International Journal of Innovative Computing, Information and Control
Volume **4**, Number **4**, April **2008**

ICIC International ©2008 ISSN 1349-4198
pp. 927–932

A METHOD OF NEAREST NEIGHBOR QUERIES FOR MOVING OBJECTS

Zhongcheng Yu and Hongtao Yu

College of Information Science and Engineering
Yanshan University
Qinhuangdao 066004, P. R. China
Yuzhongcheng1950@sina.com.cn

Received December 2006; revised August 2007

Abstract. *An indexing structure TPR tree (Time-Parameterized R-tree) and basic nearest neighbor queries are introduced in this paper. The concept of "influence time" is given and it is used in nearest neighbor queries algorithms to implement continuous nearest neighbor queries for moving objects, and finally the quality of the two continuous nearest neighbor query algorithms are tested and compared with extensive experiments.*
Keywords: Nearest neighbor query, TPR tree, Influence time, Algorithm, MBR

1. **Introduction.** Along with the rapid development of satellite position system (eg: GPS) and wireless communication technology, it becomes possible to trace and record the position of moving objects, and the continuous nearest neighbor queries for moving objects become the point of the research [1,2]. For example, "based on my current direction and speed of travel, which will be my two nearest gas stations for the next 5 minutes?"

In the static environment, two kinds of basic nearest neighbor query BaB (Branch-and-Bound) algorithms are DF (Depth-first) [6,7,15,16] and BF (Best-first) [8]. The first BaB algorithm is the way that searches for the nearest neighbors through a depth-first traversal. Specifically, starting from the root, all entries are sorted according to their *mindist* from the query point, and the entry with the lowest value is visited first (*mindist* is the minimum distance between the query object q and any object that can be in the sub-tree of entry E). The process is repeated recursively until the leaf level where the first potential nearest neighbor is found. During backtracking to the upper levels, the algorithm only visits entries whose *mindist* is smaller than the distance of the nearest neighbor already found.

BF algorithm keeps all the points that have been accessed by far by a heap. In the first place, the heap stores the son points of the root point only and puts them in order based on their *mindist* values. When a point is accessed and moved away from the heap, its son points will be added into the heap according to their *mindist* values. This process will last until the nearest neighbor is found. BF, a relatively advanced algorithm, which selects a point to be dealt with from all the points, accesses only the essential point and thus can control from the whole.

The two algorithms mentioned above, based on R-tree [9], are applicable for the nearest neighbor search in the static environment. In this paper, we introduce a new method that indexes dynamic object by the indexing structure of TPR-tree which can accomplish the continuous nearest neighbor queries effectively in the dynamic environment.