Information Sciences 180 (2010) 2176–2195

Contents lists available at ScienceDirect

Information Sciences

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

Efficient mutual nearest neighbor query processing for moving object trajectories

Yunjun Gao [a,*], Baihua Zheng [b], Gencai Chen [a], Qing Li [c], Chun Chen [a], Gang Chen [a]

[a] College of Computer Science and Technology, Zhejiang University, 38 Zheda Road, Hangzhou 310027, China
[b] School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902, Singapore
[c] Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 26 August 2008
Received in revised form 2 February 2010
Accepted 5 February 2010

Keywords:
Query processing
Nearest neighbor query
Moving object trajectories
Algorithm

ABSTRACT

Given a set D of trajectories, a query object q, and a query time extent Γ, a *mutual* (i.e., *symmetric*) *nearest neighbor* (MNN) query over trajectories finds from D, the set of trajectories that are among the k_1 nearest neighbors (NNs) of q within Γ, and meanwhile, have q as one of their k_2 NNs. This type of queries is useful in many applications such as decision making, data mining, and pattern recognition, as it considers both the proximity of the trajectories to q and the proximity of q to the trajectories. In this paper, we first formalize MNN search and identify its characteristics, and then develop several algorithms for processing MNN queries efficiently. In particular, we investigate two classes of MNN queries, i.e., MNN_P and MNN_T queries, which are defined with respect to stationary query points and moving query trajectories, respectively. Our methods utilize the *batch processing* and *reusing technology* to reduce the I/O cost (i.e., number of node/page accesses) and CPU time significantly. In addition, we extend our techniques to tackle *historical continuous* MNN (HCMNN) search for moving object trajectories, which returns the mutual nearest neighbors of q (for a specified k_1 and k_2) *at any time instance* of Γ. Extensive experiments with real and synthetic datasets demonstrate the performance of our proposed algorithms in terms of efficiency and scalability.

A preliminary version of this work has been published in the Proceedings of the 9th International Conference on Mobile Data Management (MDM 2008). Substantial new technical materials have been added to this journal submission. Specifically, the paper extends the MDM 2008 paper by mainly including (i) processing of the HCMNN query with respect to stationary query points, called HCMNN_P retrieval (Section 5), (ii) processing of the HCMNN query with respect to moving query trajectories, called HCMNN_T search (Section 5), and (iii) enhanced experimental evaluation that incorporates the new classes of queries (Section 6).

Notes: (i) This manuscript is the authors' original work and has not been published nor has it been submitted simultaneously elsewhere, except for the preliminary version (i.e., [14]) mentioned previously; (ii) The main differences between the conference version and this submission are stated above; and (iii) All authors have checked the manuscript and have agreed to the submission.

1. Introduction

Trajectory data (e.g., animal movement data, vehicle positioning data, etc.) is very important in many applications. For example, by mining the movement trajectories of migrating birds, biologists can get better understanding of their migration

* Corresponding author. Tel.: +86 571 8765 1613; fax: +86 571 8795 1250.
E-mail addresses: gaoyj@zju.edu.cn, yjgao@smu.edu.sg (Y. Gao), bhzheng@smu.edu.sg (B. Zheng), chengc@zju.edu.cn (G. Chen), itqli@cityu.edu.hk (Q. Li), chenc@zju.edu.cn (C. Chen), cg@zju.edu.cn (G. Chen).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2010.02.010

patterns. For coaches or sports researchers, it is useful to study the movement patterns of top players by analyzing the trajectories of players' motions. In this paper, we introduce and solve a new type of queries, namely, *mutual nearest neighbor* (MNN) search for moving object trajectories. Given a trajectory set D, a query object q, a query time interval Γ, and two integers $k_1(\geqslant 1)$ and $k_2(\geqslant 1)$, an MNN query over trajectories retrieves all the trajectories $Tr \in D$, such that within Γ,Tr is among the k_1 nearest neighbors (NNs) of q; meanwhile, q is among the k_2 NNs to Tr. A special case is with $k_1 = k_2 = 1$, i.e., when q's nearest neighbor (NN) is Tr and Tr's NN is q, in which we say that q's MNN (with $k_1 = k_2 = 1$) is Tr.

Actually, MNN retrieval shares some similarities with the well-studied k-nearest neighbor (kNN) search on trajectory data [12,15,16], which finds the $k(\geqslant 1)$ nearest trajectories of a specified query object (point or trajectory) inside a predefined query time extent. Note that kNN retrieval over moving object trajectories is *asymmetric*, because it only considers the proximity of the trajectories to the query object, but not that of the query object to the trajectories. However, there are applications that may demand *symmetric* relationships. For example, carpooling services (e.g., http://www.erideshare.com/ and http://www.carpoolworld.com/) match people who share similar daily travel routes (i.e., trajectories). Ideally, carpooling members should live close by, and their work places must not be too far away from each other. For a new member p, a conventional kNN query can pair p with its closest member (denoted by m), but it ignores the fact that m might have travel routes closer to others, and hence would not like to share a car with p. In contrast, MNN search that takes into account the *symmetric* NN relationship can discover the member q having daily travel route close to that of p and, meanwhile, having p's daily travel route close to q's. Note that, if the travel route involves a relatively long time period, we can employ *historical continuous mutual nearest neighbor* (HCMNN) search (to be discussed later) that partitions the time duration into disjoint periods, and applies MNN retrieval for each interval. Consequently, MNN/HCMNN search may be used for the organization of carpooling by analyzing its query results. Moreover, an efficient solution to MNN/HCMNN retrieval can also contribute to solve such data mining problems as discovering *flock* [21] (i.e., a group of objects that exhibit similar movement and proximity for a certain amount of time) or *convoy* [25] (i.e., a group of objects that have traveled together for some time) patterns. The identification of groups of vehicles that stay close to each other for some extensive periods of time via using MNN/HCMNN search may also be useful to the prevention of traffic jams. In sum, MNN/HCMNN retrieval is more suitable, compared with kNN search, for those applications involving symmetric NN relationships, including data clustering [6,10,19,22,56], outlier detection [6,26], decision making [14,17,52], and pattern recognition [20,39]. More recently, we have explored MNN search in the context of spatial databases, which is practically very useful in many decision support applications (e.g., resource allocation, matchmaking, etc.) involving spatial data [17].

In this paper, we focus on MNN query processing for moving object trajectories. To the best of our knowledge, this paper is the first piece of work to address this problem. It is worth noting that, we are interested in both spatial and temporal components, but not the *movement shapes*, of trajectories.[1] Intuitively, a naive solution to handle MNN search on trajectory data is to find a set of k_1 NNs of a specified query object q within a given time interval Γ, denoted by $NN_{k1}(q)$, and then validate each trajectory $Tr \in NN_{k1}(q)$ by checking whether Tr has q as one of its k_2 NNs during Γ. Nevertheless, this method is very inefficient, since it incurs *multiple traversals* of the dataset, which results in high I/O overhead and CPU cost, especially when the values of k_1 and k_2 are large. To improve the performance, we propose, in this paper, an efficient MNN query processing algorithm. The basic idea of our approach is to employ *batch processing* and *reusing technology* to significantly reduce the number of node accesses (i.e., I/O cost), and hence speed up the search processing.

In addition, we extend our techniques to tackle *historical continuous mutual nearest neighbor* (HCMNN) retrieval over moving object trajectories. In particular, HCMNN search retrieves from a trajectory set D the mutual nearest neighbors (MNNs) of a specified query object q, for *any time instance* of a predefined temporal extent Γ. The output of an HCMNN query includes a collection of *nearest lists*, with each list containing a set of $\langle Tr, [t_i,t_j)\rangle$ tuples. Here, Tr is a trajectory in D (i.e., $Tr \in D$), and $[t_i,t_j)$ is the time interval (within Γ) during which Tr is an MNN of q for some given k_1 and k_2. HCMNN retrieval is actually a *continuous counterpart* of the MNN search. In this paper, we also present the algorithms for HCMNN query processing on moving object trajectories.

To sum up, the main contributions of this paper are as follows. First, we formalize the MNN and HCMNN search for moving object trajectories, respectively, and identify the characteristics of MNN queries. Second, we propose several algorithms to handle MNN search over moving object trajectories. More specifically, we investigate two types of MNN queries, termed as $\mathrm{MNN_P}$ and $\mathrm{MNN_T}$ search, which are defined with respect to stationary query points and moving query trajectories, respectively. In our study, we assume that the dataset is indexed by a TB-tree [38] due to its high efficiency in *trajectory-based queries*. Two search algorithms, namely, *Naive algorithm* (NI) and *reuse two-heap algorithm* (RT), are then developed for processing MNN queries. Third, we extend our approaches to process HCMNN queries on moving object trajectories, by considering both queries issued at stationary query points (i.e., $\mathrm{HCMNN_P}$) and those issued at moving query trajectories (i.e., $\mathrm{HCMNN_T}$). Finally, we conduct extensive experiments with both real and synthetic datasets to evaluate the efficiency and scalability of our proposed algorithms under a variety of settings.

The rest of the paper is organized as follows. Section 2 surveys related work. Section 3 formalizes MNN and HCMNN queries for moving object trajectories, and reveals the properties of MNN queries. In Sections 4 and 5, we describe the algorithms for handling $\mathrm{MNN_P}/\mathrm{MNN_T}$ queries and those for $\mathrm{HCMNN_P}/\mathrm{HCMNN_T}$ queries, respectively. Section 6 presents the performance

[1] Movement shapes of the trajectories are important to answer shapes-based trajectory queries [7,33,51,54].