Indexing and Querying Moving Objects in Indoor Spaces

Sultan Alamri
supervised by David Taniar
Clayton School of Information Technology
Monash University, Australia
Sultan.Alamri@monash.edu

***Abstract*—Spatial database indexes are basically designed to speed up retrievals where it is usually assumed that the objects of interest are constant unless conspicuously updated. Therefore, capturing continuously moving objects in traditional spatial indexes will require frequent updates of the locations of these objects. This paper outlines a PhD thesis that addresses the challenges of indexing the moving objects in indoor spaces. The main goal of this thesis is to develop new indoor index structures for moving objects focusing on the following four challenges: (1) introducing a queries taxonomy for moving objects to illustrate the query types for the databases of moving objects; (2) introducing an adjacency index structure for moving objects in indoor spaces; (3) capturing both spatial and temporal properties in an indoor data structure; (4) introducing an index structure for moving objects in indoor spaces that is based on a specific type of movement pattern.**

I. Introduction

Moving objects, databases are designed to store and query information about moving objects such as vehicles and mobile users [9]. In addition, the storage and manipulation of moving objects will be based on the spatial information representing static geographical objects together with temporal information. There is an important difference between the indexing of moving objects and that of static objects, that is, objects that are stationary. With static objects, spatial indexes basically assume that the objects of interest are constant unless conspicuously updated. Whereas, spatial indexes basically will require frequent updates of the locations of continuously moving objects. The main challenge facing the indexing of moving objects derives from the need to accommodate frequent updates along with obtaining an efficient query processing [7, 12].

Methods for multi-dimensional databases indexing, such as the R-tree and its variants, have been designed to support and enable efficient query processing and efficient updates [20]. This works well in applications where queries are relatively much more frequent than updates. On the other hand, applications that involve moving objects have different aspects which include large loads of updates and varieties of queries. Several new index structures have been proposed for indexing moving objects; however, these types of indexing methods for moving objects cannot be applied to indoor environments and different movement patterns. The indexing of moving objects must take into consideration the varieties of environments in order to accomplish a complete data structure for the moving objects. One of the important challenges is to accommodate different types of queries while simultaneously producing highly efficient query processing performance. Moreover, the movement patterns of moving objects might lead to different types of queries that depend on specific movement patterns.

In addition, studies have proven that people spend the majority of their time in indoor environments: working, living, shopping and entertaining [19]. Hence, indoor environments have become increasingly large and complex. For instance, the Tokyo Metro has 274 stations and 13 lines and passengers numbering in excess of 8 million daily. Consequently, the positioning and tracking of moving objects is an important research field with many applications including the tracking of moving objects, way finding, and security [13].

Much work has been conducted during the past decade in an effort to develop data structures appropriate for moving objects in outdoor spaces [9, 10, 14, 15]. However, for a number of reasons, none of these works is suitable for indoor scenarios. For instance, the current global navigation satellite systems such as GPS are unsuitable for indoor spaces. Moreover, indoor spaces are essentially different from outdoor spaces in many respects. Firstly, outdoor and indoor spaces are measured differently. In outdoor space, Euclidean space or a spatial network is typically used; whereas, indoor space is related to the notion of cellular space. Additionally, an indoor environment is comprised of different entities such as rooms, doors and hallways that both enable and constrain movement. Hence, anything that prevents or constraints movement needs to be considered in the data structure for indoor spaces. Therefore, this research focuses on building index structures for moving objects in indoor environments. Our research contributes the following:

- A taxonomy of moving objects queries to associate the different types of queries related to moving object databases in indoor environments.
- An adjacency index structure of moving objects for indoor spaces using a new technique which can enable and support the related queries in an indoor space.
- A spatio-temporal index structure of moving objects for indoor spaces, in order to enable and support both the spatial and temporal queries in an indoor space.
- An investigation and study of the different patterns of movement in moving objects in order to introduce mov-

ing object data structures based on specific movement patterns (e.g. Levy Flight pattern).

II. Related Work and Discussion

This section examines the existing researches on the indexing of moving objects which can be classified as follows: (i) trajectories, (ii) temporal, and (iii) future. One work that concentrates on *trajectories* of moving objects [4, 5] produced the Trajectory Bundle tree (TB-tree) which used the R-tree [7]. Here, the leaf node contains line segments only from the same trajectory, which helps to retrieve the trajectory of an individual object. However, the spatiotemporal range queries in the TB-tree are negatively affected [5]. The STR tree (Spatio-Temporal R-tree) [4, 5] is intended to establish spatial closeness while preserving the trajectory.

Work that concentrates on *temporal* moving objects [16, 18]. One of the first data structures to focus on historical data [11] was the Historical R-tree (HR-tree). The HR-tree uses timestamp history to construct the R-tree. Another work that used historical data is the Multi-version 3D R-tree (MV3R-tree) [16] which combines a Multi-version B-tree [9] and a 3D R-tree [18].

Work that focuses on *future* and current positions includes that of [1, 6, 8, 15]. Saltenis et al.[15] introduced the TPR-tree, (Time Parameterized R-tree) which is based on the R*-tree, in order to construct and manage moving objects. The main idea of the TPR-tree is that the index stores velocities of objects along with their positions in nodes. As an extension of the TPR-Tree, Yufei Tao proposed the TPR*-tree [17] which develops insertion/deletion algorithms in order to improve the performance of the TPR-tree.

In investigating the various moving object index structures, we notice that the availability of GPS-type positioning is either explicit or implicit in the previous works. Moreover, these works are based on Euclidean space or a spatial network; whereas, indoor space, as mentioned previously, is related to the notion of cellular space. Therefore, we argue that the ideal way of building a data structure for indoor spaces is to consider the entities (such as walls and doors) that enable or constrain movement. Furthermore, many of these works concentrated on the typical spatial queries without any consideration being given to queries regarding specific movement patterns. Our aim is to extend these types of queries to indoor spaces to include new types of queries (such as indoor adjacency queries).

III. Proposed Methods

A. A Taxonomy for Moving Objects Queries in Spatial Databases

Most of the research related to moving objects has concentrated on the extension of specialized spatio-temporal data, with no consideration given to the possible queries regarding these data. In our work, we explore the variety of possible queries about moving objects in indoor spaces. We concentrate on building a model that can integrate any data structure and query processing so that it can be used in several applications. We start with the taxonomy to give us a clear idea about the different types of queries for moving objects in indoor spaces. Our taxonomy model uses five main dimensions to retrieve moving objects: Object-Dependent Queries, Historical-Dependent Queries, Predictive-Dependent Queries, Environment-Dependent Queries and Movement-Dependent Queries. Figure 1 illustrates our queries taxonomy dimensions.

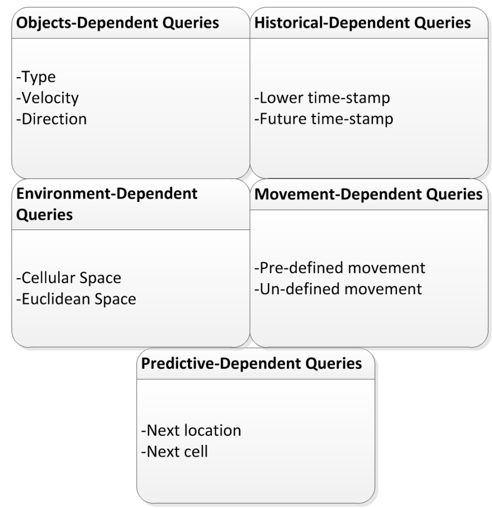

Figure 1. Moving objects' queries taxonomy

B. Efficient Indexing of Moving Objects in an Indoor Cellular Environments

As mentioned previously, an indoor environment is not a Euclidean space, where the movement constraints need to be considered to obtain the correct results. Using Figure 2, we illustrate the issue by answering a common spatial query such as (*FirstNN*). "What is the $1stNN$ objects to object O_i"? Assume that the objects are static at the current time. Given a location of O_i =[x_i,y_i] and its adjacent objects O_j =[x_j, y_j] and O_x =[x_k, y_k]. Based on the metric structure of the objects, the result will be O_j. However, we can see that O_j is incorrect because of the entities that enable or prevent movement in indoor spaces (doors and walls). The actual distance (ActDist) is greater than the Euclidean distance (Dist) (ActDist(O_i, O_j) $>$ Dist(O_i, O_j), O_j is the farthest one from O_i, and O_x is the correct result of the query *1stNN*.

Based on the adjacency concept [2], the data structure groups the entries based on their adjacency cells. This is the best means of measuring the indoor environment which is based on the adjacency and connections between cells /rooms (Definition 1). Thus, the idea is to start by grouping the objects inside the same cell. In the case of overflowing MBR, splitting will be performed to group it with one of the objects in adjacent cells based on the adjacency comparison algorithm [2, 3]. Figure 3 illustrates the spatial indoor-tree.

Definition 1: Let $P = \{O_1, O_2, ..., O_n\}$, be a set of moving objects. If $CU(P)$, the $CellDist(O_x, O_y)$ is the number of

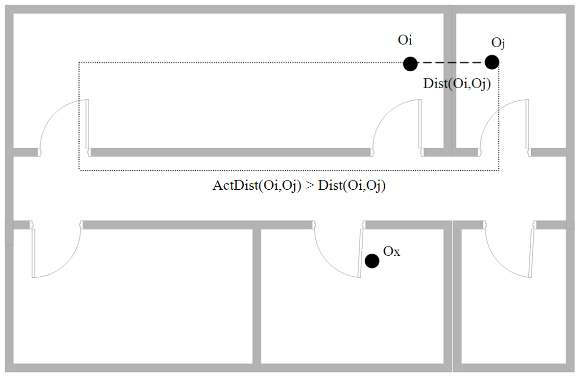

Figure 2. Using the metric/distance structures in indoor spaces can return incorrect results

hops of the cells, and is calculated as:
$CellDist(O_x, O_y)$ = $|\{C_1, C_2, ..., C_n\} - 1|$,
where $O_x \overset{in}{\to} C_1$ and $O_y \overset{in}{\to} C_n, \ \forall \ C_1 \chi C_2 .... \chi C_n$ [1]

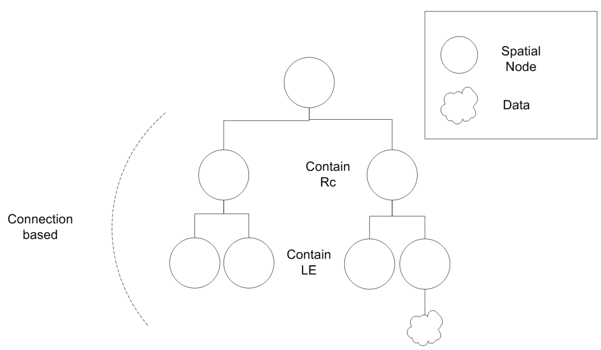

Figure 3. Indoor-tree

C. Spatio-Temporal Indexing of Objects in an Indoor Environments

In order to include the temporal part, we propose two techniques. The first technique involves monitoring and dealing with only the updated moving objects (checked out of and checked in to a new cell), this is an ideal means of reducing the updating cost. MOT-tree (Moving Objects Timestamping-tree) is also based on the adjacency method, where the data structure groups the entries based on their adjacency cells at the current time linked with their predecessors (see Figure 4). In this work, we also propose a FindPredecessor algorithm which is responsible for linking the current time with the previous times of that moving objects [3].

Definition 2. Given a set of leaf nodes $N = \{N_1, N_2, ..., N_x\}$, and a spatial object $O_v \in N_i$ located at C_j at time t_c, the Predecessor node of N_i is the leaf node that contains the last location of O_v at t_{c-1}.

[1] $Ci \ \chi \ Cj$ means Ci is connected to Cj

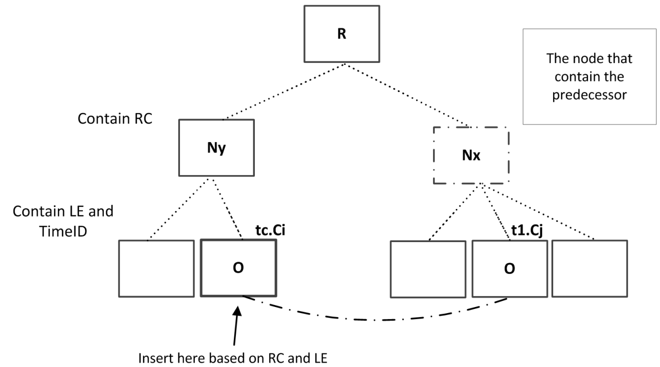

Figure 4. Moving Objects Timestamping tree (MOT-tree)

The second technique is based on the notion of a delta store Indoor trajectories delta tree (ITD-tree). Here we use the delta as a temporary recorder of the object modifications on the indoor floor. Next, we provide the trajectory delta's definitions.

Definition 3. Given a cell C_j which contains objects on the indoor floor, *CellDeltaForward* is defined as the *Cellsdelta* which records the changes made to objects in C_j at a certain time (Checked In).

Definition 4. Given a cell C_j which contains objects on the indoor floor, *CellDeltaBackward* is defined as the *Cellsdelta* which records the Checked Out made to objects in C_j at a certain time.

In this data structure, the index concentrates on the deltas of the data in each cell. This technique includes the following: after the index tree has been constructed, the updating of the moving objects is tracked by the *cellsdelta* to determine the updating of moving objects at each instant of time. Similar to the previous technique, connectivity is also based on the notion of adjacency, where we check the RC (Range cells) at the non-leaf node or the LE (the largest expand point) [2]. Figure 5 illustrates the delta-tree of the indoor trajectories.

IV. Next Steps

In the immediate future, we intend to work on indexing moving objects based on specific patterns of movement such as a levy flights pattern. Therefore, the indexing and querying of moving objects based on specific patterns of movement will allow an easy access to the data and resolve any queries better and faster.

Moreover, we intend to adapt the notion of the adjacency cells to outdoor spaces. Numerous applications are interested only in the area of the moving objects (e.g. taxi system); therefore, such applications are not useful for determining the exact location of moving objects. In these applications, spatial properties are the area or the cells that have been pre-computed and fixed by the system. Therefore, we will propose

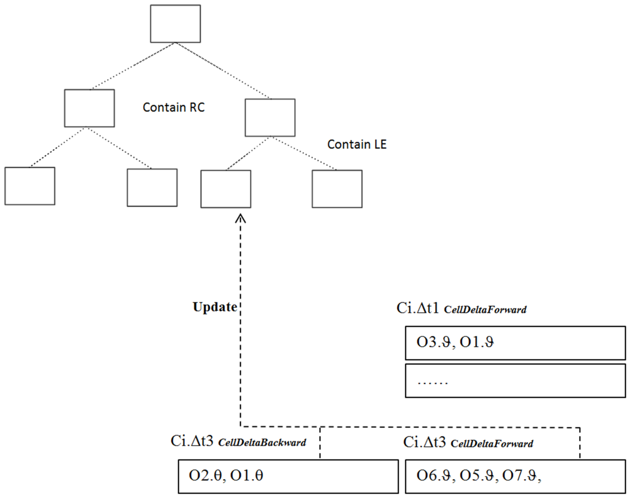

Figure 5. Indoor trajectories delta tree (ITD-tree)

an efficient data structure index for the moving objects in the adjacency data applications.

V. Conclusion

In this paper, we tackle the challenges of indexing moving objects in spatial databases. We introduce our moving objects' queries taxonomy which presents the variety of queries that can be used in moving objects databases. The taxonomy explains unsupported environments in the current data structures (indoor environments). Therefore, for indoor environments, we propose an adjacency queries approach which has been addressed in our data structure indoor-tree. Moreover, MOT-tree and ITD-tree combine the spatial and temporal properties of indoor environments with an efficient update cost. Moreover, a pre-defined movement can raise unique queries related to that movement which will be our future research direction.

References

[1] Sultan Alamri, David Taniar, and Maytham Safar. Indexing moving objects for directions and velocities queries. *Information Systems Frontiers*, pages 1–14, 2012.

[2] Sultan Alamri, David Taniar, and Maytham Safar. Indexing moving objects in indoor cellular space. In *NBiS*, 2012.

[3] Sultan Alamri, David Taniar, and Maytham Safar. Efficient indexing of spatiotemporal objects in indoor environments. Submitted to AINA, 2013.

[4] V. Prasad Chakka, Adam Everspaugh, and Jignesh M. Patel. Indexing large trajectory data sets with seti. In *CIDR*, 2003.

[5] Jae-Woo Chang, Jung-Ho Um, and Wang-Chien LeeP. A new trajectory indexing scheme for moving objects on road networks. In David Bell and Jun Hong, editors, *Flexible and Efficient Information Handling*, volume 4042 of *Lecture Notes in Computer Science*, pages 291–294. Springer Berlin / Heidelberg, 2006.

[6] Yong-Jin Choi, Jun-Ki Min, and Chin-Wan Chung. A cost model for spatio-temporal queries using the TPR-tree. *Journal of Systems and Software*, 73(1):101 – 112, 2004.

[7] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *International Conference on Management of Data*, pages 47–57, 1984.

[8] Bin Lin and Jianwen Su. On bulk loading TPR-tree. In *Proceedings of the International Conference on Mobile Data Management*, pages 114 – 124, 2004.

[9] Dan Lin. *Indexing and Querying Moving Objects Databases*. PhD thesis, National University of Singapore, Singapore, 2006.

[10] Dan Lin, Rui Zhang, and Aoying Zhou. Indexing fast moving objects for knn queries based on nearest landmarks. *Geoinformatica*, 10(4):423–445, December 2006.

[11] Mario A. Nascimento and Jefferson R. O. Silva. Towards historical r-trees. In *Proceedings of the 1998 ACM Symposium on Applied Computing*, SAC '98, pages 235–240, New York, NY, USA, 1998.

[12] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, Sung Nok Chiu, and D. G. Kendall. *Spatial Tessellations:Concepts and Applications of Voronoi Diagrams*, pages 585–655. John Wiley Sons, Inc., 2008.

[13] Mario Muñoz Organero, Pedro J. Muñoz Merino, and Carlos Delgado Kloos. Using bluetooth to implement a pervasive indoor positioning system with minimal requirements at the application level. *Mobile Information Systems*, 8(1):73–82, 2012.

[14] Dieter Pfoser, Christian S. Jensen, and Yannis Theodoridis. Novel approaches to the indexing of moving object trajectories. In *International Conference on Very Large Databases*, pages 395–406, 2000.

[15] Simonas Saltenis, Christian S. Jensen, Scott T. Leutenegger, and Mario A. Lopez. Indexing the positions of continuously moving objects. *SIGMOD Rec.*, 29(2):331–342, May 2000.

[16] Yufei Tao and Dimitris Papadias. Mv3r-tree: A spatio-temporal access method for timestamp and interval queries. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 431–440, San Francisco, CA, USA, 2001.

[17] Yufei Tao, Dimitris Papadias, and Jimeng Sun. The TPR*-tree: An optimized spatio-temporal access method for predictive queries. In *VLDB*, pages 790–801, 2003.

[18] Yannis Theodoridis, Michael Vazirgiannis . Timos Sellis, Michael Vazirgiannis, and Timos Sellis. Spatio-temporal indexing for large multimedia applications. In *ICMCS*, pages 441–448, 1996.

[19] O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang. Moving objects databases: issues and solutions. In *Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on*, pages 111 –122, jul 1998.

[20] Geng Zhao, Kefeng Xuan, W. Rahayu, D. Taniar, M. Safar, M.L. Gavrilova, and B. Srinivasan. Voronoi-based continuous k nearest neighbor search in mobile navigation. *IEEE Transactions on Industrial Electronics*, 58(6):2247 –2257, june 2011.