M.T.L. Baars, B.G.H. Gorte, P.J.M. van Oosterom and E. Verbree

Structurering, indexing, quering and visualizing moving objects in a DBMS

SUMMARY

Modelling, analyzing, and monitoring the movements of point-objects is becoming more and more important. Database Management Systems make it possible to manage large spatial datasets. In conventional spatial database applications, data is assumed to be less dynamic. Therefore, it is hard to store continuously moving objects. In this research, the available methods investigated with respect to their ability to structure, index, query and visualize moving objects in the context of a Spatio-Temporal Database Management System. A very generic model for moving point-objects is presented together with a number of views suitable for further analysis of the data. For two case studies, some of these methods are implemented to see how it works and what are the shortcomings of these methods.

KEYWORDS: Spatio-Temporal, DBMS, structuring, indexing, querying and visualizing

INTRODUCTION

Temporality is an inherent aspect of geo-information and since a number of years it became a topic in several researches. Nowadays more and more applications of spatiotemporal GISs become important. For instance with respect to Cadastral issues (Van Oosterom, 2000), national road databases (Heres, 2000) or the detection of traffic jams (NRC Handelsblad, 2003). Geo-DBMSs make it possible to manage large spatial datasets in databases that can be accessed by multiple users at the same time. These spatial datasets usually contain 2D data, while more and more applications depend on 3D data (Arens, 2003). Moving objects can also be seen as 3D data (x-position, y-position, time) or 4D data (x,y,z,time).

Continuous movement poses new challenges to database technology. In conventional (spatial) databases, data is assumed to remain constant unless it is explicitly modified (Saltenis et al, 1999). This statement causes efficient structuring methods for the data. Also querying these datasets is an issue that has to be taken into account. Examples of queries on moving-object databases are: which object has the highest speed, what's the average distance between two objects, which objects are in a specific area (polygon) in a certain time interval, etc. Answering these queries requires efficient storage and indexing method in case the dataset is large. Visualising the moving objects and answers on the queries is also an important issue.

Two main types of applications based on moving point-objects can be recognized: the distinction between real time monitoring of moving objects and modelling the data afterwards (post processing). The main question of this research can be described as follows:

How do you structure, index, query and visualise dynamic point clouds in the context of a Spatiotemporal Database Management System?

After this introduction, this paper starts with a framework for spatiotemporal data. In the third section, a generic model will be introduced for modelling moving objects in a geo-DBMS. After that, two case studies will be described in section 4. The first case deals with moving objects in a database, where the data is post processed and the second one describes real time modelling, indexing, querying and visualizing. This paper will be finished in section 5 with conclusions and suggestions for future work.

A FRAMEWORK FOR SPATIO-TEMPORAL DATA

According to (Langran, 1992), "five technical requirements will drive the development of a temporal GIS: a conceptual model of spatial change, treatment of aspatial attributes, data processing logistics, a spatiotemporal data access method and efficient algorithms to operate on the spatiotemporal data."

The first part of this technical framework, the conceptual model, can be described as "the configuration of information, as it will be represented to the computer. It defines the entities, attributes and relationships to be portrayed; it also defines the operations to be performed and the constraints to be enforced. (Langran, 1992)". A trajectory is the path described in space and time by a moving object. Such a trajectory can be represented in different ways. Several researchers describe models to structure these data into a DBMS (see for instance (Vazirgiannis and Wolfosn, 2001), (Wolfson et al, 1998), (Marchand et al, 2003) and (Meng and Ding, 2003)).

The fourth technical requirement, mentioned by Langran is the "spatiotemporal data access method". Indexing is important to get a faster access of your data. It decreases query-response times. A lot of research has been done on spatiotemporal indexing. See for instance (Mokbel et al, 2003) for an overview of these methods or (Pfoser and Jensen, 2003) for a more detailed description. The basis of all these models is the R-tree.

The last technical requirement, efficient algorithms, is also a very important one. Perhaps it is efficient to save the results of queries that are used very often. An example is the speed of moving objects. In many data structures this attribute is not available in the base table, but it can be derived. For instance in Oracle 9i Spatial, it is possible to define materialized "views". This is a method that saves the query results of pre-defined queries (see section 3).

GENERIC MODEL

As can be concluded a number of alternative models for moving point objects be constructed. Two aspects can characterize these models. First, the time dimension is either separate or integrated with the spatial dimension (in such a case a 2D point and time become a 3D point in the spatio-temporal model). Secondly, for a single object the observations are either stored in separate records (with the sampled point in time) or in one record with a polyline attribute (kind of interpolation between the time samples). In the polyline case the time again can be separate or integrated with the spatial point data: 2D spatial polyline with separate attributes for time-stamps or 3D spatiotemporal polyline). For an overview, see figure 1.

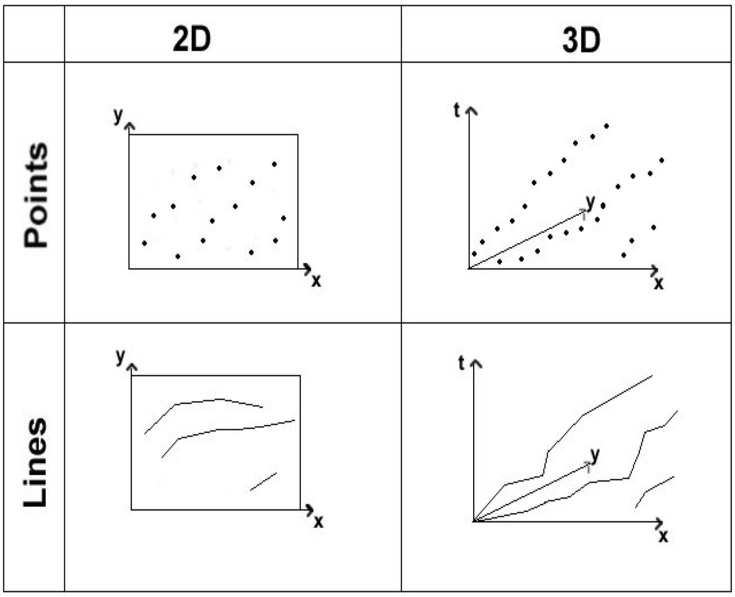

Figure 1 Top row with sampled time: 2D points (time separate) or 3D points (integrated time). Bottom row with interpolated time: 2D polylines (time separate) or 3D polylines (time integrated)

All models can be converted to each other (and could in that sense be considered equivalent) and most likely this can be realized with DBMS views (using spatial operators). In practice they may differ with respect to dynamic behaviour (suitable or not for dynamically growing data in case of real time monitoring) and ease of use during analysis and visualization. From the conceptual point of view the models are quite similar and we will use the most elementary to illustrate spatiotemporal modelling. The base table, not assuming any explicit sequence, looks like (in a kind of pseudo SQL):

```
create table mov_obj(id, t, pos) /* primary
key is pair (id, t) */
```

Based on this base table a number of views can be defined to have easy access to derived attributes such as speed and acceleration (first a view to obtain the next time stamp of a given object):

```
create view move_obj_succ as
select t1.*, t2.t as next_t
from mov_obj t1, mov_obj t2
where  t1.id=t2.id  and  t2.t=(select  min(t)
from move_obj where t>t1.t);

create view speed_obj_vw as
select   t1.id,   t1.t,   t1.next_t,   t1.pos,
dir=diff(t2.pos-t1.pos),
speed=distance(t2.pos, t1.pos)/(t2.t-t1.t)
from mov_obj_succ t1, mov_obj_succ t2
where t1.id=t2.id and t2.t=t1.next_t;

create view accel_obj_vw as
select   t1.id,   t1.t,   t1.next_t,   t1.pos,
t1.speed,     accel=(t2.speed-t1.speed)/(t2.t-
t1.t)
from speed_obj_vw t1, speed_obj_vw t2
where t1.id=t2.id and t2.t=t1.next_t;
```

Using this view one can now derive statistic such as average speed. This can either be grouped by id (of vehicle), position and time. Quite another type of view may be used to analyse how close the cars are together (as a possible indication of traffic jams).

All these views may be nice from a functional point of view. However, without the proper storage and indexing the performance may be poor. Important aspects to consider are spatiotemporal clustering (so the physical ordering of the data) and spatiotemporal indexing (efficient selections of the record addresses based on spatiotemporal (range) queries). In Oracle the initial implementation would use an indexed organized table (on the key id, t) in order to obtain ordering of the data based on id and time. Further a 2D r-tree index (on position) or 3D r-tree functional index (on position and time) is the used for initial spatiotemporal indexing.

Analysis may show that it is impossible to answer all (important) queries based on a single physical ordering the base table and in the ultimate situation redundant data storage may be considered. Oracle offer ‘materialized views’ (not part of the SQL92 standard) to implement this in a effective manner:

```
create materialized view move_obj_succ_mv1
refresh fast on commit
as select t1.*, t2.t as next_t
    from mov_obj t1, mov_obj t2
    where t1.id=t2.id and t2.t=(select min(t)
from move_obj where t>t1.t);
```

Especially in highly dynamic situations (rapid data growth) this is not without problems (as it may not be very efficient to update the materialised view after every transaction. Tuning the generic model, by choosing the appropriate storage and index structures and (materialized) views, makes it efficient for a given application, that is, a set of typical queries for a given (static or dynamic data set).

CASE STUDIES

Two case studies will be presented: one that can be considered post processing the spatiotemporal data after collection (static) and one that is intended for real time processing (dynamic). The difference is that the second one requires a dynamic data structure in order to handle the ever-growing data set (large volumes).

TRAFFIC DATA FROM HELICOPTER OBSERVATIONS

In cooperation with groups for Photogrammetry and Remote Sensing and for Traffic Management within Delft University we are populating a database with measurements of highway traffic during circumstances of congestion. The measurements are obtained by automatic analysis of image sequences that are taken with a digital high-resolution camera from a helicopter. A highway section of approximately 500m is monitored during an extended period of time (say 1 hr) with a recording frequency of 10 frames per second. On a crowded or congested highway this may lead to several millions of car observations. So, there is a dataset available with x,y and time (1/10 seconds per frame).

The moving objects database is successfully implemented in an Oracle 9i Spatial, following the principles of the generic model described in the section ‘Generic Model’. Base tables with the four geometry types from figure 1 are created from the original dataset. These tables are indexed and views with speed, distances and accelerations are created. With this set of tables and views, some interesting queries can be considered. With aggregation functions and spatial operators, as available in Oracle Spatial, some macroscopic variables (such as intensity, density and average speed) can easily be computed.

An interesting question is whether it is possible to calculate and visualize vehicles that do not keep enough distance to their predecessor. By creating a rectangle in front of the moving point object with a width of 2 meters and a length of 2 seconds (see figure 2), conform the government policy of “Keep two seconds distance” and by using a spatial operator (SDO_RELATE), the vehicles that do not meet this condition can be calculated.

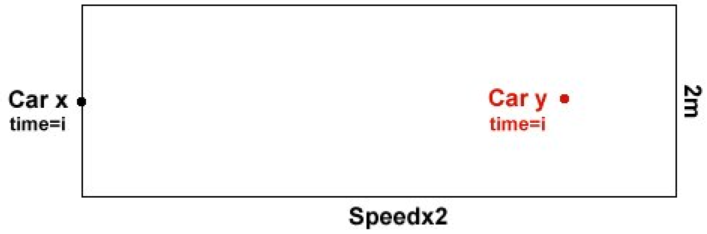

*Figure 2 A rectangle in front of car x with width 2m and length 2 seconds at current speed (unit is s * m/s = m, which is correct) where no other vehicles are allowed at time=i.*

In 'Cartography, visualization of spatial data' (1996), Kraak and Ormeling discuss the use of dynamic variables opposed to traditional graphic variables, which are used to represent spatial data within individual frames. According to them an appropriate dynamic graphic can be used if the spatial data it represents is dynamic by nature. But as Bertin has stated, the user should be aware of: 'movement only introduces one additional variable, it will be dominant, it will distract all attention from the other (graphical) variables' (Bertin, 1983). MacEachren (1994) has defined the dynamic variables: duration, order, rate of change, frequency, display time and synchronization. These dynamic variables can be seen as additional tools to design an animation.
Knowing this theory, ESRI has provided an extension on ArcGIS to handle spatio-temporal data: the Tracking Analyst (ESRI, 2004). Its functionality is comparable to its predecessor, available as an extension to ArcView 3.x, although the possibilities to serve real-time data to the application are now part of ArcIMS. It is possible to display point and track data (real time and fixed time) by temporal symbology renderers (shape and size), symbolize time by colour (show the aging of data), actions (based on attribute or spatial queries) and highlighting. Besides, the user is in control by the interactive playback manager to start, pause, stop and rewind the animation (for an example, see figure 3). Note that the colours of the objects are based on their id (which does not add much information), therefore another attribute may be more interesting to use as colour attribute basis; e.g. acceleration (red: slow down, green: speed up, and blue: about equal speed).

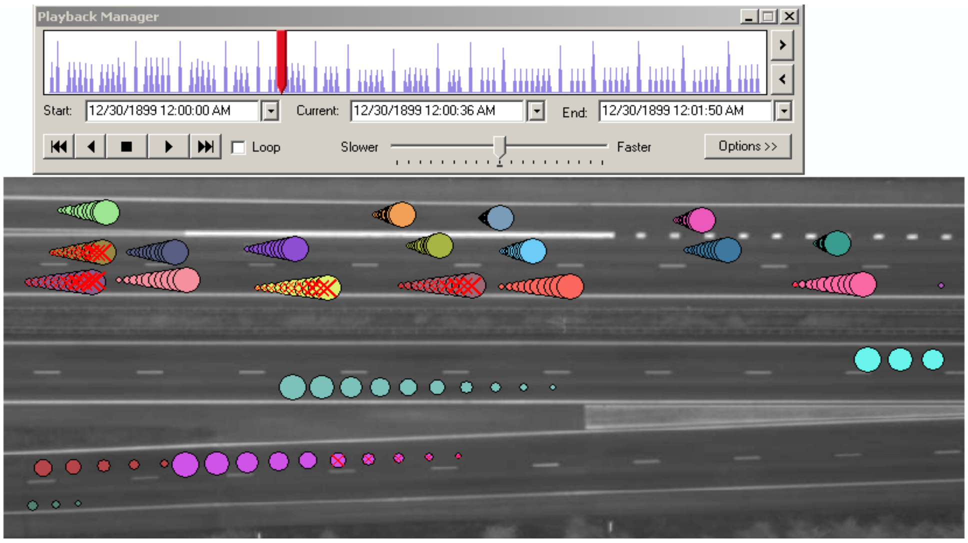

Figure3 The last 10 timestamps of vehicles (Note crosses if they do not keep enough distance)

As with most cartographical visualizations the dynamic map could be used for exploratory data analysis to reveal unknown or difficult to recover information from the data alone. One could state database queries to expose this kind of information, but some more qualitative and descriptive facts like trends, are only to be retrieved by carefully inspection of the animation. For this the dynamic variables should be appropriate and careful used, with the notification of Bertin in mind.

GPS DATA FROM MOVING VEHICLES

The generic model described earlier will become more interesting, when real time (dynamic) data is used. The Advise Office for Geo-information and ICT, part of the Dutch Ministry of Traffic and Public Works, has a dataset available for research, where GPS tracking data is stored of a number of taxis operating near Rotterdam. For a period of two years, data has been captured and stored in files.

This dataset can be simulated as real time data with an application like ArcIMS tracking server. The generic model is tested with this real time simulation. Issues like efficient storage and indexing, data consistency and performance will become more important in real time than in the post processing case.

It is proved that also for large real-time simulations, the generic model works very efficient and is very flexible. For 2D data (x,y and time as attribute) and 3D data (x,y,time), the R-tree index is tested and spatiotemporal queries have been carried out. Spatiotemporal queries demand a different kind of treatment for 3D than for 2D because of the functionality of Oracle 9i Spatial. For the indexing, it makes not much difference with respect to functionality and how many times the index has to be updated. In figure 4 you can see that the quality of the index is nearly the same in the 2D and the 3D case for the same number and same positions of objects.

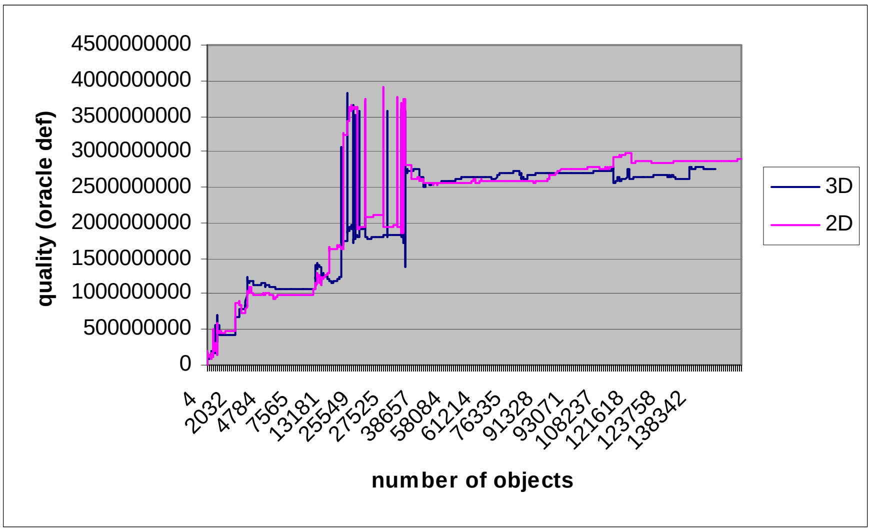

Figure 4 The index quality plotted for each number of objects for a 2D and 3D index. The quality is a number, defined by Oracle 9i Spatial. The index is updated automatically when the query times increase too much.

CONCLUSIONS AND FUTURE WORK

The first of the five technical requirements mentioned by (Langran, 1992) is the conceptual model. In this paper, 4 representations of the basic model for moving point-object (trajectories) are described. The fourth technical requirement, the spatio-temporal access method, is also described in this paper. It seems clear that there are numerous methods available to index the past, but for the current time (real time), just a few methods can be mentioned.

A very generic model for spatio-temporal point objects is presented. This model consists of a simple base table (in which data of every application should fit). Together with a number of (materialized) views, this offers a flexible and efficient method to model moving object data in a geo-Database Management System. It is fast and keeps your dataset consistent.

In a case where data was already collected (post processing), the generic model is tested and some queries and visualisations are made. In the future, the generic model will also be tested for a real time dataset. Further, the current cases have been implemented with standard Oracle functionality, such as the 3D R-tree for spatio-temporal indexing. Alternative, more specific spatio-temporal indices may be implemented in the future.

Acknowledgments

We would like to thank Dutch Ministry of Traffic and Public Works for making their data available for this research. Further, we would like to thank our colleagues, Wilko Quak and Theo Thijssen, for their comments on section 3 and specifically on the (materialized) views.

Literature

Arens, C.A. (2003), *Maintaining Reality – Modelling 3D Spatial objects in a Geo DBMS using a 3D primitive.* Delft: Master thesis, GIS Technology, TU Delft, 2003.

Bertin, J. (1983), *Semiology of graphics*. Madison, Wisconsin: University of Wisconsin Press.

ESRI (2004), *http://www.esri.com/software/arcims/tracking_server.html.* (February 2004).

ESRI, (2004), *http://www.esri.com/software/arcgis/arcgisxtensions/trackinganalyst/index.html*, 2004

Heres, L. (2000), *Hodochronologics: History and time in the National Road Database.* In: Time in GIS: Issues in spatio-temporal modelling. Delft: Nederlandse Commissie voor Geodesie, p.46-56.

Kraak, M.-J. and Ormeling, F.J. (1996), *Cartography, visualization of spatial data*. London: Addison Wesley Longman.

Langran, G. (1992), *Time in Geographic Information Systems*. London: Taylor & Francis.

MacEachren, A.M., 1994, *Visualization in modern cartography: Setting the Agenda*. In Visualization in Modern Cartography (Editors: A. M. MacEachren and D. R. F. Taylor). Oxford: UK: Pergamon), p. 1-12.

Marchand, P., Brisebois, A., Bédard, Y., Edwards, G. (2003), *Implementation and evaluation of a hypercube-based method for spatio-temporal exploration and analysis*. In: Journal of the International Society of Photogrammetry and Remote Sensing (ISPRS) theme issue “Advanced techniques for analysis of geo-spatial data” dans la catégorie “multi-scale hierarchies of spatial operators”.

Meng, X., Ding, Z. (2003), *DSTTMOD: A Discrete Spatio-Temporal Trajectory Based Moving Object Database System*. DEXA2003, LNCS 2736. Prague: Springer Verlag, p 444-453.

Mokbel, M.F., Ghanem, T.M., Aref, W.G. (2003), *Spatio-temporal Access methods*. IEEE Data Engineering Bulletin, 26(2) p 40-49.

NRC Handelsblad (9-9-2003), *article Mobiele filemeldingen* (in Dutch).

Oosterom, P. van (2000), *Time in Cadastral Maps*. In: Time in GIS: Issues in spatio-temporal modelling. Delft, Nederlandse Commissie voor Geodesie, p. 36-45.

Pfoser, D and Jensen, C.S. (2003), *Indexing of Network-Constrained Moving Objects*. In: Proceedings of the Eleventh International Symposium on Advances in Geographic Information Systems, New Orleans, 8 pages.

Saltenis, S., C.S. Jensen, C.S., S.T. Leutenegger and M.A. Lopez, (1999), *Indexing the Positions of Continuously Moving Objects*. Aalborg: A TIMECENTER Technical Report.

Vazirgiannis, M. and O. Wolfson (2001), *A spatiotemporal Model and Language for Moving Objects on Road Networks*. In: Proceedingsof the 7th international symposium on advances in Spatial and Temporal Databases, p. 20-35

Wolfson, O., B. Xu, S. Chamberlain and L. Jiang (1998), *Moving Object Databases: Issues and Solutions*. Proceedings of the 10th International Conference on Science and Statistical Database Management. Capri, Italy, p. 111-122.