Information Sciences 330 (2016) 274–292

Contents lists available at ScienceDirect

Information Sciences

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

Reverse k-nearest neighbor search in the presence of obstacles ☆

Yunjun Gao [a,b,*], Qing Liu [a], Xiaoye Miao [a], Jiacheng Yang [c]

[a] College of Computer Science, Zhejiang University, 38 Zheda Road, Hangzhou 310027, China
[b] The Lab of Big Data Intelligent Computing, Zhejiang University, Hangzhou 310027, China
[c] Department of Computer Science, Columbia University, New York, NY 10027, USA

ARTICLE INFO

Article history:
Received 9 October 2014
Revised 9 August 2015
Accepted 4 October 2015
Available online 17 October 2015

Keywords:
Reverse nearest neighbor
Obstructed reverse nearest neighbor
Obstacle
Query processing

ABSTRACT

In this paper, we study a new form of reverse nearest neighbor (RNN) queries, i.e., *obstructed reverse nearest neighbor* (ORNN) search. It considers the impact of obstacles on the *distance* between objects, which is ignored by the existing work on RNN retrieval. Given a data set P, an obstacle set O, and a query point q in a two-dimensional space, an ORNN query finds from P, all the points/objects that have q as their nearest neighbor, according to the *obstructed distance* metric, i.e., the length of the *shortest path* between two points without crossing any obstacle. We formalize ORNN search, develop effective *pruning heuristics* (via introducing a novel concept of *boundary region*), and propose efficient algorithms for ORNN query processing assuming that both P and O are indexed by traditional data-partitioning indexes (e.g., R-trees). In addition, several interesting variations of ORNN queries, namely, *obstructed reverse k-nearest neighbor* (ORkNN) *search*, *ORkNN search with maximum obstructed distance* δ (δ-ORkNN), and *constrained ORkNN* (CORkNN) *search*, have been introduced, and they can be tackled by extending the ORNN query techniques, which demonstrates the *flexibility* of the proposed ORNN query algorithm. Extensive experimental evaluation using both real and synthetic data sets verifies the effectiveness of pruning heuristics and the performance of algorithms, respectively.

1. Introduction

Given a multi-dimensional data set P and a query point q, a *reverse nearest neighbor* (RNN) query retrieves all the points in P that have q as their nearest neighbor (NN). Due to its wide application base such as decision support [15], profile-based marketing [15,26], and resource allocation [15,35], RNN is one of the most popular variants of NN queries [7,12,14,17,20]. Formally, $RNN(q) = \{p \in P \mid q \in NN(p)\}$, in which $RNN(q)$ represents the set of reverse nearest neighbors to q and $NN(p)$ denotes the NN of a point $p \in P$. Consider an example in Fig. 1a, where the data set P consists of three data points (i.e., p_1, p_2, p_3) in a two-dimensional (2D) space. Each point p_i ($1 \le i \le 3$) is associated with a vicinity circle/arc $cir(p_i, r)$ centered at p_i and having $r = dist(p_i, NN(p_i))$

☆ This paper is an extended version of the conference paper, titled "On Efficient Obstructed Reverse Nearest Neighbor Query Processing", which has been published in the Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2011), November 1–4, 2011, Chicago, IL, USA. Specifically, the paper extends the conference paper by including (i) additional three interesting variants of ORNN queries, i.e., ORkNN search (Section 7.1), δ-ORkNN retrieval (Section 7.2), and CORkNN search (Section 7.2); (ii) enhanced experimental evaluation that incorporates the new classes of queries (Section 8); and (iii) more complete and informative related work (Section 2), more pseudo-codes, more illustrative examples, and more analyzes. More details concerning this paper's extension have also been pointed out explicitly in Section 1 of the paper.

* Corresponding author at: College of Computer Science, Zhejiang University, 38 Zheda Road, Hangzhou 310027, China. Tel.: +86 571 8765 1613; fax: +86 571 8795 1250.

E-mail addresses: gaoyj@zju.edu.cn (Y. Gao), liuq@zju.edu.cn (Q. Liu), miaoxy@zju.edu.cn (X. Miao), jiachengy@cs.columbia.edu (J. Yang).

http://dx.doi.org/10.1016/j.ins.2015.10.022
0020-0255/

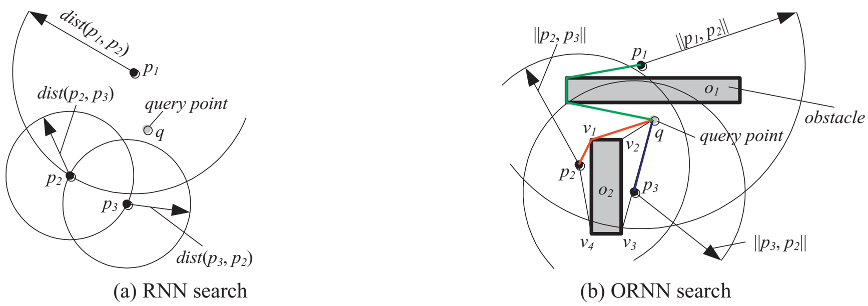

Fig. 1. Example of RNN and ORNN queries

as its radius, i.e., the vicinity circle/arc $cir(p_i, r)$ covers the NN of p_i. Here, $dist()$ refers to a specified distance metric. As shown in Fig. 1a, for the RNN query issued at a point q that uses the Euclidean distance as the distance metric, its result set $RNN(q) = \{p_1\}$, since q is only inside p_1's vicinity arc $cir(p_1, dist(p_1, p_2))$.

RNN search has been well studied, and many efficient algorithms have been proposed to support RNN query and its variants. A short review of some representative algorithms will be presented in Section 2.1. Existing algorithms employ either the *Euclidean distance* in a *Euclidean space* or the *network distance* in a *road network* to measure the proximity between objects. To the best of our knowledge, all those algorithms do not take into account the existence of *obstacles* (e.g., buildings and blindage). However, obstacles are *ubiquitous* in the real world, and their existence may change the *distance* between objects, and hence affect the final query result. Recently, the impact of obstacles on various research problems has attracted much attention from academy [10,11,13,19,23,32,34,39], and many work have been conducted, by taking the influence of obstacles into consideration. For example, the *spatial clustering in the presence of obstacles* (e.g., COD_CLARANS [29], DBRS_O [30], DBCLuC [38], etc.) is a new research direction for the data mining community formed by considering the impact of obstacles on spatial clustering.

In this paper, we study the impact of obstacles on RNN retrieval in a Euclidean space, and form a new type of RNN queries, namely, *obstructed reverse nearest neighbor* (ORNN) search. Given a data set P, an obstacle set O, and a query point q in a 2D space, an ORNN query finds from P, all the points that take q as their NN, according to the *obstructed distance*, i.e., the distance/length of the shortest path that connects two points without crossing any obstacle. An example is depicted in Fig. 1b, where $P = \{p_1, p_2, p_3\}$ and $O = \{o_1, o_2\}$. To simplify the discussion in this paper, we assume that obstacles are in rectangular shapes, although they could be in any other shape as well.

Let $||p_i, q||$ be the obstructed distance from a point p_i to q, and $ONN(p_i)$ be the *obstructed nearest neighbor* (ONN) of p_i that has the smallest obstructed distance to p_i compared with other points. We associate each point $p_i \in P$ with (i) an arc $arc(p_i, ||p_i, ONN(p_i)||)$ centered at p_i and with radius $||p_i, ONN(p_i)||$, and (ii) its obstructed path to q. For instance, the arc $arc(p_3, ||p_3, p_2||)$ centered at p_3 and having $||p_3, p_2||$ as the radius indicates that p_2 is the ONN to p_3, and the straight line from p_3 to q denotes the obstructed path between them without crossing any obstacle. It is observed that $||p_3, q|| < ||p_3, ONN(p_3) = p_2||$ and $||p_2, q|| < ||p_2, ONN(p_2) = p_3||$, and thus, q is the ONN to both p_2 and p_3, i.e., q's ORNN set $ORNN(q) = \{p_2, p_3\}$. Note that, p_1 is the RNN of q in a *Euclidean space* (see Fig. 1a), but it is not the ORNN of q in an *obstructed space* due to the block of obstacle o_1.

We focus on ORNN search because, it is not only a *challenging* problem from the research point of view, but also very *useful* in many applications. As an example, suppose KFC plans to open a new restaurant and wants to distribute coupons to its potential customers for promotion. Assume that there are some buildings and parks (i.e., obstacles) around the new restaurant, and customers who have the new restaurant as their obstructed nearest restaurant are more likely to visit. Consequently, in order to ensure the effectiveness of the promotion, KFC needs to identify the persons that take the new restaurant as their obstructed nearest restaurant, and distribute coupons to them. In addition, due to the ubiquity of obstacles, the ORNN query is obviously important, as a stand-alone tool or a stepping stone, in location-based services, geographic information systems, and complex spatial data analysis/mining involving obstacles.

In addition to the ORNN query, we also study several interesting variations, i.e., (1) *obstructed reverse k-nearest neighbor* (OR*k*NN) *search*, which retrieves all the points in the dataset P that take a given query point q as one of their obstructed k-nearest neighbors (O*k*NN); (2) *ORkNN retrieval with an obstructed distance threshold* δ (δ-OR*k*NN), which finds the OR*k*NN points that has the obstructed distances to q bounded by a pre-defined threshold δ; and (3) *constrained ORkNN* (COR*k*NN) *search*, which returns the OR*k*NN points in a specified restricted area (defined by the spatial region constraints).

In this paper, we present an efficient solution to tackle the ORNN query, which follows a *filter-refinement* framework and does not require any pre-processing. Moreover, we extend ORNN query algorithm to efficiently handle OR*k*NN, δ-OR*k*NN, and COR*k*NN queries, respectively. In brief, the key contributions of the paper are summarized as follows:

- We formalize ORNN search, a *new* addition to the family of spatial queries in the presence of obstacles.
- We introduce a *new* concept of *boundary region* to facilitate the pruning of unqualified data points and node entries.
- We develop efficient algorithms to answer *exact* or *approximate* ORNN retrieval.
- We extend ORNN query techniques to handle several variations of ORNN queries, i.e., OR*k*NN search, δ-OR*k*NN search, and COR*k*NN search.