IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 4, 2019, accepted March 17, 2019, date of publication March 25, 2019, date of current version April 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907108

A Probabilistic Range Query of Moving Objects in Road Network

YAQING SHI[1], SONG HUANG[1], JUN FENG[2], AND JIAMIN LU[2]
[1]Command and Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China
[2]College of Computer and Information, Hohai University, Nanjing 210000, China

Corresponding author: Song Huang (hs0317@163.com)

This work was supported in part by the General Financial Grant from the China Postdoctoral Science Foundation under Grant 2016M603030, and in part by the National Natural Science Foundation of China under Grant 61602151.

ABSTRACT The range query of moving objects in a road network is widely used in battlefield environment, intelligent transportation systems, and mobile phone location. It has gradually become a research hotspot in the field of spatiotemporal data management. Most of the existing research on range query of moving objects in the road network is based on the premise of dense sampling of the device and complete storage of data. It ignores the uncertainty of discrete location data caused by a blind area of actual devices and limited memory, which cannot guarantee the accuracy of the range query. In this paper, the processing framework of probabilistic range query of moving objects in the road network is constructed. A spatiotemporal index structure is proposed based on the framework. This index can effectively represent time weights and the relationship between road sections. The probabilistic range query algorithm of moving objects in the road network with an uncertain trajectory caused by sampling frequency is designed and implemented. The experiments verify that the method proposed in this paper can improve query efficiency and ensure query accuracy.

INDEX TERMS Moving objects in road network, probabilistic range query, sampling frequency, trajectories uncertainty.

I. INTRODUCTION

With the continuous development of sensor technology, intelligent terminal technology and wireless communication technology, it is more and more convenient to obtain location data of moving objects in road network. In addition, the number of cars has increased dramatically in recent years, which leads to the explosive growth of location data of moving objects in road network. The query of moving object in road network has wide application prospects [1]. On the one hand, it can promote the development of real-time road monitoring, mobile navigation and road emergency management, and bring convenience to traffic travel. On the other hand, it can provide technical support in the form of basic services for upper-level situation and other operational information systems, which is of great significance for battlefield decision-making and information acquisition. Therefore, under the premise of massive data, how to achieve efficient and accurate query of moving object in road network has gradually become an important research topic in the field of spatio-temporal data management.

The range query of moving objects in road network [2], [3] is a typical query of moving objects in road network, which is widely used in intelligent transportation systems. At present, the popular taxi-hailing apps, such as Kuaidi-Taxi, Didi-Taxi and so on, is essentially to query moving objects within a specific time and space. Another example is establishing the target within the current time or the future time to carry out precise strike in combat. Businessmen push real-time electronic advertisements, such as preferential activities, to people in the vicinity of shops. With the increasing demand of the above applications, the range query has gradually developed into the probabilistic range query [4]–[6], the continuous range query [3], [7], [8], and the predictive range query [9]. Most of the existing research on the range query of moving objects in road network is based on the premise of dense sampling of device and complete storage of data. It ignores the uncertainty of discrete location data caused by blind area of actual devices and limited memory, which cannot guarantee the accuracy of range query. As shown in Fig. 1, the time value t is labeled in the figure, and the sampling frequency is 9 time units.

The associate editor coordinating the review of this manuscript and approving it for publication was Peng Shi.

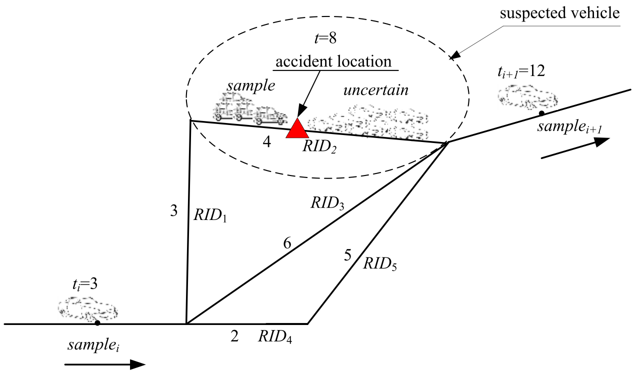


FIGURE 1. **Application example.**

Assuming that there is a traffic accident escape event in RID_2 at time $t = 8$, suspected vehicles need to be established now. *sample* $_{\mathrm{i}}$ is the sample at $t_{\mathrm{i}} = 3$ and $sample_{\mathrm{i}+1}$ is the sample at $t_{\mathrm{i}+1} = 12$. Are the vehicles suspected? Obviously, the locations of the vehicle between two samplers is uncertain at a certain time, so if the vehicle is on RID_2 at $t = 8$ time, it must have probability attributes.

The probabilistic range query of moving objects in road network: Given the query section RID, the query time t, the probability threshold α, the probabilistic range query $q(RID, t, \alpha)$ returns OID's set of all moving objects passing through RID, and probability value $P_{t,RID(OID)} \geq \alpha$ at t time.

In order to solve the above problems, this paper proposes a probabilistic range query of moving objects in road network. The main contributions include: The framework of probabilistic range query processing is constructed. It includes client, index cluster and HBase cluster. A spatio-temporal index structure is designed and implemented based on the framework, which can effectively represent time weights and the relationship between road sections. The spatio-temporal index structure combines UPA-tree (Uncertain Path-based on Assembly Method) with B$^+$-tree. A probabilistic range query algorithm of moving objects in road network with uncertain trajectory caused by sampling frequency is designed and implemented. It quickly establishes query candidate sets and solve repeated to improve query efficiency. Compared with the existing processing technology, this probabilistic range query of moving objects in road network can meet the real-time and accuracy requirements of intelligent transportation systems.

The remaining part of this paper is organized as follows. Section II introduces the related research work. Section III presents the processing framework of probabilistic range query of moving objects in road network. Section IV mainly describes the spatio-temporal index. Section V introduces the probabilistic range query algorithm. Section VI presents the experimental evaluations. Section VII mainly describes the conclusion.

II. RELATED WORK

As an important query type of moving objects database, the range query of moving objects in road network has been relatively mature on determined data [10], [11]. However, the range query [12], [13] of moving objects considering uncertainty is mostly concentrated in Euclidean space. Trajcevski *et al.* [14] propose a three-dimensional cylindrical probabilistic model for uncertain trajectories of moving objects. Various spatio-temporal operations to solve uncertainties are given, which efficiently support all kinds of static spatio-temporal range queries. Cheng *et al.* [15] study the probabilistic evaluation of uncertain data, uses multidimensional Probability Density Function (PDF) and time interval PDF to determine the possible world semantics of static data, and conducts range query research. Hua *et al.* [16] propose four types of uncertain data range queries using three threshold parameters (range threshold, probability threshold and result set scale threshold). In order to effectively solve the above probabilistic range query, the abstract index structure PRist+ is given, and the corresponding query algorithms are proposed. Emrich *et al.* [12] model the uncertain trajectories of moving objects in a random processing to support different probabilistic spatial-temporal queries. Aiming at the uncertainty of query location, Zhang *et al.* [17] propose a method based on filtering-and-verification paradigm to solve the range aggregate query in multi-dimensional space effectively. Lian and Chen [18] propose a probabilistic inverse range query, which retrieves the probabilistic threshold of a given query object in an uncertain database. This paper presents relevant pruning methods and process the query of high-dimensional uncertain data. In addition, the paper [19]–[22] discusses the queries about the continuous nearest neighbor, k nearest neighbor, probabilistic nearest neighbor, restricted nearest neighbor and other nearest neighbor based on uncertain location of moving objects, which provide a reference for the probabilistic range query of moving objects. However, the location uncertainty of moving objects studied in the above literature is based on Euclidean distance, that is, the linear distance between two points. The relevant research results cannot be directly applied to the road network.

There are a few studies about uncertainty in road network, and most of them are inconsistent with the uncertainty studied in this paper. Kuijpers and Othman [23] propose a road network uncertain trajectory model spatio-temporal prism based on Euclidean space. In this model, road network has maximum speed, but it only involves one type query, and the complexity of other queries is relatively high. Ming and Jan [24] consider the spatial query under the assumption of uncertain weights of network edges in road network. The above uncertainties are all based on the accuracy of positioning devices, positioning technology, network delay and network edge weight, which cause the location uncertainty of moving objects. This paper studies the location uncertainty of moving objects caused by the sampling frequency of positioning devices in road network. They are different in semantics, model and application background. As in the paper, the probabilistic range queries of location uncertainty caused by sampling frequency in road network are mainly literature [4] and literature [25], which assumes that the maximum speed of

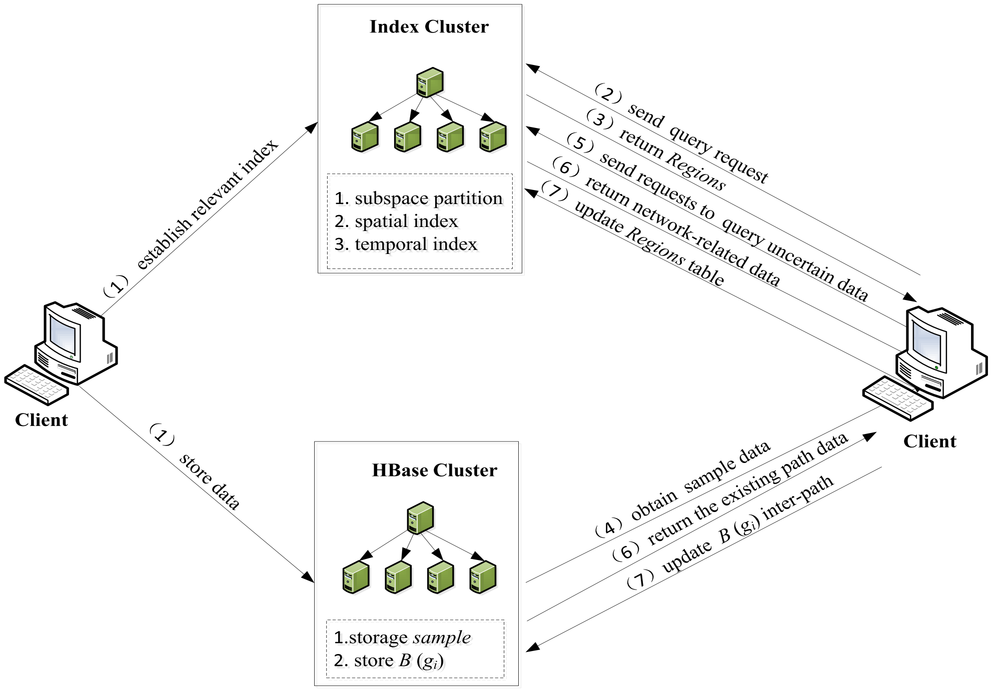


FIGURE 2. Processing framework.

each road network can be reached. Zheng *et al.* [4] consider that the moving object has its earliest arrival time and the latest departure time at both vertices of the road sections, the uncertainty of the moving object in road network can be expressed by the time-dependent probability distribution function. An indexing mechanism of the uncertain trajectory and an effective query algorithm are proposed to solve the spatio-temporal range query problem, although the probability calculation [4] is carried out in the query processing, the outstanding problem of this index method is that the track list of Uncertain Trajectories Hierarchy index (UTH) not only records the actual sample location, but also records the earliest arrival time and the latest arrival time of vertices in all possible paths of all moving objects on disk. In the index creation processing, frequent disk reading and writing are needed. The real-time processing of massive moving objects data in large-scale road network cannot be satisfied. Zheng *et al.* [25] also propose a Historical based Route Inference System (HRIS), which makes full use of the historical trajectory information of moving objects in road network to reduce uncertainty, but does not involve related queries. On the basis of literature [4], [25], Chen *et al.* [26] construct an uncertain trajectory model. It proposes a Partitioned Uncertain Trajectory Index PUTI to search for possible moving objects in specific time and space areas, which partitions according to the network distance of trajectory units of moving objects. However, the problem is that frequent insertion of uncertain trajectories in the index creation processing results in a huge burden on the system.

III. PROCESSING FRAMEWORK

Considering the magnitude of moving objects and the uncertainty of moving object trajectory between adjacent sample locations caused by sampling frequency, this paper designs a spatio-temporal index structure. The distributed database HBase [27] of open source platform Hadoop is used to store sampling data and part of path data, and the distributed file system HDFS [27] of MapReduce [28] is used to achieve high throughput data access, which ultimately achieves the purpose of efficient and accurate probabilistic range query. As shown in Figure 2, the processing framework gives the method of the probabilistic range query of moving objects in road network. The framework includes client, index cluster and HBase cluster. The client is mainly responsible for inserting data and sending query requests. Index cluster is mainly responsible for subspace partition, spatial index, and temporal index. HBase cluster is mainly responsible for the storage of the determined sample location data (samples) and the possible path between sub-graph boundary vertexes $B(g_i)$, g_i represents the ith subgraph of road network and corresponds to nodes in spatial dimension UPA-tree.

The system mainly involves two stages: data insertion and data query. In the data insertion stage, when new sample data arrives, the system sends it to both HBase cluster and index cluster. HBase cluster stores data in Region. Index cluster establishes relevant index for data (*step 1*). This way separates the index creation and maintenance from the data storage, and the former is no longer the burden of the latter. In the data query stage, the client sends a query request to the index

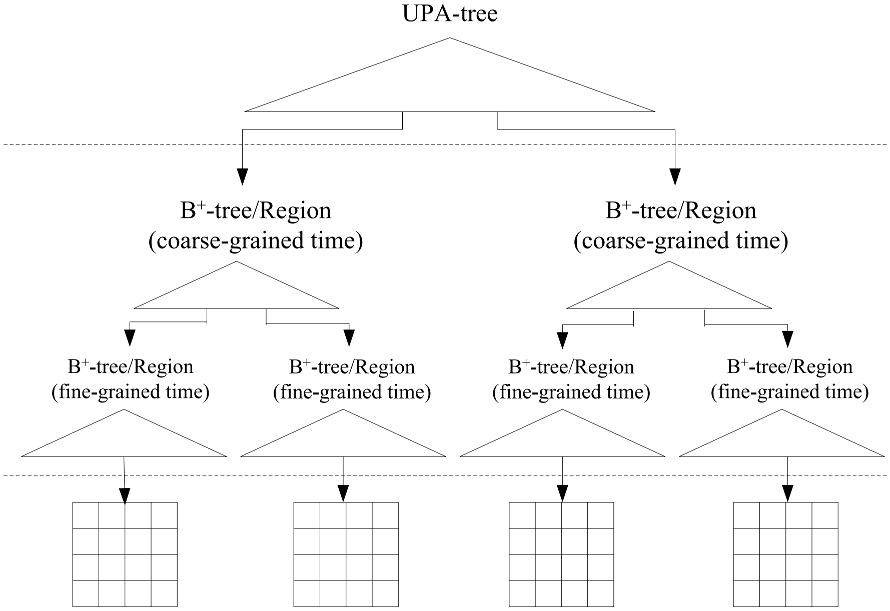


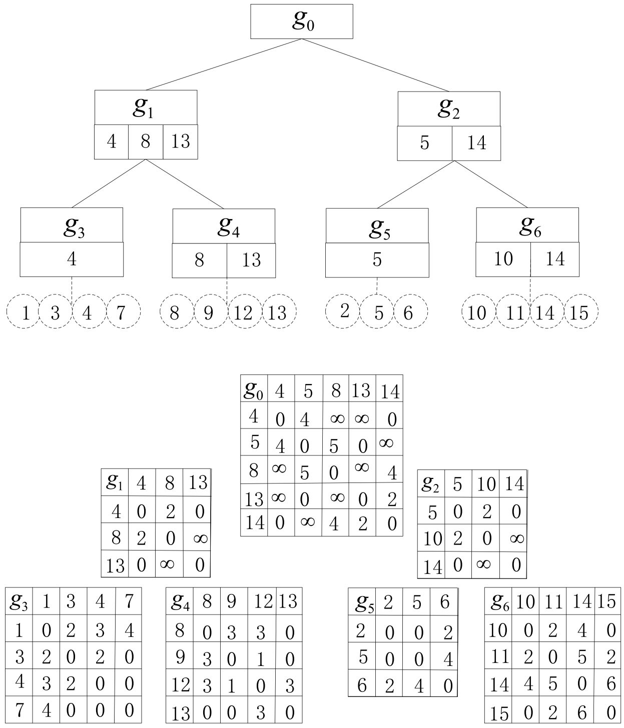
FIGURE 3. Index structure.

cluster (*step 2*). After receiving the request, the index cluster filters to get the Regions where the query data is located, and returns Regions to the client (*step 3*). The client obtains the determined sample data from the relevant Regions query (*step 4*). Clients continue to send requests to index clusters to query uncertain data between samples (*step 5*). Index clusters return network-related data of spatial dimension, such as link relationship, maximum time interval (*step 6*). The HBase cluster returns the existing path data between the boundary vertexes $B(g_i)$ of the subgraph g_i for the client to compute the uncertain data (*step 6*). In this processing, if there is new $B(g_i)$ inter-path, the Regions table in the corresponding index cluster is updated, and the $B(g_i)$ inter-path in the HBase cluster is updated synchronously (*step 7*).

IV. INDEX DESIGN

A. INDEX STRUCTURE

The index is divided into two layers. The first is the spatial dimension UPA-tree, and the second is the temporal dimension B$^+$-tree or the Region table established by time division. The leaf node of B$^+$-tree points to the determined location data of moving objects stored in HBase. The Region table records the Regions' ID which the queried possible path data are located. The index supporting probabilistic range query of moving objects in road network is shown in Figure 3.



g_0	4	5	8	13	14
4	0	4	∞	∞	0
5	4	0	5	0	∞
8	∞	5	0	∞	4
13	∞	0	∞	0	2
14	0	∞	4	2	0

g_1	4	8	13
4	0	2	0
8	2	0	∞
13	0	∞	0

g_2	5	10	14
5	0	2	0
10	2	0	∞
14	0	∞	0

g_3	1	3	4	7
1	0	2	3	4
3	2	0	2	0
4	3	2	0	0
7	4	0	0	0

g_4	8	9	12	13
8	0	3	3	0
9	3	0	1	0
12	3	1	0	3
13	0	0	3	0

g_5	2	5	6
2	0	0	2
5	0	0	4
6	2	4	0

g_6	10	11	14	15
10	0	2	4	0
11	2	0	5	2
14	4	5	0	6
15	0	2	6	0

FIGURE 4. UPA-tree.

B. SPATIAL DIMENSION

Based on G-tree [29] UPA-tree is design to handle the shortest path. It compares the time interval (t_{i+1}- t_i) between two adjacent samplings <*sample*$_i$, *sample*$_{i+1}$> and the shortest time t_m (ph_j) of the uncertain path between samplings. Only the possible path that meets $t_m(ph_j) \leq t_{i+1}$- t_i can be obtained, so as to solve the uncertain sections selection problem. Figure 4 is the UPA-tree index structure of the partition-based road network.

UPA-tree is a full binary tree, and the root node corresponds to graph G. All nodes except the root represent a subgraph. The parent node represents a subgraph which must be a hypergraph of its subnodes. Each node contains its corresponding subgraph g_i's boundary vertex set and the shortest time matrix. The row and column of the shortest time matrix of the leaf node are all vertices of the corresponding subgraph g_i, and the value of the matrix represents the shortest time of all the interior edges of the g_i. The row and column of the shortest time matrix of the intermediate node are all the boundary vertices of its sub-nodes, and the value is the shortest time for all the outer edges of the sub-nodes.

Regarding the shortest time matrix, the matrix values are uniformly stipulated as follows:

Firstly, in graph $G = (V, E)$, $\forall\, u, v \in V$, if $(u, v) \in E$, then $t_m\ (u,v)$ is the value of row u and column v in the shortest time matrix, that is to say, the sections are expressed with the beginning vertex of the row and the ending vertex of the column in the shortest time matrix.

Secondly, if the number of boundary vertices of a child's node is greater than 1, the shortest time of these boundary vertices in the shortest time matrix of the intermediate node is expressed as ∞, and ∞ can be set larger than all the shortest time in UPA-tree which is allowed by computer. As shown in Figure 4, g_2's child node g_6 has two boundary vertices v_{10} and v_{14}, so the shortest time value between v_{10} and v_{14} is set to ∞ in the matrix of g_2, indicating that v_{10} and v_{14} are boundary vertices of the same subgraph g_6, and there are inner edge connections. The specific value is in the corresponding node g_6's matrix of the subgraph.

Lastly, all cases except above are set to 0, indicating that there is no direct link between two vertices.

UPA-tree uses two Hash tables. One is used to store the corresponding relationship between *RID*, starting point v_s. and end point v_e, so as to locate vertexes of the segment quickly. The other is used to store the corresponding relationship between all vertices and partition subgraph, so that the UPA-tree leaf nodes where the vertices are located can be quickly determined in querying.

C. TEMPORAL DIMENSION

Considering the magnitude of trajectory data of moving objects in road network, the index only involves two kinds of data, one is sample data of determined sample location, and the other is possible paths between subgraph boundary vertices $B(g_i)$ recorded step by step in the query process. The temporal index only needs to index the time point of sample data. It is different with the paper [20], [21]. They need to store all the relevant sections of uncertain trajectories, and the temporal index object is the maximum time interval $[t_{ea}(v_s), t_{ld}(v_e)]$ of the moving object. Therefore, as shown in Figure 3, the temporal dimension uses B$^+$-tree, and each UPA-tree leaf node corresponds to a B$^+$-tree. According to the sampling time interval of the actual electronic sampling equipment in road network, two cases can be distinguished. One case is that the sampling time interval distribution is uniform in the general intelligent transportation system. The B$^+$-tree index based on time granularity is used to index the sampling time points, which can ensure the balance of B$^+$-tree. In the case of war or energy saving, the distribution of sampling time interval is not uniform. The traditional B$^+$-tree index based on sampling time points is directly adopted, which can also ensure the balance of B$^+$-tree.

In the spatio-temporal index supporting probabilistic range query of moving objects in road network, UPA-tree is directly combined with B$^+$-tree to index determined sample data in HBase Region. For possible paths between subgraph boundary vertices $B(g_i)$, UPA-tree is combined with Region table recording possible paths between boundary vertices. The purpose of indirectly indexing uncertain data is achieved by indexing the boundary vertices' possible paths between each node and their relation nodes, such as parent node, child nodes and sibling nodes. The possible paths are stored step by step in Region. Obviously, the size of the Region table gradually expands as queries proceed, but when all possible paths between boundary vertices in the space are recorded, the region table does not change again.

D. COMPLEXITY ANALYSIS

Complexity mainly discusses UPA-tree of spatial dimension. UPA-tree is a full binary tree, and its leaf nodes contain the number of vertices of the road network is χ, $\chi \geq 1$, and V is the total number of vertices of the road network. Thus, the high of UPA-tree is $H\ = \log_2(V/\chi) + 1$. The UPA-tree index is mainly composed of the nodes of the tree itself, the boundary vertices of each node representing the subgraph, and the shortest time matrix corresponding to each node. According to the relationship between binary tree nodes and tree height, the number of nodes of UPA-tree is $O\left(2^H - 1\right) = O\left(2^{\log_2(V/\chi)+1} - 1\right) = O(V/\chi + 1) = O(V/\chi)$. According to the Planar Separator Theorem proposed in paper [30], the number of boundary vertices is $O(\sqrt{V/2^{i-1}})$, so UPA-tree has boundary vertex $O(\sum\limits_{i=1}^{H} \sqrt{V/2^{i-1}})\ =\ O(V/\sqrt{\chi})$. Because the shortest time matrix rows and columns of all leaf nodes in UPA-tree are composed of vertices of corresponding subgraphs, the shortest time matrix size of all leaf nodes is $O(\chi.\chi.(V/\chi)) = O(\chi \cdot V)$, the shortest time matrix rows of intermediate nodes are all the boundary vertices of their corresponding child nodes, because each intermediate node on the ith level generates $O(\sqrt{V/2^i})$ boundary vertices, so the shortest time matrix size of the node is $O(V/2^i)$, because there are 2^i nodes on the layer i, so the matrix size of the layer i is $O(V)$, and the shortest time matrix size of the intermediate node is $O(\mathrm{H}.\ V) = O((\log_2(V/\chi) + 1).V)\ = O(\log_2(V/\chi).V)$. The spatial complexity of UPA-tree is $O(V/\chi + V/\sqrt{\chi} + log_2(V/\chi).V)\ = O(log_2(V/\chi).V)$.

V. QUERY ALGORITHM

A. QUERY ALGORITHM

According to the data nature, the probabilistic range query algorithm aims at determined data query and uncertain data query respectively. Determined data query does not involve path choice and probability calculation. It is relatively simple. Therefore, traditional spatio-temporal query method

Algorithm 1 Probabilistic Range Query Algorithm for Determined Data

```
Algorithm PPTRange_Query ($RID, t, \alpha$)
Input: Segment label $RID$, Query time $t$, Probability
       threshold $\alpha$
Output: Set of Moving Object $OID$s
1. Locate the UPA-tree leaf node of $RID$ according to Hash
   table.
2. Locate the $B^+$-tree root node, and locate the leaf node
   according to $t$.
3. Find the $sample_i$ at $t = t_i$.
4. IF $sample_i$ is located at $RID$.
5. Output $OID$.
6. END IF
7. FOR all other $OID$s except Step 3.
8. Find the sample pair of the same $OID$ with $t_i < t < t_{i+1}$.
9. END FOR
10.UPPTRange_Query ($RID, t, \alpha, < sample_i, sample_{i+1} >$).
End PPTRange_Query
```

is directly adopted. Leaf nodes are retrieved in UPA-tree according to the query section label *RID*, and corresponding B^+-tree root nodes are located. Then B^+-tree leaf nodes are found according to the query time t. The sample data is queried from the Region of HBase according to *OID* and *RowKey_TimeStam*p in B^+-tree leaf node items. *OID* are outputted at last. Probabilistic range query algorithm for determined data is as Algorithm 1(*step* 1- 6).

Considering the large amount of data and the complexity of the processing about uncertain data, Hadoop's MapReduce framework is used to process large amounts of spatial and temporal data in parallel. MapReduce-based probabilistic range query for uncertain data is as Algorithm 2.

The probabilistic range query processing for uncertain data includes Map and Reduce stages:

Map stage. The set of adjacent sampling pairs < *ID*, *samples*> is input. The MapReduce divides it into M fragments, corresponding to the same number of *Map* tasks. The input of each *Map* operation is the key-value pair <*ID*,*samples* >. *Map* operation calls the user-defined spatial pruning function according to the R-restrict to determine whether the *OID* of input sample pair may be in the query segment *RID* at the query time t. If it meets the requirement, the subdivision of the road network is determined according to the spatial location of the initial $sample_i$. The output of the *Map* operation is an intermediate key-value pair < *sub-partition ID*, *samples* >. Then the output data sets are sorted according to the *sub-partition ID*, and a new < *sub-partition ID*, *list*> tuple is generated. The purpose is to centralize the same initial $sample_i$'s queries in the same *sub-partition list*, and effectively use the calculated results to improve the query efficiency. These tuples are then divided into R fragments according to the *sub-partition ID*, corresponding to the number of *Reduce* tasks.

Algorithm 2 Probabilistic Range Query Algorithm for Uncertain Data

```
Algorithm UPPTRange_Query ($RID, t, \alpha, < sample_i, sample_{i+1} >$)
Input: Segment label $RID$, Query time $t$, Probability thresh-
old $\alpha$, Sample pair $< sample_i, sample_{i+1} >$ of the same $OID$
with $t_i < t < t_{i+1}$
Output: Set of Moving Object $OID$s
1. Divide samples to be queried into $M$ fragments
corresponding to $M$ Map tasks.
2. Call Map function to deal with space pruning and
   implement Map operation.
     Judge whether the $OID$ is likely to be in the $RID$ at $t$.
     Determine UPA-tree subdivision according to
     $sample_i$.
     Output < sub-partition ID, sample pair >.
        Sort the output set according to the sub-partition
     ID, and generate < sub-partition ID, list>.
3. Call Reduce function to process possible path query,
   probability pruning and probability calculation. Output
   the calculation results according to UPA-tree and
   Region.
4. Set the input and output paths and start MapReduce
   parallel operation.
5. Call the sub-query result merging program to merge all
   query results into complete results.
End UPPTRange_Query
```

Reduce stage. The input of each *Reduce* operation is <*sub-partition ID*,*list*>. *Reduce* operation calls possible path query, probability pruning and location probability calculation, and finally calls sub-query result merging program to merge all query results into complete results.

B. COMPLEXITY ANALYSIS

The probabilistic range query algorithms of moving objects in road network mainly focus on two kinds of queries, one is the determined sample data query, and the other is the uncertain objects query. For determined sample data queries, the time complexity is $O(n)$, n is the number of sample locations. For the uncertain objects query, MapReduce-based parallel processing is adopted. M and R are the number of *Map* and *Reduce* respectively. The computation of time complexity is divided into two parts. *Map* is aim to divide spatial and solve spatial pruning. In the worst case, spatial pruning calculates all sample pairs, i.e. $O(n^2)$, n is the total number of sample locations. *Reduce* is aim to possible path query, probabilistic pruning and location probability calculation. The key of time consumption is possible path query. The time complexity of possible path query is $O(1/\omega + \chi^2 + \log_2(V/\chi) \cdot V)$, ω is the sampling frequency, χ is the number of vertices in leaf nodes, and V is the number of vertices in road network.

The following conclusions can be drawn:

Firstly, the query time of probabilistic range query increases with the decrease of ω. The smaller the sampling frequency is, the larger the sampling time interval is. The larger the distance between two adjacent sample locations $<sample_i,\ sample_{i+1}>$ is, the more the possible path query time is, and the more the corresponding probabilistic range query time is.

Secondly, the probabilistic range query time increases with the increase of V. The more vertices there are, the larger the candidate vertex set is, the longer the query time is, and the more the corresponding probabilistic range query time is.

Thirdly, the query time of probabilistic range query first decreases and then increases with the increase of χ. Because the probabilistic range query mainly depends on the possible path query. When χ is relatively small, the possible path queries are mainly concentrated between different leaf nodes. The possible path queries between boundary vertices increase, so the overall query time is longer. With the increase of χ, query time will first have a decreasing trend, and then an increasing trend after reaching a certain value. This is because χ is so large that the query is transferred from the possible path query between boundary vertices to the query within leaf nodes, and the query time of the latter is relatively large. The probabilistic range query time keeps the same trend as the possible path query time.

Fourthly, the query time of probabilistic range query increases with the increase of n. The larger n is, the more time consumed to determined $<sample_i,\ sample_{i+1}>$ by spatially pruning *R-restrict*, the more time spent to the whole probabilistic range query.

Lastly, for parameters involving data set changes, such as n and V, the query time of probabilistic spatio-temporal range query increases linearly with the increase of these parameters. Hadoop can expand computing nodes dynamically when the data set is enlarged.That is to say, increase the value of M and R reasonably. The query time is effectively suppressed with the increase of M and R, and the query efficiency is significantly improved. This is also the main reason why Hadoop is used to solve the problem in this paper.

VI. EXPERIMENTAL ANALYSIS

The experiment mainly verifies the efficiency and accuracy of the probabilistic range query of moving objects proposed in this paper, and compares it with the existing processing technology [4], [26], in order to verify that the method proposed in this paper can improve the query efficiency and ensure the query accuracy at the same time.

The experimental Hadoop cluster consists of one NameNode, four DataNodes, processor Intel Core i5-2450M 2.5GHz dual-core, memory DDR3 4GB, hard disk WDC 500GB 7200R/Min 2MB cache, operating system Ubuntu Linux, JDK version jdk1.6, Hadoop version 1.0.4, HBase version 0.90.4. The data used in the experiment are divided into two parts: road network data and mobile vehicle data. The data of the road network is based on Colorado's traffic network, which has 43566 intersections and 1057066 sections. The paper uses mobile vehicle generator [31] to simulate and generate 10,000 vehicles on Colorado road network. The location information of these vehicles is continuously recorded at the same sampling time interval, generating 0.1, 1, 3, 5 and 10 million location records respectively.

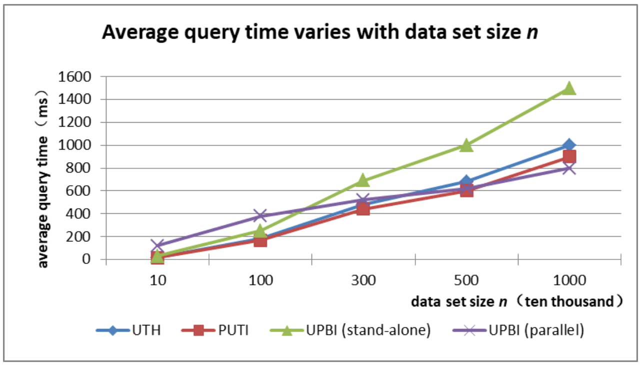


FIGURE 5. Average query time varies with data set size n.

A. QUERY TIME

1) DATA SET SIZE

The experiment is conducted to investigate the trend of average query time when the data set size$n$ changes. Fifty queries are made at different times of the same road section when different data sets are designed, and the average query time is compared. The query method (stand-alone and parallel) proposed in this paper is compared with the existing query technology based on UTH [4] and PUTI [26] index structure. As shown in Figure 5, the average query time increases with the size of the data set n. Parallel query method has higher query time than stand-alone query before the data set 2 million, but lower after 2 million. This is because data set segmentation and data transmission between nodes are time-consuming during the start-up of MapReduce. Later, the single query task is stable and the query time is restrained by dynamically adjusting the number of Map and Reduce tasks. The query technology based on UTH and PUTI are better than the query technology before the data set 4.5 million, and then the parallel query technology has obvious advantages. Because the uncertain trajectories of the former two have been acquired and indexed in the data insertion stage. It is possible to query directly through the index in the query stage. However, the query method need to deal with possible path queries and probability calculations in the query processing, they consume more time than the former two. However, with the increasing size of data, the computational complexity of the first two queries also shows a linear growth trend.

2) VERTICES NUMBER IN LEAF NODES OF SPATIAL INDEX

The experiment is carried out to investigate the trend of the average query time when the vertices number χ in leaf nodes of spatial index changes. The number of vertices is set to 32, 64, 128, 256 and 512, and the data set size is 0.1, 1 and 3 million respectively. The sampling interval is 180 seconds temporarily, and the probability threshold is 0.7. As shown in Figure 6, the query time decreases first and then

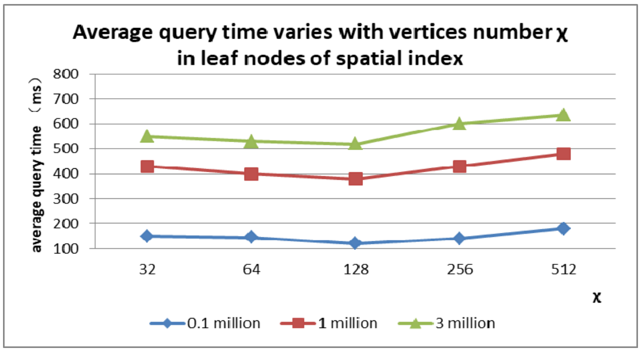


FIGURE 6. Average query time varies with vertices number χ in leaf nodes of spatial index.

increases with the increase of χ. Because when χ is small, the possible path query is concentrated between different leaf nodes. It is necessary to query the possible paths between adjacent layer boundary vertices from leaf nodes to the first common ancestor, and the query overhead is high. As the number of vertices increases, the query types of possible paths gradually change from different leaf nodes to the same leaf nodes. The breadth-first search is the main method at this time. When χ is large, the query time consumption is also large. When χ is 128 the same and different leaf nodes in the query are the best, so the number of vertices in subsequent experiments is 128.

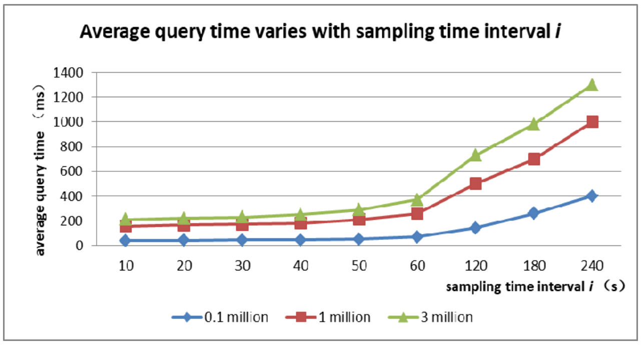


FIGURE 7. Average query time varies with sampling time interval.

3) SAMPLING TIME INTERVAL

The experiment mainly observes the change trend of average query time caused by the sampling time interval i. The sampling intervals are set to 10, 20, 30, 40, 50, 60, 120, 180 and 240 seconds respectively. The probability threshold is 0.7, and the number of vertices in leaf nodes is 128. As can be seen from Figure 7, the query time shows an increasing trend with the increase of sampling time interval, and the growth is flat before 50 seconds, and fast after 50 seconds. The reason is that the longer the sampling time interval, the more possible paths between two adjacent sampling points of the moving object, that is, the more possible locations of the moving object, the greater the uncertainty. This leads to the increase of uncertain queries in the query processing, and ultimately increases the query time. In the following experiments, the sampling time interval is 180 seconds considering the practical application and data storage.

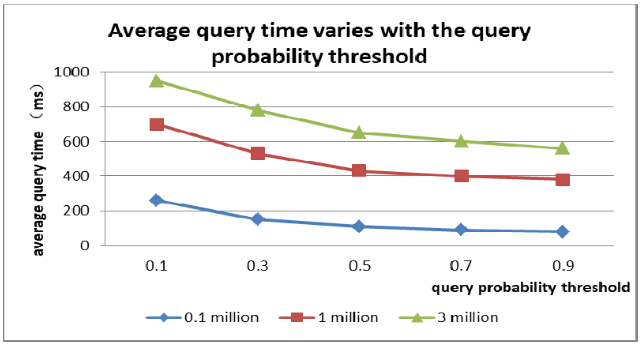


FIGURE 8. Average query time varies with the query probability threshold α.

4) PROBABILITY THRESHOLD

Probability threshold α in probabilistic range query involves location probability calculation and probability pruning, so it is also an important factor affecting query time consumption. The experiment investigates the effect of probability threshold on average query time. The sampling interval is 180 seconds and the number of vertices in leaf nodes is 128. Figure 8 shows that the query time decreases with the increase of probability threshold. The larger the query probability threshold α is, the greater the possibility of $\sum p(v_s, v_e) < \alpha$ is. $\sum p(v_s v_e) < \alpha$ is the probability sum of paths which are the possible paths of moving object OID between adjacent sampling $<sample_i, \quad sample_{i+1}>$ contains query segment RID. These moving objects can be discarded directly without calculating the exact location probability, which directly reduces the size of query candidate set. From Figure 8, we can see that the query time increases with the increase of data size. This is mainly due to the large data, which leads to the enlargement of the query candidate set as a whole, and the time for possible path query, probability pruning and location probability calculation also increases. Considering the decreasing trend of probability threshold at 0.7 in Figure 8, the probability threshold is 0.7 in subsequent experiments.

B. QUERY ACCURACY

Probabilistic range query accuracy comparison mainly involves query accuracy and query inverse accuracy. Query accuracy is expressed as $\#(Q^i_{obtain} \cap g^{10}_{real})/\#(Q^i_{obtain}) \times 100\%$, where Q^i_{obtain} is the probabilistic range query results with the sampling time interval i seconds. i is set 20, 40, 60, 120, 180 and 240 seconds respectively. g^{10}_{real} is the sampling results on the actual section every 10 seconds as a comparison benchmark. The query accuracy can reflect the percentage of the real value obtained by the query, but the percentage of the real value lost in the query processing cannot be expressed. It is expressed by query inverse accuracy $(\#(g^{10}_{real}) - \#(Q^i_{obtain} \cap g^{10}_{real}))/\#(Q^i_{obtain}) \times 100\%$. Figure 9 shows that the query

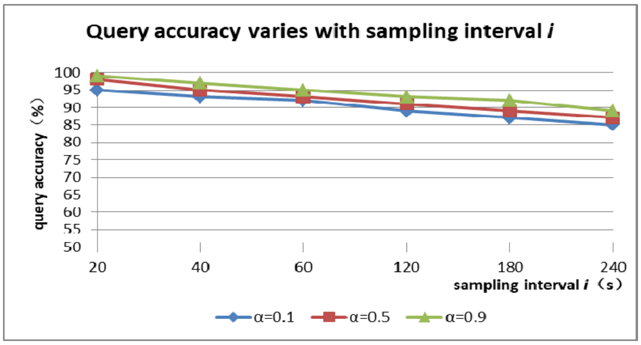


FIGURE 9. Query accuracy varies with sampling interval *i*.

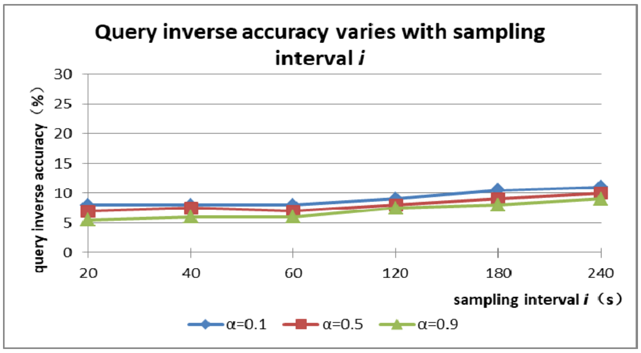


FIGURE 10. Query inverse accuracy varies with sampling interval *i*.

accuracy decreases with the increase of sampling interval. The accuracy of all sampling time interval queries is above 85%, the query time interval 180 seconds and the probability threshold 0.5 can reach more than 90%. Figure 10 shows that with the increase of sampling time interval, the query inverse accuracy increase, but the increase is not significant, and the overall query inverse accuracy value is between 6% and 11%. The larger the sampling time interval, the more possible paths between two adjacent sampling points for the same moving object, the greater the uncertainty of the trajectory, the lower the query accuracy and the higher the query inverse accuracy. Conversely, the shorter the sampling time interval, the smaller the uncertainty of the trajectory of moving objects, the higher the query accuracy and the lower the inverse accuracy. Figure 9 and Figure 10 show that the query meets the accuracy requirements and proves the effectiveness of the probabilistic range query algorithm.

VII. CONCLUSION

In the actual traffic network, the trajectory of moving objects is uncertain due to the low sampling frequency. In order to ensure the efficiency and accuracy of range query, a probabilistic range query processing framework including client, index cluster and HBase cluster is designed and implemented. This paper constructs a multi-dimensional index structure. UPA-tree is used in spatial dimension to process non-Euclidean spatial index, and B^+-tree or Region table established by time division is used in temporal dimension. On this basis, determined data range query algorithm and uncertain data probabilistic range query algorithm are designed and implemented. The performance of the proposed probabilistic range query algorithm is verified by theory and experiment. This paper can further study the load balancing and dynamic partitioning in distributed processing, and consider index construction and query processing in distributed environment.

REFERENCES

[1] J. Feng, L. X. Zhang, and J. M. Lu, "Review on moving objectsquery techniques in road network environment," *J. Softw.*, vol. 28, no. 6, pp. 1606–1628, 2017.

[2] J. Xu, H. Lu, and R. H. Güting, "Range queries on multi-attribute trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1206–1211, Jun. 2018.

[3] H. Wang and R. Zimmermann, "Processing of continuous location-based range queries on moving objects in road networks," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1065–1078, Jul. 2011.

[4] K. Zheng, G. Trajcevski, X. Zhou, and P. Scheuermann, "Probabilistic range queries for uncertain trajectories on road networks," in *Proc. ACM Int. Conf. Extending Database Technol.*, 2011, pp. 283–294.

[5] Z. J. Wang *et al.*, "SMe: Explicit & implicit constrained-space probabilistic threshold range queries for moving objects," *GeoInformatica*, vol. 20, no. 1, pp. 19–58, 2016.

[6] Y. Shi, J. Feng, Z. Ren, and W. Xie, "Hadoop-based probabilistic range queries of moving objects on road network," *Int. J. Smart Home*, vol. 10, no. 9, pp. 113–122, 2016.

[7] Z. Yu , F. Xhafa, Y. Chen, and K. Ma, "A distributed hybrid index for processing continuous range queries over moving objects," *Soft Comput.*, pp. 1–15, Dec. 2017. doi: 10.1007/s00500-017-2973-0.

[8] Y. Q. Shi, J. Feng, and Z. X. Tang, "UPBI—An efficient index for continues probabilistic range query," *Int. J. Multimedia Ubiquitous Eng.*, vol. 10, no. 5, pp. 355–372, 2015.

[9] X. Xu, L. Xiong, V. Sunderam, and Y. Xiao, "A Markov chain based pruning method for predictive range queries," in *Proc. ACM Sigspatial Int. Conf. Adv. Geograph. Inf. Syst.*, 2016, p. 16.

[10] S. Liu, L. Chen, and G. Chen, "Voronoi-based range query for trajectory data in spatial networks," in *Proc. SAC*, 2011, pp. 1022–1026.

[11] L. G. Xiang, D. H. Wang, and J. Y. Gong, "Organization and efficient range query of large trajectory data based on geohash," *Geomatics Inf. Sci. Wuhan Univ.*, vol. 42, no. 1, pp. 21–27, 2017.

[12] T. Emrich, H.-P. Kriegel, N. Mamoulis, M. Renz, and A. Zufle, "Querying uncertain spatio-temporal data," in *Proc. IEEE 28th Int. Conf. Data Eng.*, Apr. 2012, pp. 354–365.

[13] R. Cheng *et al.*, "Managing uncertainty in spatial and spatio-temporal data," in *Proc. ICDE*, Mar./Apr. 2014, pp. 1302–1305.

[14] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain, "Managing uncertainty in moving objects databases," *ACM Trans. Database Syst.*, vol. 29, no. 3, pp. 463–507, 2004.

[15] R. Cheng, D. V. Kalashnikov, and S. Prabhakar, "Evaluating probabilistic queries over imprecise data," in *Proc. ACM Sigmod Int. Conf. Manage. Data*, 2003, pp. 551–562.

[16] M. Hua, J. Pei, and X. Lin, "Ranking queries on uncertain data," *VLDB J.*, vol. 20, no. 1, pp. 129–153, 2011.

[17] Y. Zhang, X. Lin, Y. Tao, W. Zhang, and H. Wang, "Efficient computation of range aggregates against uncertain location-based queries," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 7, pp. 1244–1258, Jul. 2012.

[18] X. Lian and L. Chen, "Probabilistic inverse ranking queries in uncertain databases," *VLDB J.*, vol. 20, no. 1, pp. 107–127, 2011.

[19] G. Trajcevski, R. Tamassia, H. Ding, P. Scheuermann, and I. F. Cruz, "Continuous probabilistic nearest-neighbor queries for uncertain trajectories," in *Proc. 12th Int. Conf. Extending Database Technol.*, Saint-Petersburg, Russia, 2009, pp. 874–885.

[20] R. Cheng, J. Chen, M. Mokbel, and C.-Y. Chow, "Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 973–982.

[21] R. Cheng, L. Chen, J. Chen, and X. Xie, "Evaluating probability threshold *K*-nearest-neighbor queries over uncertain data," in *Proc. 12th Int. Conf. Extending Database Technol.*, Saint-Petersburg, Russia, 2009, pp. 672–683.

[22] Y.-K. Huang and Z.-H. He, "Processing continuous K-nearest skyline query with uncertainty in spatio-temporal databases," *J. Intell. Inf. Syst.*, vol. 45, no. 2, pp. 165–186, 2014.

[23] B. Kuijpers and W. Othman, "Modeling uncertainty of moving objects on road networks via space–time prisms," *Int. J. Geograph. Inf. Sci.*, vol. 23, no. 9, pp. 1095–1117, 2009.

[24] H. Ming and P. Jan, "Probabilistic path queries in road networks: Traffic uncertainty aware path selection," in *Proc. EDBT*, 2010, pp. 347–358, Lausanne, 2010.

[25] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories," in *Proc. ICDE*, Washington, DC, USA, Apr. 2012, pp. 1144–1155.

[26] L. Chen, Y. Tang, M. Lv, and G. Chen, "Partition-based range query for uncertain trajectories in road networks," *GeoInformatica*, vol. 19, no. 1, pp. 61–84, 2015.

[27] (Feb. 6, 2019). *Apache Software Foundation Project Home Page, HBase[EB/OL]*. Accessed: Feb. 6, 2019. [Online]. Available: http://hadoop.apache.org/HBase/

[28] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[29] R. Zhong, G. Li, K.-L. Tan, and L. Zhou, "G-Tree: An efficient index for KNN search on road networks," in *Proc. CIKM*, vol. 13, Oct. 2013, pp. 39–48.

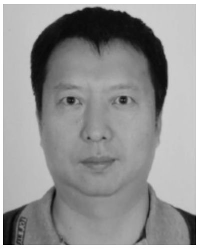
[30] R.-J. Lipton and R.-E. Tarjan, "A separator theorem for planar graphs," *SIAM J. Appl. Math.*, vol. 36, no. 2, pp. 177–189, 1979.

[31] C. Düntgen, T. Behr, and R. H. Güting, "BerlinMOD: A benchmark for moving object databases," *VLDB J.*, vol. 18, no. 6, pp. 1335–1368, 2009.

YAQING SHI was born in Liyang, Jiangsu, China, in 1981. She received the B.S. degree in computer science and technology from Xi'an Technological University, Xi'an, in 2004, and the M.S. and Ph.D. degrees in computer application technology from Hohai University, China, in 2007 and 2015, respectively.

She is currently an Associate Professor with the Command and Control Engineering College, Army Engineering University of PLA, Nanjing, China. Her research interests include spatiotemporal indexing and searching methods, and ITS.

Dr. Shi's awards and honors include the General Financial Grant from the China Postdoctoral Science Foundation (No. 2016M603030).



SONG HUANG was born in Huainan, Anhui, China, in 1970. He received the B.S., M.S., and Ph.D. degrees from the PLA University of Science and Technology.

He is currently a Professor of software engineering with the Software Testing and Evaluation Center, Army Engineering University of PLA. He has contributed more than 100 journal articles to professional journals. His current research interests include software testing, quality assurance, data mining, and empirical software engineering. He is a member of the advisory boards of the *Journal of Systems and Software* and the IEEE Transactions on Reliability.

JUN FENG was born in Xuzhou, Jiangsu, China, in 1969. She received the B.S. and M.S. degrees in computer science and technology from Hohai University, China, in 1991 and 1994, respectively, and the Ph.D. degree in information engineering from the University of Nagoya, Japan, in 2004.

She is currently a Professor with the College of Computer and Information, Hohai University. She has authored the book *Index and Query Methods in Road Networks* (Springer, 2015). Her research interests include data management, spatiotemporal indexing and search methods, ITS, and domain data mining.

JIAMIN LU was born in Nantong, Jiangsu, China, in 1983. He received the B.S. and M.S. degrees in computer science and technology from Hohai University, Nanjing, China, in 2004 and 2008, respectively, and the Ph.D. degree in information science from FernUniversität, Hagen, Germany, in 2014.

He is currently a Lecturer with the College of Computer and Information, Hohai University. His research interests include parallel processing on MOD, cloud infrastructure, and knowledge graph construction.