Information Sciences 278 (2014) 187–205

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Efficient processing of *continuous min–max distance bounded query* with updates in road networks

Yuan-Ko Huang *, Lien-Fa Lin

Department of Information Communication, Kao-Yuan University, Kaohsiung County, Taiwan, ROC

ARTICLE INFO

Article history:
Received 11 December 2012
Received in revised form 4 March 2014
Accepted 11 March 2014
Available online 19 March 2014

Keywords:
Spatio-temporal query
Road network
Continuous min–max distance bounded query
Bounded object
Bounded objects updating mechanism

ABSTRACT

In the last decade, the research community focuses on the design of various methods in support of spatio-temporal queries in road networks. In this paper, we present a novel and important query, named the *continuous min–max distance bounded query* (CM^2DBQ). Given a moving query object q, a minimal distance d_m, and a maximal distance d_M, a CM^2DBQ retrieves the *bounded objects* whose road distances to q are within the range $[d_m, d_M]$ at each time instant. We propose two algorithms, named the *Continuous Within Query-based (CWQ-based) algorithm* and the *CM^2DBQ algorithm*, to efficiently determine the *bounded objects*. In addition, we design a mechanism, named the *bounded objects updating mechanism*, to rapidly evaluate the new query result when object updates occur. Extensive experiments using real road network dataset demonstrate the efficiency of the proposed algorithms and the usefulness of the *bounded objects updating mechanism*.

1. Introduction

With the rapid advances of wireless communication and positioning technologies in mobile systems, the acquisition of spatio-temporal data using mobile devices is becoming pervasive. Many applications, such as traffic control systems, geographical information systems, and location-aware advertisement can benefit from efficient processing of spatio-temporal queries. Early methods [10,12,21,20,25–27] proposed to efficiently process spatio-temporal queries focus exclusively on Euclidean spaces (i.e., the query results are determined based on the Euclidean distance between each moving object and the object issuing the spatio-temporal query). However, in most real-world applications, the movements of objects (e.g., cabs and pedestrians) are constrained to a transportation network. As a result, the distance between two objects should be computed based on the connectivity of the network rather than the two objects' locations so that the query results obtained from performing the early methods are not always useful. Recently, several studies [5,4,9,13,15,22,24] have investigated how to process the spatio-temporal queries in road networks, where the criterion for determining the query results is the shortest network distance (i.e., shortest path) between objects. However, the focus of these studies is on providing efficient algorithms to process the *K*-Nearest Neighbor (*KNN*) and range queries over moving objects.

We present a novel and important type of spatio-temporal queries, named the *continuous min–max distance bounded query* (or CM^2DBQ for short). Given a query object q moving in a road network, a minimal distance d_m, and a maximal distance d_M, a CM^2DBQ retrieves the objects whose road distances to q are within the range $[d_m, d_M]$ at each time instant. The objects satisfying the CM^2DBQ are named the *bounded objects*. The CM^2DBQ is a useful query that can be found in many fields and application domains. Two real-world examples of CM^2DBQ are presented as follows.

* Corresponding author. Tel.: +886 919629876.
E-mail addresses: huangyk@cc.kyu.edu.tw (Y.-K. Huang), lienfa@cc.kyu.edu.tw (L.-F. Lin).

http://dx.doi.org/10.1016/j.ins.2014.03.040
0020-0255/

- Consider a traveler who wants to know the public transports in the vicinity. In some cases, the traveler is not interested in the public transports closer to him/her because of their high price or poor quality. In this scenario, the CM^2DBQ can be used to find the better transports that are not far away from the traveler, by bounding them within the distance range $[d_m, d_M]$.
- Consider a set of road segments that exhibit traffic congestion. In order to effectively reduce heavy traffic in these road segments, the CM^2DBQ can be issued to find the vehicles (i.e., the *bounded objects*) outside the congested area so as to prevent them entering this area.

Our efforts in this paper are devoted to investigating the CM^2DBQ problem under the following three conditions: (1) all objects (including the query object) move continuously in a road network, (2) the distance between two objects is defined as the distance along the shortest path between them in the network, and (3) the query result for CM^2DBQ at each time instant would be completely determined. Let us use an example in Fig. 1 to illustrate the problem tackled in this paper, where a set of objects o_1 to o_5 moves in a road network which is represented as a graph consisting of nodes and edges. In this example, each object moves steadily towards the direction indicated by the corresponding arrow. For ease of illustration, the query object q is stationary (note that this is however not required in our method). Assume that a CM^2DBQ is issued to find within a certain period of time the *bounded objects* whose road distances to q are within the range $[d_m, d_M]$ (depicted as gray lines). As shown in Fig. 1(a), at time t_1 there is no object within $[d_m, d_M]$ (i.e., no *bounded object* exists). At time t_2, the distance of object o_1 to q is equal to the minimal distance d_m (as shown in Fig. 1(b)). It means that object o_1's distance is greater than d_m after time t_2 so that o_1 is a *bounded object*. As the distance of object o_2 to q is equal to the maximal distance d_M at time t_3 (as shown in Fig. 1(c)), o_2 also becomes a *bounded object* after t_3 because its distance is less than d_M. Finally, the query result consists of tuples $\langle [t_1, t_2], \{null\} \rangle, \langle [t_2, t_3], \{o_1\} \rangle, \langle [t_3, t_4], \{o_1, o_2\} \rangle, \ldots$, where each tuple $\langle [t_s, t_e], S_{BOs} \rangle$ represents that objects in the set S_{BOs} are the *bounded objects* within time interval $[t_s, t_e]$.

To efficiently process the CM^2DBQ over moving objects in road networks, we first address the problem of how to significantly reduce the overhead of representing the road distance between moving objects at each time instant. Although the road distance can be computed based on the Dijkstra's algorithm [6] or the A* algorithm [19] that have been shown to be simple and efficient for computing the road distance between stationary objects, recomputing the road distance whenever objects change locations would incur extremely high computational cost which makes the idea infeasible, especially for mobile environments in which objects move continuously. In order to greatly reduce the recomputation cost, we design a technique using the information about the relative speed of moving objects and the shortest path between network nodes. By exploiting this technique, the network distance between two objects at each time instant is simply represented as a linear function of time, and thus can be easily computed. Another major problem we need to tackle is to avoid repetitively processing snapshot query at each time instant. Let us consider again the example in Fig. 1, where S_{BOs} changes from $\{null\}$ to $\{o_1\}$ at time t_2 and then from $\{o_1\}$ to $\{o_1, o_2\}$ at time t_3. We term these time points at which S_{BOs} changes from one to another (e.g., t_2 and t_3) the *result changing time points*. An important characteristic is that the query result in between two consecutive *result changing time points* remain the same. Based on this characteristic, the problem of performing repetitive queries can be greatly reduced to finding the *result changing time points* and their corresponding query result. In this paper, we design two methods combined with the proposed distance model to determine *result changing time points* with corresponding *bounded objects*.

Moreover, providing a useful mechanism to maintain the query result is also crucial for efficiently processing the CM^2DBQ in road networks. Once an object changes its *moving direction* (that is, it reaches an intersection of the road network), an update reporting the new direction of this object would be issued to the server. In addition, due to varying traffic conditions such as jams and accidents on the road network, the *moving speed* of an object must also be affected and hence an object update occurs. Such an update incurred by the change of direction and/or speed can potentially outdate the previous query result. As a result, the server needs to first examine whether the update affects the query result and then evaluate the new result if necessary. To support real-time processing of object updates, we have taken into account the problem of efficiently maintaining the query result in developing the processing methods. Based on the proposed methods, the server can not only quickly determine which updates influence the query result but also rapidly evaluate the new result.

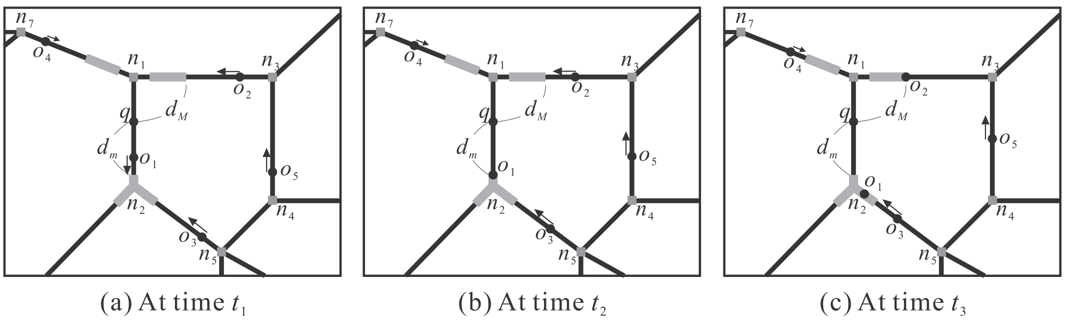

Fig. 1. An example of CM^2DBQ.