

Monitoring and Tracking Moving Objects in Mobile Environments

Dragan Stojanović

University of Nis, Serbia

Slobodanka Djordjevic-Kajan

University of Nis, Serbia

Apostolos N. Papadopoulos

Aristotle University, Greece

Alexandros Nanopoulos

Aristotle University, Greece

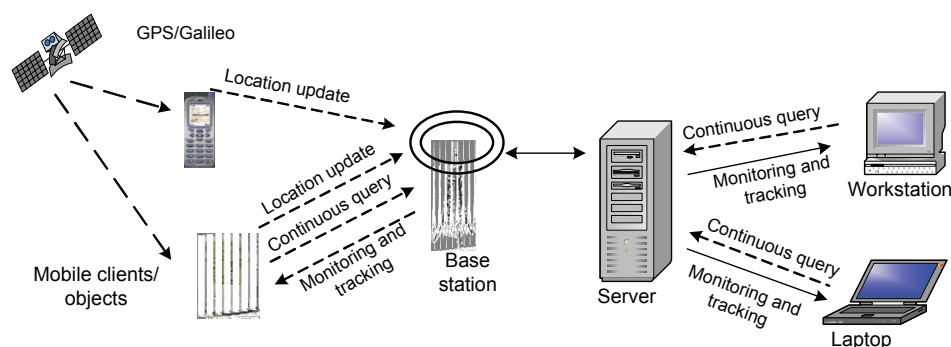
INTRODUCTION

Advances in mobile and ubiquitous computing, wireless communications and mobile positioning have given a rise to a new class of mobile information systems and services, called location-based services (LBS) (Schiller & Voisard, 2004). Such services, like fleet management, cargo tracking, child-care, tourist services, transport management, traffic control and digital battlefield rely on the tracking of the continuously changing positions of entire populations of moving objects. LBSs are becoming ubiquitous. A traffic service may inform its users about traffic jams, traffic accidents and weather situations that are expected to be of relevance to the service user. A friend finder service may inform each user about the current whereabouts of friends. Other services may monitor and track the positions of emergency vehicles, police cars, security personnel, hazardous materials, or public transport. A more advanced location-based game service may allow a group of users to play and try to surround and catch “hostile” players. Monitoring and tracking moving objects require continuously registering the positions of mobile objects, and at any instance in time to know whether those objects are

within the specified area, or a specified distance from known mobile/static objects, or which are its k nearest neighbors (k -NN). To provide monitoring and tracking of moving objects in mobile application environments, it is highly desirable and sometimes critical for the service efficiency to provide accurate results to these requests and update them in real time, whenever moving objects enter or exit the regions of interest, or become the closest neighbors to the objects of interest.

Monitoring and tracking LBS applications require database and application support to model and manage moving objects in both database and application domains. Such services must also provide efficient processing of continuous queries over moving objects. In contrast to regular queries that are evaluated only once, a continuous query remains active over a period of time. At any time there will be a number of continuous queries simultaneously running at the server. Each of these queries needs to be periodically re-evaluated as the objects and/or queries move. A major challenge for this problem is how to provide efficient processing of continuous queries with respect of CPU time, I/O time and network bandwidth utilization. The architecture of the monitoring and tracking LBS system is given in Figure 1.

Figure 1. The architecture of monitoring and tracking LBS



BACKGROUND

Monitoring and tracking moving objects in mobile environments by processing continuous queries over moving objects is an active area of research, resulting in the proposal of many query processing methods, techniques and indexing schemes. One of the challenges in monitoring and tracking LBS development is how to handle different types of queries in a mobile environment, where both queries and objects can be moving. Different types of location dependent queries are significant for the monitoring and tracking purposes, such as range queries, k -nearest neighbor (k -NN) queries, reverse neighbor queries, distance joins, closest pair queries and skyline queries. The most important type of query for the purpose of monitoring and tracking moving objects is the range query. The range may represent a user selected area, a map window, a polygonal feature, a part of the road segment or an area specified by the distance from a reference point. The map window of the LBS client application represents the simplest continuous range query that must be supported in monitoring and tracking LBS application. Using such a query, up-to-date information about moving/stationary objects in a user's surrounding is continuously represented in the map window, as he/she, as well as objects of interest, move.

Continuous query processing in a location-aware environment is an active area of research, resulting in the proposal of many query processing methods, techniques and indexing schemes. In Prabhakar et al. (2002), velocity constrained indexing and query indexing (Q-index) has been proposed for efficient evaluation of stationary continuous range queries. According to the proposed method, in-memory data structures and algorithms are developed and presented (Kalashnikov et al., 2004). By indexing queries, and not moving objects, the Q-index method avoids frequent updates of the index structure and thus expensive maintenance of this structure. The MQM method presented (Cai et al., 2004) focuses on stationary continuous range queries. It is based on partitioning the query space into rectangular sub-domains, and the assignment of the resident domain to each moving object. A moving object is aware only of the range queries intersecting its resident domain, and reports its current location to the server only if it crosses the boundary of any of these queries. Gedik and Liu (2004) propose a method and a system for distributed query processing, called Mobieyes. Mobieyes ships some part of the query processing to the mobile clients while the server mainly acts as a mediator between moving objects. The method tries to reduce the load on the server and save communication costs between moving objects and the server. In the paper by Gedik et al. (2004), the authors propose a scheme called motion adaptive indexing (MAI), which enables optimization of continuous query evaluation according to the dynamic motion behavior of the objects. They use the concept of motion sensitive bounding boxes (MSB) to model

and index both moving objects and moving queries. Mokbel et al. (2004a) present SINA, a server-side method based on shared execution and incremental evaluation of continuous range and k -NN queries. Shared execution is achieved by implementing query evaluation as a spatial join between the moving objects and the queries. Incremental evaluation means that the query processing system produces only the positive or negative updates of the previously reported answer, not the complete answer for every evaluation of the query. Both the object and query indexes are implemented as disk-based regular grids. The same authors in Mokbel et al. (2004b) present a continuous query processor that extends a relational database management system and a data stream management system, to support efficient continuous query processing of spatiotemporal streams. They implemented the proposed query processor inside the PLACE (pervasive location-aware computing environments), scalable location-aware database server. Based on the authors' previous work, the proposed continuous query processor provides incremental evaluation of continuous queries, shared and scalable execution of a set of concurrent continuous queries and integration of data streaming management and semantics to support location-aware environments.

Hu et al. (2005) propose a generic framework for monitoring continuous spatial queries over moving objects, both range and k -NN queries. The work of Tao et al. (2002) focuses on continuous k -NN query evaluation, for moving queries over stationary objects. They propose an algorithm for precalculating k nearest neighbors with a line segment representing the continuous motion of the object.

Koudas et al. (2004), propose a method, called DISC, for approximate processing of k -NN queries over streams of multidimensional points, where the returned k th NN point is further than the actual k th NN point, within a specified distance threshold. Yu et al. (2005) describe a method for continuous monitoring of k -NN queries. Their method uses a main memory grid as an index structure and utilizes two algorithms using grid indices. The first one is based on object indexing, and the second is based on query indexing. Each k -NN query is evaluated for the first time by a two-step NN search technique, and is further re-evaluated every T time units. The initial step visits the cells in squares around the cell covering the query point until k objects are found. The second step refines the search by examination the cells outside the examined squares in order to determine the actual k -NN set of objects and remove false candidates appearing in the initial step.

Xiong et al. (2005) propose the method SEA-CNN for monitoring changes in the k -NN set, assuming that the initial result of a query is available. Objects are indexed by a regular grid on the disk. The method defines the answer region of a k -NN query as the circle centered at the query point with radius the distance to the current k -th NN, and the cells that intersect the answer region hold information about

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/monitoring-tracking-moving-objects-mobile/17152

Related Content

Indexing Mobile Objects: An Overview of Contemporary Solutions

Panayiotis Bozanis (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications* (pp. 313-333).

www.irma-international.org/chapter/indexing-mobile-objects/26511

Physical Layer Security in Military Communications: A Three Levels Approach

Elias Yaacoub (2019). *International Journal of Mobile Computing and Multimedia Communications* (pp. 26-40).

www.irma-international.org/article/physical-layer-security-in-military-communications/241786

Mobile Data Offloading Using Opportunistic Communication and AP Deployment: A Case Study

Sanjit Kumar Dash, Sasmita Mishra and Jibitesh Mishra (2017). *International Journal of Mobile Computing and Multimedia Communications* (pp. 66-84).

www.irma-international.org/article/mobile-data-offloading-using-opportunistic-communication-and-ap-deployment/193260

Data Ownership: Legalities Concerning Wearable Technologies

Thora Knight (2021). *Privacy Concerns Surrounding Personal Information Sharing on Health and Fitness Mobile Apps* (pp. 60-82).

www.irma-international.org/chapter/data-ownership/261905

Mobile Caching for Location-Based Services

Jianliang Xu (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications* (pp. 3031-3039).

www.irma-international.org/chapter/mobile-caching-location-based-services/26710