Information Systems 44 (2014) 1–21

Contents lists available at ScienceDirect

Information Systems

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

Continuous visible k nearest neighbor query on moving objects

Yanqiu Wang [a], Rui Zhang [b], Chuanfei Xu [a], Jianzhong Qi [b], Yu Gu [a], Ge Yu [a,*]

[a] Department of Computer Software and Theory, Northeastern University, China
[b] Department of Computing and Information Systems, University of Melbourne, Australia

ARTICLE INFO

Article history:
Received 25 January 2013
Received in revised form
6 August 2013
Accepted 5 February 2014
Recommended by: Marian Amélie
Available online 22 February 2014

Keywords:
Spatio-temporal databases
Continuous visible k nearest neighbor queries
Safe region
Invisible time period

ABSTRACT

A visible k nearest neighbor (Vk NN) query retrieves k objects that are visible and nearest to the query object, where "visible" means that there is no obstacle between an object and the query object. Existing studies on the Vk NN query have focused on static data objects. In this paper we investigate how to process the query on moving objects continuously. We propose an effective filtering-and-refinement framework for evaluating this type of queries. We exploit spatial proximity and visibility properties between the query object and data objects to prune search space under this framework. A detailed cost analysis and a comprehensive experimental study are conducted on the proposed framework. The results validate the effectiveness of the pruning techniques and verify the efficiency of the proposed framework. The proposed framework outperforms a straightforward solution by an order of magnitude in terms of both communication and computation costs.

1. Introduction

The *visible k nearest neighbor* (Vk NN) query has attracted great research interest [12,13,20,21] recently due to emerging applications such as security camera placement and sightseeing site recommendation. This query assumes a set $\mathcal{P}$ of data objects, a set $\mathcal{O}$ of obstacles (represented by line segments) and a query object q. Then it retrieves k data objects from $\mathcal{P}$ that are visible and nearest to q. Fig. 1 gives an example. Suppose k=2. The data objects are listed according to their Euclidean distance to q as: $p_5, p_6, p_4, p_3, p_1, p_2$. Since p_5, p_4 and p_1 are blocked by the obstacles and invisible to q, they are not answers to the Vk NN query. The Vk NN set of q, denoted by $Vk\,NN(q)$, is $\{p_6, p_3\}$.

In this paper we study a continuous version of the Vk NN query, namely, the *continuous Vk NN query*, which computes the Vk NN from a set of moving objects for a moving query object continuously (i.e., for every timestamp).

The continuous Vk NN query has various applications. For example, in a military simulation, there can be more than 100,000 moving objects [37] such as soldiers and military vehicles interacting with each other. A soldier needs to keep track of his/her nearest visible enemies, so that he/she can attack or avoid them. As the soldier and the enemies are moving constantly, the simulator needs to monitor them continuously and report to the soldier his/her nearest visible enemies. In another example, massively multiplayer online first-person shooter (MMOFPS) games like CrossFire[1] need to show a player his/her nearby visible players so that he/she can shoot them. Again, as the players are moving constantly, the game system needs to monitor the players continuously and report to the player

* Corresponding author. Tel.: +86 2483683113.
E-mail addresses: wangyanqiu@ise.neu.edu.cn (Y. Wang), rui.zhang@unimelb.edu.au (R. Zhang), chuanfeixu@research.neu.edu.cn (C. Xu), jianzhong.qi@unimelb.edu.au (J. Qi), guyu@ise.neu.edu.cn (Y. Gu), yuge@ise.neu.edu.cn (G. Yu).

[1] http://crossfire.z8games.com/

http://dx.doi.org/10.1016/j.is.2014.02.003
0306-4379

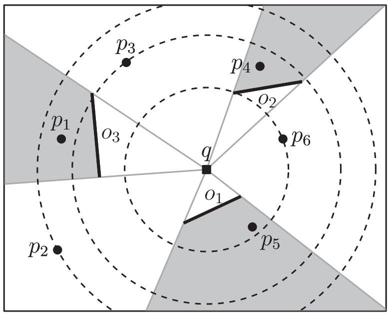

Fig. 1. V*k* NN$(q)=\{p_6,p_3,p_2\}$ $(k=2)$.

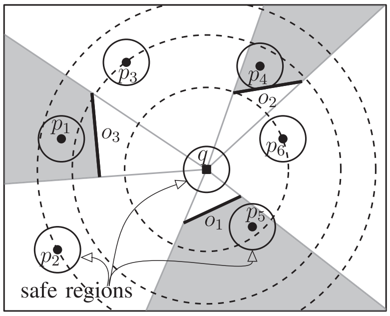

Fig. 2. Safe regions.

his/her nearest visible players. There may be millions of players [10] online at the same time. Therefore, a highly efficient algorithm is required to provide nearest-visible-player monitoring in real time.

The continuous V*k* NN query is interesting not only for its real applications but also for the technical challenges it raises. To the best of our knowledge, there is no existing work considering the query on moving objects. The main challenge here is that the query result needs to be up-to-date at every timestamp, which incurs significant communication and computation costs. To mitigate the costs, we propose a filtering-and-refinement query processing framework, and exploit spatial proximity and visibility properties between the query object and the data objects to prune search space under the framework. For spatial proximity based pruning, we use the *safe region*, which is a circular region centered at an object and is defined to bound the movement of the object for a certain period of time T (cf. Fig. 2). The safe regions that are close and visible to the query object further define a pruning region, which can be used to rule out objects that are too far away to be in the V*k* NN set within T timestamps. For objects that survive the safe region based pruning, their distance to the query object is not too far, but they may still be invisible to the query object due to obstacles. This motivates the visibility based pruning, which utilizes sub-periods within T that an object is invisible to the query object, so that the object can be excluded from the V*k* NN candidates during those sub-periods. We call such sub-periods the *invisible time periods*. All pruning techniques together keep the number of objects that pass the filtering stage small, and hence substantially reduce the costs of the refinement stage. As a result, we achieve a highly efficient query processing framework.

We summarize the contributions of this paper as follows.

- This is the first study that addresses the continuous V*k* NN query on moving objects. We propose a filtering-and-refinement framework that can process the query effectively.
- We develop two pruning strategies, namely, safe region based pruning and invisible time period based pruning, to reduce the search space for query processing under the proposed framework.
- We conduct a detailed cost analysis for the proposed pruning techniques. Extensive experiments using both real and synthetic data sets demonstrate the high efficiency of the pruning techniques as well as the proposed query processing framework.

The rest of this paper is structured as follows. We first review related work in Section 2. Then we formalize the continuous V*k* NN query on moving objects and present the filtering-and-refinement framework in Section 3. In Section 4, we present two pruning strategies under the framework and in Section 5 we provide a cost analysis for algorithms based on these pruning strategies. We report the experimental results in Section 6 and conclude the paper in Section 7.

2. Related work

We review three classes of related studies, namely, continuous spatial queries in general, continuous *k* nearest neighbor queries on moving objects and visible *k* nearest neighbor queries on static objects.

2.1. Continuous spatial queries in general

There is a large body of literature on continuous evaluation of spatial queries. For example, Šaltenis et al. [29] propose the Time Parameterized R-tree (TPR-tree) that indexes moving points as linear functions of time, based on which time-parameterized queries [26] are proposed to retrieve moving objects that satisfy certain time-parameterized predicates continuously. Ali et al. [1] study continuous retrieval of 3D objects using incremental computation to reduce the computational costs. Hu et al. [16] propose a safe region based framework for monitoring continuous spatial queries over moving objects on a client-sever based system. Each moving object (a client) in the system is aware of the current query result and only reports its new location to the server if it is likely to cause changes to the query result. Mokbel et al. [17,18] propose two frameworks for processing continuous spatial queries. They process the queries incrementally by computing the effect of each individual update on the query answer. They also propose shared execution techniques to process multiple queries at the same time. Benetis et al. [7] study the problem of continuous reverse nearest neighbor (RNN) monitoring over moving objects. Xia and Zhang [31] also study the continuous RNN query. They propose a so-called