Moving Objects in Networks Databases*

Victor Teixeira de Almeida

LG Datenbanksysteme für neue Anwendungen
Fachbereich Informatik, Fernuniversität Hagen
D-58084 Hagen, Germany
`victor.almeida@fernuni-hagen.de`

Abstract. Moving objects databases have become an intensive field of research in recent years with many applications such as location-based services, traffic monitoring, fleet management, etc. Most of the works in the literature assume free movement in the 2-dimensional space, although in some cases, the objects move within spatially embedded networks, e.g. vehicles in highways and trains in railways. Moreover, these works are focused on isolated aspects such as efficient query processing with specialized index structures. The aim of this PhD. project is to present a prototype of a complete database management system for efficiently storing and querying moving objects in networks, providing a comprehensive data model supporting the description of complete histories of movement, index structures for efficient query processing, an extended model handling uncertainty, and a complete integrated implementation as an algebra inside the SECONDO extensible database system.

1 Introduction

With the development of wireless network communications (e.g. the IEEE 802.11 protocol) and positioning technologies such as the Global Positioning System (GPS) and the European Satellite Navigation System (GALILEO), devices equipped with such technologies like handheld devices, on-board units in vehicles, or even mobile phones have become relatively cheap and are predicted to be in widespread use in the near future. This trend will lead to many new kinds of applications, e.g. location-based services and spatio-temporal data mining. The challenge to the database community is how to handle such complex spatio-temporal data in database management systems assuming the presence of huge amounts of historical data.

There are two main approaches in the literature that try to model this problem. These can be characterized as the *location management* and the *spatio-temporal database* perspectives. First, Wolfson et al. in [25,28] developed a model called Moving Object Spatio-Temporal (MOST) and the Future Temporal Logic

---

* This work was partially supported by grants Gu 293/8-1 and Gu 293/8-2 from the Deutsche Forschungsgemeinschaft (DFG), project “Datenbanken für bewegte Objekte” (Databases for Moving Objects)

(FTL) language for querying current and anticipated future locations of moving objects. Motion vectors are stored into dynamic attributes for the moving objects. Only moving point objects are considered.

Second, the spatio-temporal database perspective was explored, which means that the complete trajectories of the moving objects are stored such that querying past states is possible. Güting et al. in [6, 9] provide a complete framework for the representation and querying of spatio-temporal data, namely *moving point* and *moving region*. Such data types can be embedded as attribute types into extensible database management systems. Our work follows this approach and therefore throughout this paper when we mention moving objects we are interested in the spatio-temporal database perspective.

Since then, the field has flourished and a lot of work has been done especially on efficient query processing providing index structures mainly focused on the range and nearest neighbor queries, e.g. [8,13,18,21] just to mention some of the most recent ones.

An important observation that has not been addressed in the research mentioned above is that in many cases objects do not move freely in the 2-dimensional space but rather within spatially embedded networks, e.g. roads or highways. One could then describe movement relative to the network rather than 2-dimensional space which would enable easier formulation of queries and, even more important, more efficient representations and indexing of moving objects.

There exist some works in the literature focusing on modeling issues for moving objects in networks and some presenting specialized index structures and query processing algorithms for the range query, which are discussed on Section 2. However, there is a big gap between data modeling and query processing. A comprehensive data model and query language for objects moving in networks does not yet exist. As long as this is so, it is not clear how the proposals for efficient indexing and query processing can be integrated and used in a database system.

The purpose of this PhD. project is then to build a complete prototype of a vertical database system for moving objects in networks, i.e. to provide a model containing data types for the network as well as for static and moving objects together with a comprehensive set of operations among them, efficient algorithms for the operations, index structures to improve query processing, optimization rules that enable the usage of such indexes, and a complete integrated implementation inside the SECONDO extensible database system [5, 10, 12].

This paper is organized as follows: Section 2 presents the most closely related work. Section 3 details the PhD. project proposal, the results achieved so far, and the future work that needs to be done. Finally, Section 4 concludes the paper.

2 Related Work

Network query processing, in particular shortest path computation, has been considered by Shekhar et al. in [23, 24] and Rundensteiner et al. in [15, 16].

In [24] an adjacency list data structure clustered into pages is based on the z-order of node positions. A similar one using the Hilbert ordering is presented in [19]. These works are considered inside the internal structures of our proposed network representation.

Considering models, Vazirgiannis and Wolfson in [27] present a first model for querying moving objects in road networks. The network model basically corresponds to an undirected graph, where nodes are street crossings and edges are city road blocks. Moving objects are described by geometric polylines, as in the earlier work for unconstrained movement mentioned in Section 1. Besides, the network model and the query language are limited compared to our proposal.

Two papers by Jensen et al. [14, 17] have also looked at data modeling issues for spatial networks with respect to possible uses for location-based services. They describe as a case study the data model used by the Danish road directory and a Danish company. The emphasis is to explain that real road networks are quite complex, and that just simple directed graph models are not sufficient. The case study suggests a model that uses several interrelated representations which are expressed in terms of relational tables. This is an interesting application study, and we have drawn some of our motivation to use a route-oriented model from the first of these papers. However, moving objects are not considered.

The same group has described a more formalized model incorporating some of these ideas into [26]. They propose to use two complementary models of a network together, namely the 2-dimensional representation and the graph representation. The first is geared to describing a network at very high detail, while the second should support efficient computations. Data and query points are available in both models to represent static objects (e.g. facilities) and moving query objects (e.g. vehicles). The paper further describes how the graph representation can be derived from the 2-dimensional representation.

This model is the closest to our network model, but both representations are graph-oriented, i.e. they do not offer a route-oriented model as we do (see Section 3.2, and they do not offer a model for moving objects in networks, in the sense that trajectories of moving points relative to the network are not available. Furthermore, only point objects are considered.

The route-oriented model is closely related to the kilometer-post representation in [17] and to the concept of *linear referencing* widely used in the GIS-T (Geographic Information Systems in Transportation) literature, e.g. the work from Scarponcini in [22], where positions are described relative to the length of a road. Linear referencing is also already available in commercial database products such as Oracle Spatial.

Finally, index structures for the trajectories of moving objects in networks are presented by Frentzos in [7] and by Pfoser and Jensen in [20]. Both use the same idea of converting a 3-dimensional problem into two sub-problems with lower dimensions, where the first one is to index the network data and the second is to index the moving objects. It is shown that the problem then becomes simpler using this approach and their index structures outperform 3-dimensional structures.

3 The PhD. Project

3.1 The Proposal

The PhD. work started in September 2002 as part of a Deutsche Forschungsgemeinschaft (DFG) project named “Datenbanken für bewegte Objekte” (Databases for Moving Objects) under supervision of Prof. Dr. Güting. The project is divided into two parts having two years duration each. The main goals of the first part, related to the PhD. work presented in this paper, were to build the main model for moving objects in networks, and extensions to this model supporting dynamic networks and to cope with uncertainty, all integrated; to define an implementation strategy; and to start building a prototype given the implementation strategy.

The second part of the project is still in progress and is more focused on implementing a prototype running inside SECONDO, on proposing efficient algorithms for the operations, on studying efficient execution of query processing using indexes, and finally on extending the optimizer to cope with complex objects such as moving objects (constrained to networks or not).

Inserted into this project, the main goal of this PhD. work is to build a complete database system for moving objects in networks providing a vertical solution from a model with data types and operations, efficient algorithms for the operations, indexing structures for efficient query processing, and optimization rules enabling the usage of such indexes.

3.2 Results Achieved So Far

A model for moving objects in networks is presented in [11]. The core of this model are the data types *network*[1], *gpoint*, and *gline* to represent the underlying network (highway network), network positions (motels or gas stations), and network regions (speed limit or construction areas), respectively. Obviously, the corresponding moving data types for network positions (vehicles or trains) and regions (traffic jam area or part of the network affected by a snow storm) are defined, namely *moving*(*gpoint*) and *moving*(*gline*).

One should note that the model is consistent with (and can be seen as an extension of) the one in [9] allowing us to re-use many concepts and facilities provided there and to completely integrate them in order to be able to handle interactions between both network constrained and unconstrained spatial and spatio-temporal data. An example is the second **inside** operation in the list of operations below which uses a 2-dimensional region (*region* data type defined in [9]) as argument.

The main novelty on the network data type is that it is modeled in terms of *routes* and *junctions* and not in terms of nodes and edges of a graph, and we name it the *route-oriented model.* A route corresponds to a path over a graph possibly containing several edges, and junctions store intersections between pairs

---

[1] We write data types in italics underlined, and operations in bold face.

of routes. This representation has several advantages, which are detailed in [11], but the perhaps most practical one is that the representation of a moving object becomes much more compact in this way. If positions are given relative to edges, then for example a vehicle going along a highway at constant speed needs a change of description at every node (exit/junction) because the edge identifier changes. If positions are given relative to routes, then the description needs to change only when the vehicle leaves the highway.

Another important point is that the description of the network is not too simplistic. Routes can be bi-directional, i.e., admit movement in two directions, and positions on the two sides of a route, e.g. on a highway, can be distinguished, so that the distance between positions on each side of a route can be quite big in some cases. On the other hand, there are also cases where one does not want to distinguish between positions on two sides of a road, e.g. people moving around in a pedestrian zone. Therefore, two kinds of routes called *simple* and *dual routes* are provided. Furthermore, we do not assume that all transitions are possible in a junction, which is not realistic. The possible transitions between routes at junctions are stored into 4x4 matrices encoded into integer numbers, namely the *connectivity code.*

Object data types are then described relative to the network rather than the embedding 2-dimensional space, leading to a more compact representation of moving objects, since no geometric information needs to be stored. Geometry is stored once and for all with the network. Besides, discovering relationships between objects and parts of the network becomes much simpler and more efficient in this way.

The point data type (*gpoint*) is represented by a route location containing a route id, a relative position on that route, and its side, for dual routes. The network region data type (*gline*) is represented by a set of intervals of route locations. To the best of our knowledge, this is the first work that handles network regions. For the moving counterparts of these data types, we provide linear functions for the time-dependent location (*moving*(*gpoint*)) and for the route interval boundaries (*moving*(*gline*)).

Operations are then provided for these data types. Some examples are

mgpoint × *gline*	→ *mbool*	**inside**
mgpoint × *region*	→ *mbool*	**inside**
mgline	→ *mreal*	**length**
mgpoint × *gpoint*	→ *mreal*	**distance**

Their semantics are straightforward and can be found in [11] together with the complete set of operations. Three applications containing some sample queries are presented in [11] as well as some implementation issues.

An index structure to store the trajectories of moving objects supporting the route-oriented model is presented in [2, 3], namely the MON-Tree. It supports the range query employing a similar approach of dividing the problem into two sub-problems presented in [7, 20], where a top R-Tree indexes the routes in the network and bottom R-Trees index objects' movements inside each route (Figure 1. Given the advantages of using the route-oriented model, the MON-Tree outperforms the competing index structures.

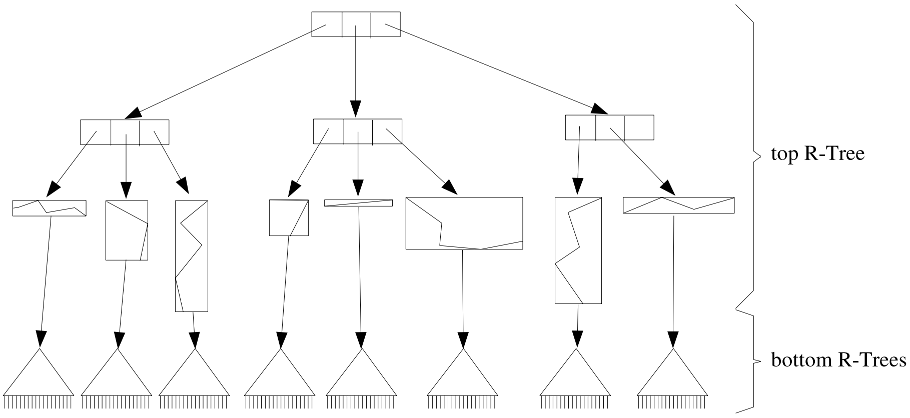


Fig. 1. The index structure of the MON-Tree.

An extension to the model in [11] to cope with uncertainty is presented in [4]. Most of the data types were extended to their uncertain counterparts, e.g. the 2-dimensional point is now expressed as a region with uniform distribution function and the boolean data type has a new *maybe* value. The geometry of the uncertain trajectories of point objects with movement constrained to networks is presented (Figure 2), as well as the data type representation and the operations from [9,11] are extended. Finally, it is explained how we could modify the MON-Tree to index the trajectories of such moving objects with uncertainty.

Before starting the implementation of moving objects in networks, several implementation tasks have been done in the SECONDO system. A persistent version of the *Relational Algebra* containing almost all operations such as selection, projection, sorting, hash join, sort-merge join, loop join, etc. supporting large objects in tuples was implemented, which was demonstrated at ICDE'05 ([10]). The data types and the most important operations from the model in [9] were also implemented as an algebra in SECONDO, namely the *Spatial Algebra*. A demonstration of all these features together is accepted to the demo session at MDM'06 ([5]). Finally, several other improvements in the SECONDO system have been done during this PhD. project, which are not here discussed given space limitations.

3.3 Further Work

The SECONDO extensible database system is composed by three major components written in different languages:

- the kernel, written in C++, implements specific data models, is extensible by algebra modules, and provides query processing at executable level over the implemented algebras;

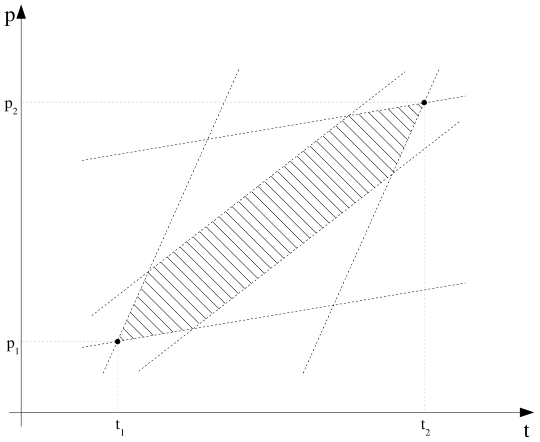


Fig. 2. The uncertain geometry of a moving object between two measurement points p_1 and p_2.

- the optimizer, written in Prolog, provides as its core capability conjunctive query optimization, currently for a relational environment and also implements the essential part of SQL-like queries;
- and the graphical user interface (GUI), written in Java, which is an extensible interface for such an extensible system like SECONDO, where new data types or models can provide their own way to be displayed.

The implementation of the data types and operations presented in the models in [4,11] as an algebra in the SECONDO kernel is still in progress and we plan to finish it in the near future.

In the optimizer, we are currently investigating how to convert from SQL queries to the best executable plan, or at least to an efficient one, in the presence of such complex data types. In this case, it is important to note that the choice of SECONDO as a database system was a good one, because we do not need to provide complex selectivity estimation functions. SECONDO uses the sampling approach to estimate selectivities.

For cost estimation, since we use abstract data types, and in this case instances of these data types can be very big, the execution time of some operations in one object (inside a tuple) is not negligible in the query processing time and must be taken into account by the optimizer. We plan to estimate the time for complex operations also using the sampling approach.

In order to provide efficient query processing, we need to identify the operations where an index can be helpful, provide such an index (if needed), and provide some optimization rules in order to use such indexes. In some cases, we discovered that we can re-write the query adding some predicates in the *where* clause to enforce the usage of such indexes.

As an example, let us take the parcel delivery application presented in [11]. The application models a company offering express delivery of packages in the city network of Hagen, Germany. We assume that we have the road network of the city of Hagen as a data object, a relation *road* mapping road names to route identifiers in the network, and a relation called *postman* describing post workers' trips.

```
road( name: string, route: int )
postman( name: string, trip: mgpoint )
```

We focus our discussion on query **P4**, which returns all post workers who stayed in the street “Hagener Strasse” for more than one hour yesterday. This query should be written as

```
SELECT p.name
FROM   postman AS p, road AS r
WHERE  r.name = 'Hagener Strasse' AND
       duration(deftime(at(atperiods(trip, yesterday), route)))
         < one_hour
```

assuming that *yesterday* and *one_hour* are pre-defined objects storing the period of yesterday and the duration of one hour, respectively. This query first reduces the trips to the period of yesterday (**atperiods**) then to the times where they were at the route named ”Hagener Strasse” (**at**), computes their temporal dimension (**deftime**) and compares their duration (**duration**) to the *one_hour* duration. The most expensive part of this query is

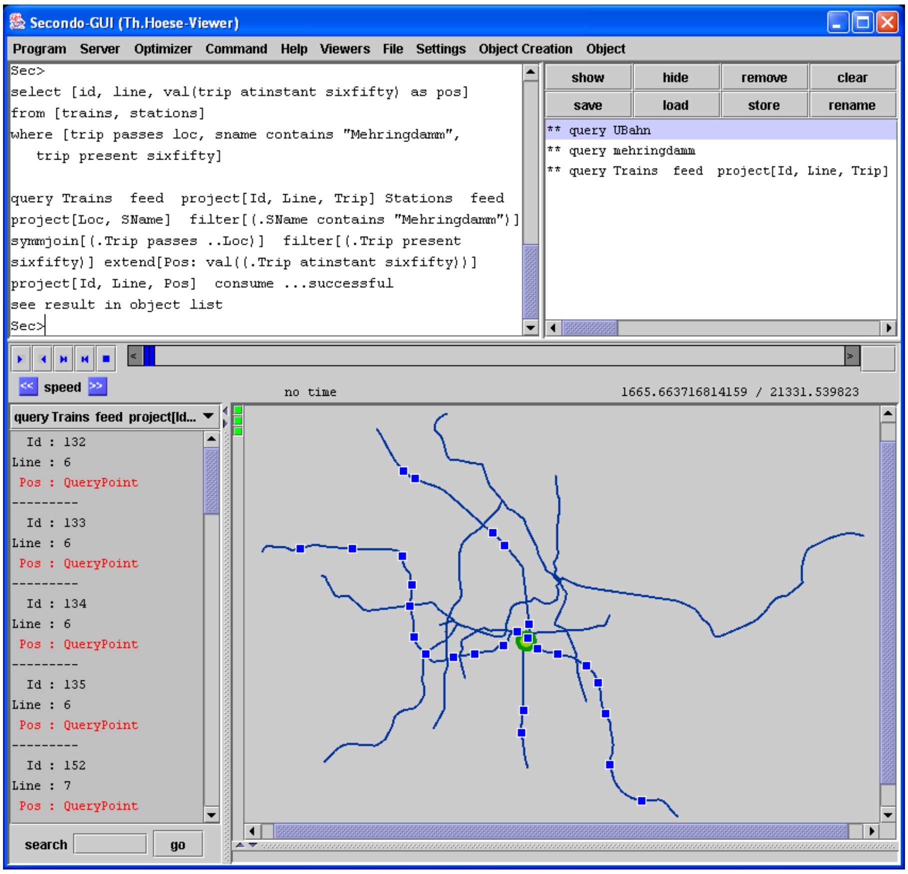
```
at(atperiods(trip, yesterday), route)
```

which is a selection in time and integer spaces. A temporal index could be available in the system with entries in the format ⟨*time_interval*, *route_id*⟩. An example of such an index is [1]. Moreover, the query optimizer should be smart enough to recognize this pattern and to add further conditions to the query so that this index is used. The query that would be then evaluated is

```
SELECT p.name
FROM   postman AS p, road AS r
WHERE  r.name = 'Hagener Strasse' AND
       duration(deftime(at(atperiods(trip, yesterday), route)))
         < one_hour
       present(trip, yesterday)
       passes(trip, route)
```

where **present** and **passes** are the counterpart predicates for **atperiods** and **at**, respectively.

Finally, we will also provide specific methods for displaying the data types in the SECONDO GUI. We think that a visualization tool is very helpful for doing research in moving object databases. An example of the GUI with spatio-temporal data of some trains of the city of Berlin is shown in Figure 3.



```
Sec>
select [id, line, val(trip atinstant sixfifty) as pos]
from [trains, stations]
where [trip passes loc, sname contains "Mehringdamm",
   trip present sixfifty]

query Trains  feed  project[Id, Line, Trip] Stations  feed
project[Loc, SName]  filter[(.SName contains "Mehringdamm")]
symmjoin[(.Trip passes ..Loc)]  filter[(.Trip present
sixfifty)] extend[Pos: val((.Trip atinstant sixfifty))]
project[Id, Line, Pos]  consume ...successful
see result in object list
Sec>
```

```
** query UBahn
** query mehringdamm
** query Trains  feed  project[Id, Line, Trip]
```

```
  Id : 132
Line : 6
 Pos : QueryPoint
---------
  Id : 133
Line : 6
 Pos : QueryPoint
---------
  Id : 134
Line : 6
 Pos : QueryPoint
---------
  Id : 135
Line : 6
 Pos : QueryPoint
---------
  Id : 152
Line : 7
 Pos : QueryPoint
```

Fig. 3. The Secondo graphical user interface.

Acknowledgments

I would like to thank Prof. Dr. Ralf Güting for giving me the opportunity to come to Germany and to work with his group, Dr. Zhiming Ding who contributed a lot with the project, to all my colleagues in the FernUniversität in Hagen that helped me during my PhD. work, and to all students that contributed to the Secondo extensible database system.

4 Conclusions

The proposed PhD. work investigates the problem of building a prototype of a complete database system for moving objects in networks. We showed the importance of this field of research and the lack of such solution in the literature.

We believe that this is the first attempt to build such a prototype of a complete database system for both unconstrained and network constrained moving objects.

References

1. B. Becker, S. Gschwind, T. Ohler, B. Seeger, and P. Widmayer. An asymptotically optimal multiversion B-tree. *VLDB Journal*, 5(4):264–275, 1996.
2. V. T. de Almeida and R. H. Güting. Indexing the trajectories of moving objects in networks (Extended Abstract). In *Proc. of the 16th Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*, pages 115–118, 2004.
3. V. T. de Almeida and R. H. Güting. Indexing the trajectories of moving objects in networks. *GeoInformatica*, 9(1):33–60, 2005.
4. V. T. de Almeida and R. H. Güting. Supporting uncertainty in moving objects in network databases. In *Proc. of the 13th Intl. Symp. on Advances in Geographic Information Systems (ACM-GIS)*, pages 31–40, 2005.
5. V. T. de Almeida, R. H. Güting, and T. Behr. Querying moving objects in SECONDO. Accepted for publication at the Demonstration Session in the 7th Intl. Conf. on Mobile Data Management (MDM), 2006.
6. L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider. A data model and data structures for moving objects databases. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*, volume 29(2), pages 319–330, 2000.
7. E. Frentzos. Indexing objects moving on fixed networks. In *Proc. of the 8th Intl. Symp. on Spatial and Temporal Databases (SSTD)*, pages 289–305, 2003.
8. E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis. Nearest neighbor search on moving object trajectories. In *Proc. of the 9th Intl. Symp. on Spatial and Temporal Databases (SSTD)*, pages 328–345, 2005.
9. R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems (TODS)*, 25(1):1–42, 2000.
10. R. H. Güting, V. T. de Almeida, D. Ansorge, T. Behr, Z. Ding, T. Höse, F. Hoffmann, M. Spiekermann, and U. Telle. SECONDO: An extensible dbms platform for research prototyping and teaching. In *Proc. of the 21st Intl. Conf. on Data Engineering (ICDE)*, pages 1115–1116, 2005.
11. R. H. Güting, V. T. de Almeida, and Z. Ding. Modeling and querying moving objects in networks. Technical Report 308, Fernuniversität Hagen, Fachbereich Informatik, 2004. *Accepted for publication in the VLDB Journal.*
12. R.H. Güting, T. Behr, V. T. de Almeida, Z. Ding, F. Hoffmann, and M. Spiekermann. SECONDO: An extensible DBMS architecture and prototype. Technical Report 313, Fernuniversität Hagen, Fachbereich Informatik, 2004.
13. M. Hadjieleftheriou, G. Kollios, P. Bakalov, and V. J. Tsotras. Complex spatio-temporal pattern queries. In *Proc. of the 31st Very Large Data Bases Conference (VLDB)*, pages 877–888, 2005.
14. C. Hage, C. S. Jensen, T. B. Pedersen, L. Speicys, and I. Timko. Integrated data management for mobile services in the real world. In *Proc. of the 29th Intl. Conf. on Very Large Data Bases (VLDB)*, pages 1019–1030, 2003.
15. Y. Huang, N. Jing, and E. A. Rundensteiner. Path queries for transportation networks: Dynamic reordering and sliding window paging techniques. In *Proc. of the 4th Intl. Symp. on Advances in Geographic Information Systems (ACM-GIS)*, pages 9–16, 1996.

16. Y. Huang, N. Jing, and E. A. Rundensteiner. Integrated query processing strategies for spatial path queries. In *Proc. of the 13th Intl. Conf. on Data Engineering (ICDE)*, pages 477–486, 1997.
17. C. S. Jensen, T. B. Pedersen, L. Speicys, and I. Timko. Data modeling for mobile services in the real world. In *Proc. of the 8th Intl. Symp. on Spatial and Temporal Databases (SSTD)*, pages 1–9, 2003.
18. J. Ni and C. V. Ravishankar. PA-Tree: A parametric indexing scheme for spatio-temporal trajectories. In *Proc. of the 9th Intl. Symp. on Spatial and Temporal Databases (SSTD)*, pages 254–272, 2005.
19. D. Papadias, J. Zhang, Nikos Mamoulis, and Yufei Tao. Query processing in spatial network databases. In *Proc. of 29th Intl. Conf. on Very Large Data Bases (VLDB)*, pages 802–813, 2003.
20. D. Pfoser and C. S. Jensen. Trajectory indexing using movement constraints. *GeoInformatica*, 9(2):93–115, 2005.
21. S. Rasetic, J. Sander, J. Elding, and M. A. Nascimento. A trajectory splitting model for efficient spatio-temporal indexing. In *Proc. of the 31st Very Large Data Bases Conference (VLDB)*, pages 934–945, 2005.
22. P. Scarponcini. Generalized model for linear referencing in transportation. *GeoInformatica*, 6(1):35–55, 2002.
23. S. Shekhar, A. Kohli, and M. Coyle. Path computation algorithms for advanced traveller information system (atis). In *Proc. of the 9th Intl. Conf. on Data Engineering (ICDE)*, pages 31–39, 1993.
24. S. Shekhar and D. Liu. CCAM: A connectivity-clustered access method for networks and network computations. *IEEE Trans. Knowl. Data Eng.*, 9(1):102–119, 1997.
25. A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Modeling and querying moving objects. In *Proceedings of the 13th Intl. Conf. on Data Engineering (ICDE)*, pages 422–432, 1997.
26. L. Speicys, C. S. Jensen, and A. Kligys. Computational data modeling for network-constrained moving objects. In *Proc. of the 11th Intl. Symp. on Advances in Geographic Information Systems (ACM-GIS)*, pages 118–125, 2003.
27. M. Vazirgiannis and O. Wolfson. A spatiotemporal model and language for moving objects on road networks. In *Proc. of the 7th Intl. Symp. on Spatial and Temporal Databases (SSTD)*, pages 20–35, 2001.
28. O. Wolfson, S. Chamberlain, S. Dao, L. Jiang, and G. Mendez. Cost and imprecision in modeling the position of moving objects. In *Proceedings of the 14th Intl. Conf. on Data Engineering (ICDE)*, pages 588–596, 1998.