Benchmarking and Data Generation in Moving Objects Databases

Theodoros Tzouramanis
University of the Aegean, Greece

INTRODUCTION

Moving objects databases (MODs) provide the framework for the efficient storage and retrieval of the changing position of continuously moving objects. This includes the current and past locations of moving objects and the support of spatial queries that refer to historical location information and future projections as well. Nowadays, new spatiotemporal applications that require tracking and recording the trajectories of moving objects online are emerging. Digital battlefields, traffic supervision, mobile communication, navigation systems, and geographic information systems (GIS) are among these applications. Towards this goal, during recent years many efforts have focused on MOD formalism, data models, query languages, visualization, and access methods (Guting et al., 2000; Saltenis & Jensen, 2002; Sistla, Wolfson, Chamberlain, & Dao, 1997). However, little work has appeared on benchmarking.

BACKGROUND

The role of benchmarking in MODs is to compare the behavior of different implementation alternatives under the same settings. It is important that the results must hold not only for a specific environment but in more general settings as well. Thus, the user is able to repeat the experiments and come to similar conclusions (Zobel, Moffat, & Ramamohanarao, 1996).

An example of a benchmark is the comparison of space requirements and query execution time of access methods for MODs. In order to compare the behavior of different indexing schemes under the same platform, there must be a flexible performance benchmarking environment. The general architecture of such a benchmarking toolkit is presented in Figure 1, and according to Theodoridis, Silva, and Nascimento (1999) and Tzouramanis, Vassilakopoulos, and Manolopoulos (2002), it consists of the following components:

- a collection of access methods for MODs,
- a module that generates synthetic data sets which cover a variety of real-life applications,
- a set of real data that also represents various real-life examples,
- a query processor capable of handling a large set of queries for extensive experimentation purposes,
- a reporter to collect all relevant output report logs, and
- a visualization tool to visualize data sets for illustrative purposes.

Good benchmarking must correspond to a recognizable, comprehensible real-life situation. In case large real data sets are not available, benchmarking requires the generation of artificial data sets following the real-world behavior of spatial objects that change their locations, shapes, and sizes over time and cover a variety of real-life applications. Thus one of the most important components of a benchmarking tool is the module that

![Figure 1. A simplified benchmarking environment for access methods for MODs](image-url)
generates synthetic data sets. In MODs, the work on the generation of synthetic data is limited, and only a few pioneering articles have recently addressed the topic of moving objects data generators.

MAIN THRUST

Synthetic Data Generators

A data set generator for MODs, called generator spatio-temporal data (GSTD), has been proposed by Theodoridis, Silva, and Nascimento (1999). It can generate moving points or rectangles and starts by distributing their centers in the workspace according to certain distributions. After the initialization phase, there are three main parameters to control the evolution of the objects throughout time, according to a desired distribution. These parameters are: (a) the duration of object instances, which involves time-stamp changes between consecutive instances; (b) the shift of the objects, which involves changes of spatial locations of the object centers; and (c) the resizing of objects, which involves changes of object sizes (only applicable to rectangular objects).

The GSTD supports three alternative approaches for the manipulation of invalid object instances in cases where an object leaves the spatial data space. However, a limitation of the GSTD approach is that the objects are moving almost freely in the workspace without taking into consideration the interaction between other objects or any potential restrictions. In particular, it could be argued that in any possible scenario, the objects are scattered all over the data space, or are moving in groups and the whole scene has the appearance of an unstructured polymorphic cloud movement. In the left part of Figure 2, a snapshot of an example of two synthetic data sets being displayed concurrently is illustrated.

In order to create more realistic scenarios, Pfoser and Theodoridis (2003) have extended the latter approach. They introduced an additional GSTD parameter to control the change of direction, and they used static rectangles for simulating an infrastructure, where the scenario indicates that each moving object has to be outside of these rectangles.

Saglio and Moreira (2001) propose the Oporto generator for time-evolving points and rectangles. It uses the modeling of fishing ships as a motivation. Ships are attracted by shoals of fish, while at the same time they are repulsed by storm areas. The fish themselves are attracted by plankton areas. Ships are moving points, whereas shoals, plankton, and storm areas are moving regions. Although useful for testing access methods, the original algorithm is highly specialized and turns out to be of limited use with respect to the demands of other real-world applications. In the right part of Figure 2, an example of the use of the Oporto generator is presented. The snapshot illustrates the motion of two ships being attracted by a gray shoal of fish.

Brinkhoff (2002) demonstrates a generator for “network-based” moving objects. It combines a real network with user-defined properties of the resulting data set. The driving application is the field of traffic telematics, and the presented generator satisfies exactly the requirements of this field. Important concepts of the generator are the maximum speed and the maximum edge capacity, the maximum speed of the object classes, the interaction between objects, the different approaches for determining the starting and the destination point of a moving object, and the recomputation of a route initiated by a

Figure 2. A snapshot of GSTD’s Web-based interface (left) and a snapshot of Oporto’s interface (right)