently. The methods which can explain the concept of change in the space regarding time found place among moving object databases. As the concept of space and time region started to find a place in databases, they naturally brought new techniques associated with adding data, deleting data, updating data and performance criterion. The achieved improvements concerning indexing methods in moving object databases, which is a very popular subject, will surely provide enormous benefits to the pertinent technology field.**

Index Terms—**Indexing algorithms, moving object databases, spatio-temporal data.**

I. Introduction

Database means relational data which is linked to each other. Apart from storing the data, database also stores the relations between the data. The idea of processing on a database comes from running a query on data, updating the database regarding changes that formed or might form and making some sort of reports out of the data [1]. It is essential to have an access to information whenever needed.

If we take a look on the improvements related to database over the course of time, the initial systems held the data as a file base [2] and this file-based storing unnecessarily caused to increase coverage range of the data, hence prevent the data from being used by other applications [3]. File-based data storage has two significant disadvantages; descriptions of the data is contained inside the application programs and there is not any control mechanism associated with data access and data management [4], [5].

The hierarchical data model and network data model on information systems were formed along with the improvements on database over years. These systems which have hierarchical and network data model solved many problems related to file-based system, yet they also caused new problems such as accessing the data through complex structured programs. Classical database or relational database models were improved over time in order to support information systems. Thus subjects like reducing the unnecessary data, simplifying data, sharing data and data security were progressed and brought along massive convenience [5], [6].

Relational database simply holds data forms like number, text, character, date and provides opportunity to manage these. However when locational data and temporal data is present relational database is deficient to manage such databases.

Moving object databases were developed because they enable to embody object movements, spatial and temporal changes in object interactions on database management systems and also run applications on these subjects.

Manuscript received February 5, 2014; revised March 26, 2014.

The authors are with the Turkish Air Force Academy Computer Engineering Department, Turkey (e-mail: irfanayabakan@hvkk.tsk.tr, p.kilimci@hho.edu.tr).

II. Moving Object Database

A. Purpose

Main reason that moving object databases were developed is because the objects relocate, as shown in Fig. 1, according to their spatial and temporal region. Therefore we can indicate the database offers a solution between temporal and spatial data [7].

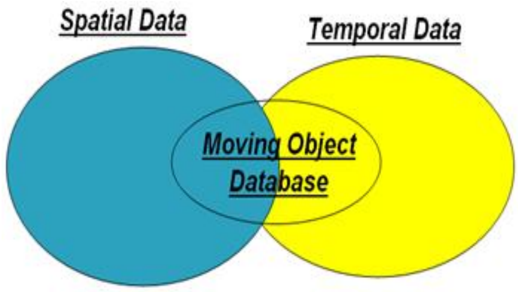

Fig. 1. Moving Object Database regions.

The main approach which leads the development of logic behind the moving object database is that "is it possible to hold constantly moving data in databases?" As here in defined, coming from the denotation of data, there are changes in space conditional to time. One of the tipping points that can be achieved is effectively managing spatial data depending on past, present and future.

Moving object databases consider classical databases as a tool for storing the data and bringing it up and classical databases consider moving object databases as a subsidiary application.

B. Approach to Movement

Lets approach to movement with a question like *"find landing fields that are by farthest 200 kilometers far from the A01 tail-code plane"*.

If we handle the query according to the question the coordinates of A01 tail-code plane in space, the coordinates of the landing fields that are 200 kilometers proximate to the given reference point and once again ranges of these landing fields; they all go into the spatial data field. Variety of the fields, their names, types of the planes and their tail codes are part of the classical data field.

DOI: 10.7763/IJCTE.2014.V6.909

If we examine the query as *"the point which the plane passed 10 minutes ago"* then we will see there is a matter of past time. According to this we will understand that the path which the plane followed in the past related to time must be effectively stored and processing techniques must as well be applied regarding to this criterion. Some of the other examples are as follows;

- F-22 plane listed as tail code A02 how many times came to the point S1 more than 3 minutes late? (Past time)
- Find the planes at the moment within 100 km radius of S1 point. (Present time)
- Is there a helicopter which will pass above S1 point in 10 minutes? (Future time)

As we can see from the questions, the term movement has a broad concept. In terms of temporal, future time's development methods are bringing a new dimension to the moving object databases. Because, unlike past and present time there is movement's behavior and approach present in the future. For instance, weather forecasting is based on assumptions which are also related to a future time approach, hence, a design for tornado and storm movement's behavior can be foreseen by using this method.

If spatial/temporal data are held together this is called embedded system and if spatial/temporal data are held separately then it is called discrete layer system. In embedded systems like Oracle Spatial, IBM DB2 Spatial all the data is held at the same tables. In the discrete systems like PostGis, MySQL Spatial Extender the data which are not spatial/temporal are held in discrete data set [8].

C. Scope of Moving Object Database

Moving object database has a broad application scope especially for tracking of movements. It includes:

- Geographical Applications
- Digital War Fields
- Traffic Control Units and Applications
- Air Traffic Control Units and Applications
- Data Mining
- Logistic Management, etc.

Moving object database is quite popular due to the fact that it has efficient support for those kind of sectors hence it is developing very rapidly.

D. Secondo

Secondo is a typical example of moving object database which can be used to understand architecture of these databases and algebras, query non-standard data types as complex data, data histories of movement as time dependent moving point and moving region. It is designed at the Fern Universität in Hagen [9].

There are some operators to explain movement better:

- Trajectory: This operator can compute the path that moving point follows through space as a function of time.
- Distance: Its possible to compute the distance between two locations as moving point and static point depending on the time region.
- Passes: This operator checks whether a moving point passes a given region.
- Inside: A boolean value operator checks whether a moving point is inside the moving region depending on the time region.

As shown in Fig. 2 there is a demonstration of moving objects which are belonging to Berlin database. It is possible to animate spatial objects in the graphical user interface (GUI) of Secondo depending on the time region [10], [11].

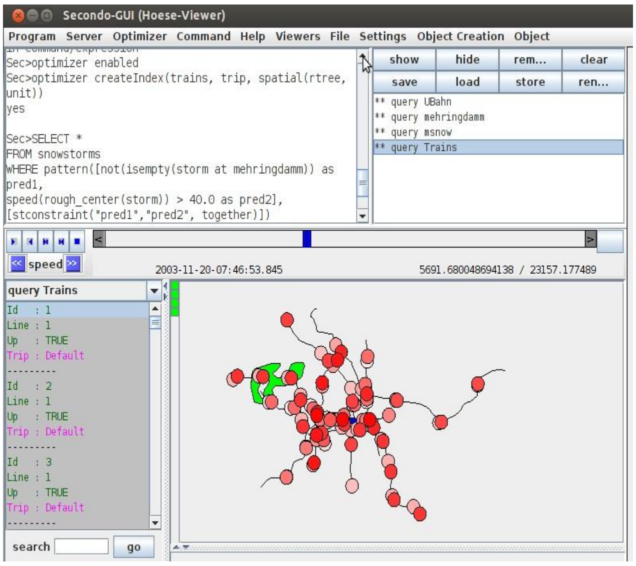

Fig. 2. Moving objects in the GUI of secondo.

III. INDEX

A. Index Approach

Index allows people to reach information rapidly. It is about direct inquiry performance. At the part where the searches are concentrating, the main logic of index is, building itself at that related part. Therefore the performance of the results are higher and more accurate. If we define a column, as shown in the below inquiry, as an index domain at the same time database will create directory inside of it which it will use later for the search. Searching in an indexed domain means searching through the directory, thus receiving feed-back will be much faster. In order to make select inquires faster, if we index the every column then this will be a mistake. As the number of indexes increases add-on and updates, then features might slow down and system performance might be affected badly in general.

```
CREATE INDEX system.plane_idx
ON system.plane(planelocation)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

The main factor in Spatial/temporal data indexing is creating groups which will have any sort of relation with the groups that are formed out of these objects and the inquiry sentence when there is a query [12].

```
SELECT *
FROM plane, landing
WHERE plane.tailcode ='A01' AND
distance(plane.geodata, landing.geodata) < 200000 ;
```

The query above will group the plane data with landing field data within the same time interval and will be configured by the database to form an answer.

These methods have the fundamental main idea of moving object database approaches. With the help of this, the spatial objects which has geometrical features can be abstracted from their complexities so they can be viewed in a more simplified shapes.

Thanks to this abstraction the substances which take place in space (point, line and region) can be structurally modeled. With these attempts, depending on the variety of indexing algorithms, time factor can take place. The relation between the modeling objects (intersection, tangent etc.) in terms of their features like (distance < 200000) and the operations defined on the objects will involve in the problem.

B. Minimum Bounding Rectangle

The main and most valid example of these approaches is minimum bounding rectangle (MBR) logic. As shown in Fig. 3 MBR surrounds an object to limit it and is defined as the smallest rectangle box. Consequently processing will not run on the objects that do not have an ordered shape yet on the MBR which surrounds them, hence, index performance will be increased considerably [12].

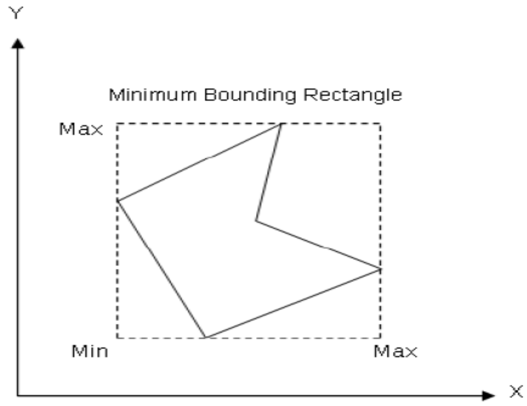

Fig. 3. Minimum Bounding Rectangle.

MBR technique depends on containing the full object and working as a filter in the usage of index. Spatial index records will be stored as a pair of [object-id, mbr]. Defining the bounding rectangles of sub-spaces dynamically and naturally will bring an extra load to system.

IV. INDEXING ALGORITHMS

Indexing methods will enable inquiry types such as spatial selection, merging, windows search, intersection, neighborhood inquiry. We can sum up the methods that apply to spatial base into 2 headlines [12]:

- Space Driven Methods: Data space will be divided into rectangular cells and distribution of data will not take into consideration. The cells and the objects that take place in space will be paired according to their features and this will create the method.
- Data Driven Methods: Distribution F of the data groups is related to segmentation of the space. Segmenting the space is aimed to accommodate with data distribution.

With the capability of demonstrating the trees in a hierarchy, Quadruple Trees or R-Trees can be listed as main examples to the methods which are applied regarding spatial base.

The bounding boxes in Quadruple Trees are indexed that is based on dividing the space into quadruple cells with recursion. Each loop that creates Quadruple Trees has 4 children, leaves will point the pages in the disc space and relevant data will be indexed according to rectangles.

The other method, data driven, can be thought of as R-Trees. R-Trees are also in a stable phase like B-Trees.

A. B-Tree

B-Tree is a stable tree in which all the records are held in tree's leaves and leaves are sequentially connected. It is a stable tree and in its each loop it shows a disc page where there are entries.

As seen in the Fig. 4, B-Tree is based on ranking one dimensional data on its leave's nodes, thus it is not efficient to store complex data which has a spatial location.

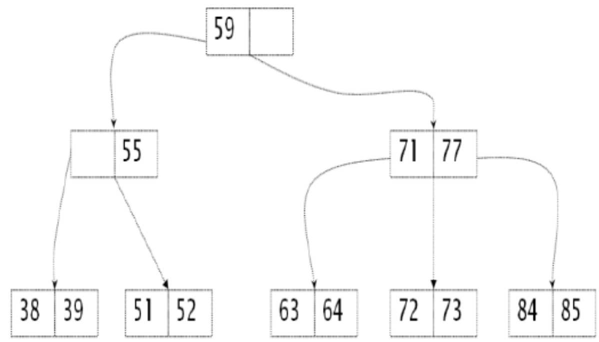

Fig. 4. B-Tree.

B. R-Tree

It comes from the B-Trees and is used to index spatial object. As the overlap decreases the query performance of R-Tree is more successful.

As the number of overlaps increases in closely spaced objects, it might not be always possible to directly get to the sought leave while running a search an object.

The most significant problem of R-Tree structure, also the problem of general tree structures, is the high cost of adding and removing objects. Due to the high costs absolute occupancy rates can not be ensured and it will cause a lower performance [12].

C. STR-Tree

As seen in Fig. 5(b) STR-Tree (Sort Tile Recursive-Tree) formation infrastructure was developed based on possible over-loads which could take place in R-Trees' infrastructure, as shown in Fig. 5(a). The idea is based on increasing the occupancy rates might give very good results in systems that do not need to be updated often and is in a stationary phase.

STR algorithm, with its current structure, offers a solution for the applications that target the stable data sets.

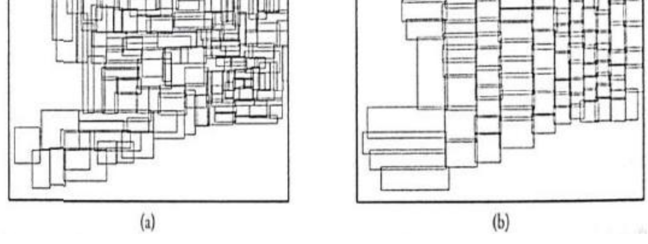

Fig. 5(a). R-Tree, (b) STR-Tree [12].

Since it decreases the tree depth and with the help of overhead operations STR-Trees contribute to performance enhancement in spatial queries due to bundling related and closer objects in the same leaves [12].

D. MX-CIF4

MX-CIF4 Tree is used for segmenting the data sets, that are formed from rectangles, in a hierarchic order and enables us to view them that way. This segmentation will keep going until there are not any loops left that can be bounded by a rectangle. The segmentation might also be ended by if the size of segmented cells get under a certain threshold level.

R-Tree and MX-CIF4-Tree are more suitable for dynamic environments thanks to their structures. In Fig. 6 there is an example structure of MX-CIF4 Tree [12].

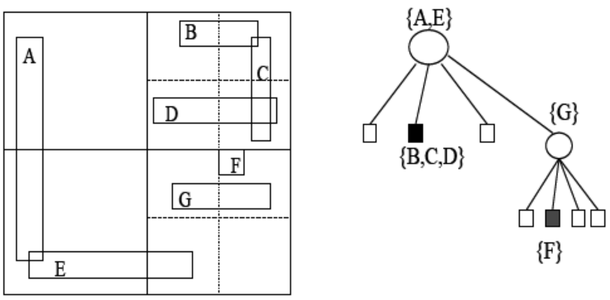

Fig. 6. MX-CIF4 tree structure.

E. MON-Tree

MON-Tree (Moving Object Network-Tree) adds time dimension to the moving objects and can be thought as a derivative of a R-Tree in the past time dimension.

The data structure is designed to suit indexing dynamic objects. The main reason behind this difference is objects that have different fields of applications. For example, the shape of the objects (rectangle, point, etc.), change in their own shape and movement in 3 dimensional space can be addressed as different applications of the same problem or subsets of it.

Nonetheless, query types bring another point of view to the problem. Time based (Past, Present and Future) movements' query models require different approaches.

R-Tree can be updated yet not for constant movement, hence there is a 2 level R-Tree designed for spatial/temporal matters. In the first level of 2 level R-Tree the path on which the movement takes place is indexed and in the second level the objects that move on the paths that come to the edges are indexed [13].

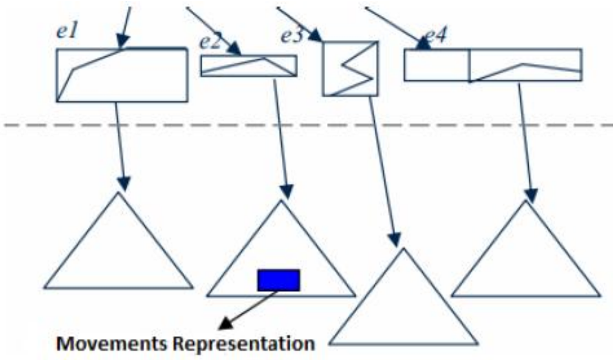

Fig. 7. MON-Tree Movement Representation [14].

During the creation of MON-Tree there are two types of add-ons; edge/route and object movement and the tree will be created accordingly. While the first level is being created with the help of edge/route adding the current network infrastructure will be incurred and this will enable to create the R-Tree. In the second level, where the actual process takes place, object movements are added according to the MON-Tree movement representation as shown in Fig. 7 [14].

F. TPR-TREE

TPR-Tree (Time Parametrized R-tree) only grounds on the logic when there is a difference in an angular speed and change in Minimum Bounding Rectangle and updating the data. This logic, generally, derived from the idea of thinking spatial data as a function of a time. In Fig. 8 bounding rectangles' movement (a,b,c and d) are represented as a unit of time and bounding rectangles adjusted themselves according to this defined time [15].

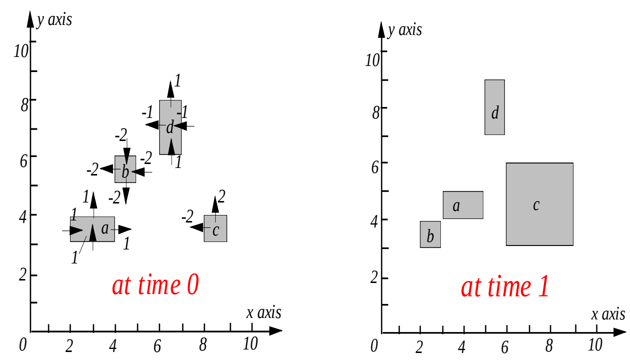

Fig. 8. TPR-Tree temporal action [15].

V. Conclusions

The transition from relational databases to moving object databases is explained and the basis which will benefit processes of spatial/temporal queries and indexing methods are examined. Moving object databases have a broad range, hence, they sprawl and evolution is still happening at a rapid rate. They will provide enormous contributions to the database field.

Indexing methods must prevent unnecessary accesses and reduce disc accesses to make efficient and fast processes possible. Indexing structure is directly proportional with the performance efficiency.

References

[1] T. Yomralıoğlu, *Coğrafi Bilgi Sistemleri, Temel Kavramlar ve Uygulamalar*, 2000.

[2] R. H. Güting, "An introduction to spatial database systems," *VLDB Journal*, Springer-Verlag, vol. 3, pp. 357-399, 1994.

[3] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 3rd ed., Addison-Wesley, 2000.

[4] T. Connolly and C. Begg, *Database Systems A Practical Approach to Design, Implementation, and Management*, 3rd ed. Addison-Wesley, 2002.

[5] A. Karaburun, "Storing spatial data in multiuser level and tracking history in database management systems," Ph.D. Dissertation, Dept. of Geodesy and Photogrammetry, Istanbul Technical Univ., Istanbul, Türkiye, 2006.

[6] J. A. Hoffer, M. B. Prescot, and F. R. McFadden, *Modern Database Management*, 2004.

[7] R. Kaftanchikova, *Moving Objects Databases*, Worcester Polytechnic Institute, April 2004.

[8] M. Hacıömeroğlu, "Geometric library in geographic ınformation systems," M.S. thesis, Dept. Comp. Eng., Başkent Univ., Ankara, Türkiye, 2006.

[9] Secondo. (2013). [Online]. Available: http://dna.fernuni-hagen.de/Secondo.html/

[10] R. H. Güting, "How to build your own moving objects database system," in *Proc. Keynote at the 8th Intl. Conf. on Mobile Data Management (MDM 2007)*, Mannheim, Germany.

[11] V. T. de Almeida, R. H. Güting, and T. Behr, "Querying moving objects in SECONDO," in *Proc. 7th Intl. Conf. on Mobile Data Management (MDM 2006)*, Demo-Paper, Nara, Japan, pp. 47-51.

[12] M. S. Ayhan, H. Sever, H. Gürçay, and S. Ak, "Comparision of spatial ındexing methods," *Ulusal Coğrafi Bilgi Sistemleri Kongresi*, Karadeniz Technical University, Trabzon, Türkiye, 2007.

[13] E. Frentzos, "Indexing objects moving on fixed networks," in *Proc. 8th ISSTD*, 2003.

[14] U. Kalay and O. Kalıpsız, "Hareketli nesnelerin İndekslenmesi," *ELECO*, 2004.

[15] Y. Tao, D. Papadias, and J. Sun, "The TPR*-tree: an optimized spatio-temporal access method for predictive queries", *VLDB*, pp. 790-801, 2003.

İrfan Ayabakan was born in İzmir in 1984. He graduated from the Computer Engineering Department of Turkish Air Force Academy (TuAFA) in 2007. He is currently a MSc student in the Department of Computer Engineering at TuAFA Aeronautics and Space Technologies Institute (ASTIN), Istanbul, Turkey. His research interests include moving object databases, geographical information systems and cyber warfare and security.

Perihan Kilimci is an instructor at the Turkish Air Force Academy, Istanbul, Turkey. Her current research interests include spatio-temporal data modeling, moving objects indexing and management of moving objects. She has also publications in the areas of electronic commerce, data mining and workflow management. She holds MS and PhD degrees in computer science from Yildiz Technical University (Turkey) respectively in 2001 and 2009.