Materialization-Based Range and k-Nearest Neighbor Query Processing Algorithms*

Jae-Woo Chang and Yong-Ki Kim

Dept. of Computer Engineering, Chonbuk National Univ.,
Chonju, Chonbuk 561-756, South Korea
jwchang@chonbuk.ac.kr, ykkim@dblab.chonbuk.ac.kr

Abstract. Recently, the spatial network databases (SNDB) have been studied for emerging applications such as location-based services including mobile search and car navigation. In practice, objects, like cars and people with mobile phones, can usually move on an underlying network (road, railway, sidewalk, river, etc.), where the network distance is determined by the length of the practical shortest path connecting two objects. In this paper, we propose materialization-based query processing algorithms for typical spatial queries in SNDB, such as range search and k nearest neighbors (k-NN) search. By using a materialization-based technique with the shortest network distances of all the nodes on the network, the proposed query processing algorithms can reduce the computation time of the network distance as well as the number of disk I/Os required for accessing nodes. Thus, the proposed query processing algorithms improve the existing efficient k-NN (INE) and range search (RNE) algorithms proposed by Papadias et al. [1], respectively. It is shown that our range query processing algorithm achieves about up to one of magnitude better performance than RNE and our k-NN query processing algorithm achieves about up to 150% performance improvements over INE.

1 Introduction

In general, spatial databases has been well studied in the last two decades, resulting in the development of numerous spatial data models, query processing techniques, and index structures for spatial data [2]. Most of existing work considers Euclidean spaces, where the distance between two objects is determined by the ideal shortest path connecting them. However, in practice, objects, like cars and people with mobile phones, can usually move on an underlying network (road, railway, sidewalk, river, etc.), where the network distance is determined by the length of the practical shortest path connecting two objects on the network. For example, a gas station nearest to a given query q in Euclidean space may be more distant from q in a given network space than any other gas stations. Therefore, the network distance, rather than the Euclidean one, is an importance measure in spatial network databases. Recently, the

* This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) though the fostering project of the Lab of Excellency.

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 65 – 74, 2006.

spatial network databases (SNDB) have been studied for emerging applications such as location-based services including mobile search and car navigation. [3]. Studies on SNDB can also be divided into three research categories, that is, data model, query processing techniques, and index structures. First, Speicys et al. [4] dealt with a computational data model for spatial network. Secondly, Jensen et al. [5] presented k-nearest neighbor (k-NN) query processing algorithms for SNDB. Thirdly, Papadias et al. [1] proposed query processing algorithms for range search, spatial joins, and closest pairs as well as k-NN. Finally, Pfoser and Jensen [6] designed a novel index structure for SNDB. In this paper, we propose materialization-based query processing algorithms for typical spatial queries in SNDB, such as range and k-NN queries. By using a materialization-based technique with the shortest network distances of all the nodes in the spatial network, the proposed query processing algorithms can reduce the computation time of the network distance of two nodes as well as the number of disk I/Os accesses for visiting the nodes. Thus, the proposed query processing algorithms can improve the existing efficient k-NN and range search algorithms proposed by Papadias et al. [1]. This paper is organized as follows. In Section 2, we introduce related work on query processing algorithms for SNDB. In Section 3, we present the architecture of underlying storage and index structures for SNDB. In Section 4 and 5, we propose materialization-based range and k-NN query processing algorithms, respectively. In Section 6, we provide the performance analysis of our k-NN and range query processing algorithms. Finally, we draw our conclusions and suggest future work in Section 7.

2 Related Work

In this section, we overview related work on query processing algorithms for spatial network databases (SNDB). First, Jensen et al. described a general framework for k-NN queries on moving objects in road networks [5]. The framework includes a data model and a set of concrete algorithms needed for dealing with k-NN queries. The data model captures road networks and data points with continuously changing locations. It encompasses two data representations. The detailed two-dimensional representation captures the geographical coordinates of the roads and moving objects. The more abstract graph representation captures the road and moving objects in a form that enables k-NN queries to be answered efficiently by using road distances instead of Euclidean distance. The algorithms for k-NN queries employ a client-server architecture that partitions the NN search. First, a preliminary best-first search for a nearest-neighbor candidate (NNC) set in a graph is performed on the server. Secondly, the maintenance of the query result is done on the client, which re-computes distances between data points in the NNC set and the query point, sorts the distances, and refreshes the NNC set periodically to avoid significant imprecision. Finally, the combination of NNC search with the maintenance of an active result provides the user with an up-to-date query result.

Next, Papadias et al. proposed a flexible architecture for SNDB by separating the network from the entity datasets [1]. That is, they employ a disk-based network representation that preserves connectivity and location, while spatial entities are indexed by respective spatial access methods for supporting Euclidean queries and dynamic