International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.5 (2015), pp.355-372
http://dx.doi.org/10.14257/ijmue.2015.10.5.34

UPBI: An Efficient Index for Continues Probabilistic Range Query of Moving Objects on Road Network

Yaqing Shi[1, 2], Jun Feng[2] and Zhixian Tang[2]

[1] *Command Information System Institute, PLA University of Science and Technology*
[2] *College of Computer and Information, Hohai University Nanjing, China*
yqshi_nanjing@163.com, fengjunhhu@gmail.com, hohaitangzx@gmail.com

Abstract

With the development of mobile terminal technology, the continuous range query of moving objects on road network has been widely applied in the field of transportation, military and communication. In practical applications, the sampling frequency of positioning equipment could not eliminate uncertainty, resulting in moving objects' position uncertainty between two adjacent samples. The index existing for continues probabilistic range query are based on the centralized or the traditional cluster distributed environment. In this paper, we construct UPBI index structure for the continuous probability range queries on road network based on Hadoop firstly. Secondly, we design the continuous probability range query parallel algorithm considering moving objects' position uncertainty on road network. Finally, we simultaneously give space constraint R-restrict and the probability calculation method. The experiment demonstrates that index and query algorithm proposed effectively solve the mass data problem about moving objects, and enhance query efficiency.

Keywords: *UPBI, continues probabilistic range query, uncertain data, sampling frequency, moving objects, road network*

1. Introduction

The range query of moving objects on road network is a typical query in Moving Objects Database (MOD). With the gradually increase of application requirements of Intelligent Transportation Systems, Command Information System and Mobile Communication System, the range query has been developing to the field of continuous range query and predicted range query. But the main problems of the existing range query of moving objects on road network are as follows: Firstly, with the growth of time, the position data of moving objects on road network has been showed a trend of mass, and distribute stored in each server in the network environment. Many ITS's queries are very high demand for real-time. Distributed storage and processing technology becomes more and more important. Secondly, the accuracy of range query is greatly affected by the sampling frequency. Most of existing researches suppose that high sampling can ignore position uncertainty of moving objects between two adjacent samples. Because of data storage, equipment blind, energy saving and equipment consumption, the equipment's sampling frequency couldn't exclude the influence of uncertain data in practical applications. The uncertain data must be obtained by the adjacent samples under the certain limit conditions. Lastly, the existing continuous range query usually adopts combining the snap-shot range query results in the continuous time period, the query error is large, and the query accuracy cannot be guaranteed. For these reasons, the continuous probabilistic range query of moving objects on road network has become a new hotspot in the field of moving object database.

ISSN: 1975-0080 IJMUE

In this paper, we mainly use Hadoop as the solution of mass storage and distributed computing framework because of its high efficiency, strong reliability and open-sourced. In summary, our main contributions are as following:

- A Hadoop-based index structure UPBI and a MapReduce-based index parallel create algorithm are proposed to index the certain sample data and obtain the uncertain data between two adjacent samples.
- A continues probabilistic range parallel query algorithm of moving objects on road network based on UPBI index is proposed to solve the uncertainty affected by the sampling frequency.
- A space constraint R-restrict and probability calculation method are proposed to reduce the query scope of the possible path and resolve the probability calculation problem.
- Experimental evaluation is conducted to demonstrate the benefits of the UPBI index and continues probabilistic range query algorithm proposed in this work.

2. Related Work

Range query of moving objects on road network as an important moving objects database query type is relative mature and has abundant research results. The literature [1-3] proposed range query algorithm, continuous range query algorithm and multi-level range query algorithm based on idea of Voronoi graph partitioning. Wang et al. [4] introduced a Shortest-Distance-based Tree (SD-Tree) to reduce the continuous query update cost. Abdeltawab *et al.* [5] proposed the iRoad framework which is equipped with reachability tree to scale up to handle real road networks with millions of nodes, and it can process heavy workloads on large numbers of moving objects. Liu *et al.* [6] gave the optimization method to discussed the dynamic range space under the network environment. Chen *et al.* [7] proposed a generic and adaptive updating protocol for moving object databases with less number of updates between objects and the database server. The premise of spatio-temporal range query on road network is assumed that the position of moving objects is certain, without considering the uncertain.

Range query of moving objects considering the uncertainty is more concentrated in the Euclidean space, and researches about uncertainties on the road network are based on the uncertainty of moving objects position information caused by the devices' precision, positioning technology, the network delay and the network edge weights. It is different with ours in semantic, model and application background. The main literature about this uncertainty as our paper is Zheng [8] and Chen *etc*. [9]. Zheng [8] considered moving objects having earliest arrival time and latest departure time at two vertices of the segment, and used probability distribution function depending on time to present the uncertainty. Chen *etc.* [9] established uncertain trajectory model. It partitioned the road network according to network distance of moving object trajectories unit. It proposed a partition-based uncertain trajectory index PUTI to search for the moving objects at specific time and specific region. But the same problem of those researches is that frequent uncertain trajectory inserting will cause huge burden to the system in the processing of index construction.

3. UPBI Index Structure

UPBI (UPA-tree and B^+-tree Index) index is divided into spatial index and temporal index. The spatial index adopts the UPA-tree structure for the road network. The temporal index adopts B^+-tree structure to index certain sample data stored in Hbase, and Region tables are established to index the Region ID which record the possible paths of boundary vertices of each node of UPA-tree. Figure 1 is the sample of the partitioned road network. UPBI index structure about it is shown in Figure 2:

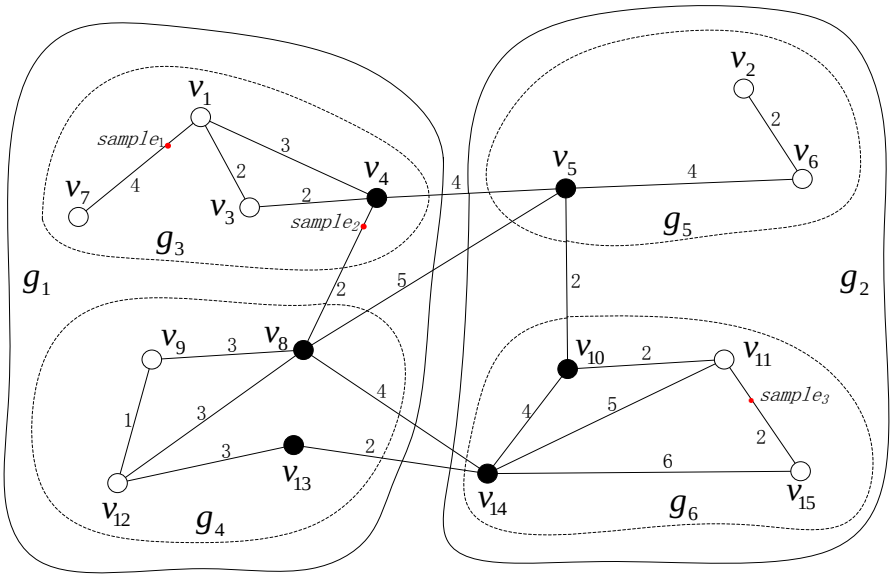

Figure 1. The Partitioned Road Network

Firstly, the road network is partitioned into k subgraph. Vertices of each subgraph keep in χ. UPA-tree is a full binary tree, and the keys that it can be used to quickly query the possible paths between two samples are the following two aspects: Firstly, each node of the tree records corresponding boundary vertices of subgraph. Secondly, each node contains a minimum time matrix, and the rows and columns of leaf nodes' matrix are all vertices of the subgraph. The rows and columns of intermediate nodes' matrix are the sum of boundary vertices of two child nodes. The value of matrix is the shortest time that the moving objects driving as the maximum speed on the road. It should be noted, if a cell of an intermediate node' matrix records its children's inner edge, then the cell is marked a maximum value, which means the shortest time value of the cell has been stored in the corresponding children's node matrix. If the cell doesn't exist then the shortest time is uniformly marked 0. For the certain sample position, UPA-tree directly indexes space by the subgraph represented using leaf node, while for uncertain position, it is resolved by nodes' boundary vertices of every layers, the shortest time matrix and the time constraints of samples.

Temporal index of UPBI only needs to index the time of the sample data. It is different than paper [8, 9] which need to index the max interval $[t_{ea}(v_s), t_{ld}(v_e)]$ of moving object on the segment in order to store the relevant segment of all uncertain paths. $t_{ea}(v_s)$ is v_s*'s* earliest arrival time in possible paths, and $t_{ld}(v_e)$ is v_e*'s* latest departure time. Therefore, as shown in Figure 2, temporal index of UPBI adopts B^+-tree index structure, and each leaf node of UPA-tree is corresponding to a B^+-tree index. The specific construction method of B^+-tree is divided into the uniform sampling and non-uniform sampling according to whether sampling interval of the actual road network electronic devices is consistent or not.

Region table indexes gradually stored possible paths between nodes' boundary vertices and its parent-child nodes or brother nodes. Obviously, the scale of Region table expands gradually with the queries, but when all possible paths between all boundary vertices in the space were recorded, Region table no longer changes again.

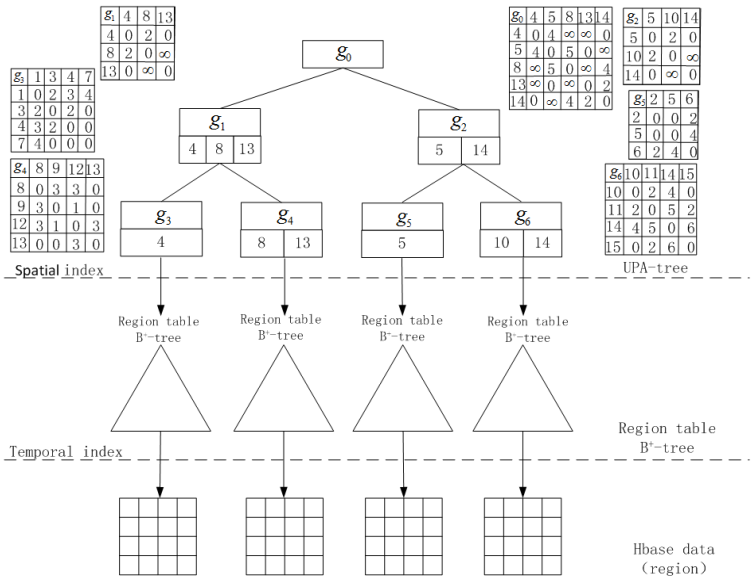

Figure 2. UPBI Index Structure

5. Continuous Probabilistic Range Query Processing

5.1 Query Algorithm

Continuous probabilistic range query does not distinguish certain sample data and uncertain data. It implements possible path and position probabilistic value calculation uniformly.The input set of algorithm here is no longer adjacent samples of moving objects but all the samples of each *OID*, and $sample_s$ is the latest sample which t_s is earlier than or equal to t_1 , $sample_e$ is the earliest sample which t_e is later than or equal to t_2. Continuous probabilistic range parallel query algorithm based on Hadoop is as follows:

Algorithm C_PPTRange_Query (*RID*, t_1, t_2, α, $sample_i$)

/* Input: *RID*: ID of the segment, $(t_2- t_1)$: query time period,

α: probabilistic threshold, $sample_i$: sampling points

Output: *OID*'s set of passing by *RID* probabilistic value is greater than α in $(t_2- t_1)$ */

1 Dividing $sample_i$ data set into *M* segments which are corresponding to *M* Map tasks;

2 Calling Map function to deal with time and **space constraint**:

For input $sample_i$, finding <$sample_s$, $sample_e$>;

For <$sample_s$, $sample_e$>, judging whether *OID* can be on *RID* in $(t_2- t_1)$ or not;

Judging the UPA-tree subdivision according to $sample_s$;

Outputing <subdivisionID, ($sample_s$, $sample_e$)>;

Sorting the output set according subdivisionID, generating <subdivisionID, *list*>;

3 Calling Reduce function to deal with path query and **probability calculation**:

Calculating and outputting results according to UPA-tree and Region table;

4 Setting the input and output paths, starting MapReduce parallel computing;

5 Calling merger program to merge all the sub-query results into a complete result;

End C_PPTRange_Query

Algorithm 1. Continuous probabilistic range parallel query algorithm

5.2 Space Constraint

Supposing v_i is any point between v_s and v_e on segment *RID*, t_i is any time between t_1 and t_2, $t_1 \le t_i \le t_2$, if *OID* of $<sample_s, sample_e>$ which meets $t_s \le t_1 < t_2 \le t_e$ has passed by *RID* in $(t_2 - t_1)$, then there must be t_i which makes *OID's* position on *RID* between two vertices of *RID*'s v_s and v_e, namely it is at v_i. Supposing the maximum interval between $sample_s$ and v_i is Δt, Δt must meet $\Delta t \le t_i - t_s$. We uniformly take actual city road maximum limit speed s_{max} =70km/h as the maximum speed of moving object. The road network distance r_1 between *samples* and v_i must meet $r_1 \le (t_i - t_s) \cdot s_{max}$, considering $t_1 \le t_i \le t_2$, so $r_1 \le (t_1 - t_s) \cdot s_{max}$. On the basis of the R-region proposed by Feng [10] in our research team, we can get the circular area R-region1which takes v_i as the center and $(t_1 - t_s) \cdot s_{max}$ as the radius. Similarly the maximum interval Δt between $sample_s$ and v_i must meet $\Delta t \le t_e - t_i$, so the road network distance r_2 between both meets $r_2 \le (t_e - t_i) \cdot s_{max}$, considering $t_1 \le t_i \le t_2$, so $r_2 \le (t_e - t_2) \cdot s_{max}$, and we can also get the circular area R-region2 which takes v_i as the center and $(t_e - t_2) \cdot s_{max}$ as the radius. Considering that the boundary of v_i are respectively v_s and v_e, so the center of the circle can be any point between v_s to v_e on *RID*. We push the center of R-region1 from v_s to v_e to form R-restrict1, and similarly push the center of R-region2 from v_s to v_e to form R-restrict2, as shown in Figure 3. So if $<sample_s, sample_e>$ is requested, then $sample_s$ and $sample_e$ must be in R-restrict1 and R-restrict2 respectively.

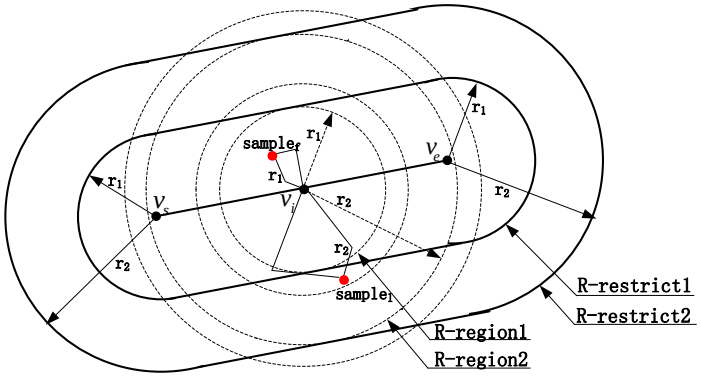

Figure 3. Space Constraint R-restrict

5.3 Probability Calculation

For all $<sample_s, sample_e>$ which meet $t_s \le t_1 < t_2 \le t_e$, the probability value calculation formula is provided for moving objects passing by given segment *RID* in $(t_2 - t_1)$ in continuous probabilistic range query:

$$P_{(t_2-t_1),RID}(OID) = \sum_{ph_k} p(v_s, v_e) \cdot p^k_{(t_2-t_1),RID}(OID)$$

$p(v_s, v_e)=1/|ph_k|$ is the probability value of query segment *RID* of possible path which meets time condition, $|ph_k|$ is the number of possible paths of $<sample_i, sample_{i+1}>$.

$p(v_s, v_e) \cdot p^k_{(t_2-t_1),RID}(OID)$ is the probability value of *OID* passing by *RID* of the path which meets t_m $(ph_k) \le t_e - t_s$ in query period $(t_2 - t_1)$. As shown in Figure 4, we respectively provide the earliest reach function and the latest departure function of segment *RID* on the basis of $t_{ea}(v_s)$, $t_{ea}(v_e)$ and $t_{ld}(v_s)$, $t_{ld}(v_e)$.

According to the intersection area of line t_1, line t_2, the earliest reach function and the latest departure function of segment *RID*, the calculation of $p^k_{(t_2-t_1),RID}(OID)$ can be divided into three cases:

(1)If $t_{ld}(v_s) \leq t_1 < t_2 \leq t_{ea}(v_e)$, then *OID* must be on the query segment *RID*, namely $p^k_{(t_2-t_1),RID}(OID) = 1$;

(2)If $t_2 \leq t_{ea}(v_s)$ or $t_1 \geq t_{ld}(v_e)$, then *OID* must not be on the query segment *RID*, namely $p^k_{(t_2-t_1),RID}(OID) = 0$;

(3)Except (1) and (2), *OID* may pass by query segment *RID*, and $0 < p^k_{(t_2-t_1),RID}(OID) < 1$.

For (3), as shown in the grid area in Figure 4, v_k is an intersection vertex on some possible path ph_k between $<sample_s, sample_e>$, R is the set of all possible positions of *OID* on the road network in $(t_2\text{-}t_1)$, namely the intersection area of line t_1, line t_2, the earliest reach function and the latest departure function of segment *RID*, R_{se} is the intersection of R, line v_s, line v_e, the earliest reach function and the latest departure function of segment *RID*. Obviously, R_{se} is the area of *OID* on *RID* in $(t_2\text{-}t_1)$, so $p^k_{(t_2-t_1),RID}(OID) = R_{se} / R$. When the query segment *RID* is the first segment as $sample_s$ or the last segment ending as $sample_e$, we need forward or backward construct the extension of $t_{ea}(v_k)$ function and $t_{ld}(v_k)$ function for probability calculation according to the minimum time t_m(e) of the before and after segment.

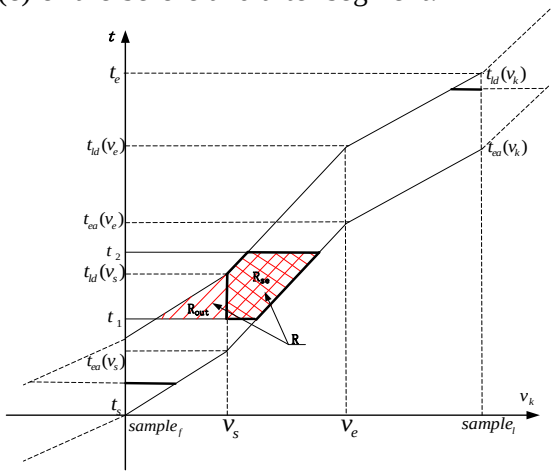

Figure 4. Position Probability Calculation Method

6. Experiment Evaluation

The road network data is the traffic network of U.S. Colorado [11] which has 435666 crossings, 1057066 segments. We use moving vehicle generator proposed in paper [12] to simulate 10000 vehicles on Colorado road network, and then continuously record the location of these vehicles with the same sampling interval, and finally obtain 0.1, 1, 3, 5 and 10 million records. We use four Datanodes and one Namenode, each node has Intel Core i5-2450M, 2.5 GHz dual-core processor running Ubuntu Linux OS with 4GB memory. All experiments are based on Hbase-0.90.4 and Hadoop-1.0.4, and are compiled and run using JAVA language.

6.1. Comparison Experiments of the UPBI Index Structure Performance

In the comparative experiment, make TPR^{uv}-tree [13] replace the UPA-tree in UPBI index structure, named TPBI index structure. The speed limit SL which edge information contains in direct accessing table of each leaf node in TPR^{uv}-tree changed into the minimum time $t_m(e)$. $t_m(e)$'s calculation is the same to UPA-tree index.

TPBI index create time is larger than the UPBI index in Figure 5. It is because UPBI can make the number of leaf nodes less by graph partitioning. With the network scale larger, the advantages will become more and more obvious. At the same time, when the data set is less than 4 million, the index create time of UPBI (parallel) was significantly higher than that of UPBI (single), the main reason is time consuming need to start the MapReduce tasks. When the data set is more than 4 million, the UPBI (single) index requires frequent node update operation, create time increases linearly, while UPBI (parallel) can dynamic adjust the number of Map and Reduce task, stable single task to suppress the index create time.

As show in Figure 6, UPBI index size is large than TPBI index. This is because UPBI index use the adjacency matrix and TPBI index use the adjacency list. But the gap is not great as UPBI using the graph partitioning.

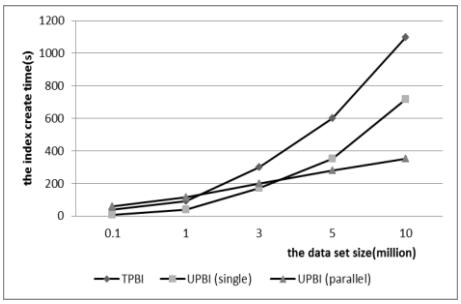

Figure 5. Effect of Date Set Size on the Index Create Time

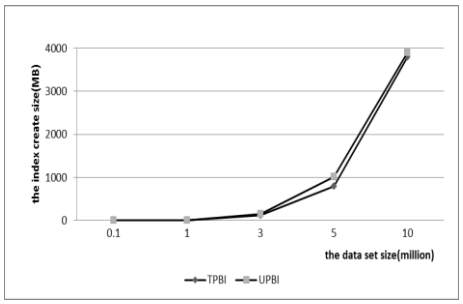

Figure 6. Effect of Date Set Size on the Index Size

6.2 Comparison Experiment of the Continuous Probability Range Query Performance

6.2.1. Date Set Size and Query Interval: Figure 7 shows that, UTH [8] and PUTI [9] query are better than the UPBI (single) query in time. Analysis of the reasons, mainly uncertain trajectory has been obtained in the index data insertion during the first two indexes, and another two indexes need the possible path query and the probability value calculation during the query processing. And the time consumption of UPBI (parallel) query is higher than that of UPBI (single) query when the date set size is less than 3 million, and is significantly lower than UPBI (single) when query data sets is more than 3 million. As shown in Figure 8, set the data set size is 7 million, the vertex number of leaf node is 128, the sampling time interval is 180 seconds, UPBI (parallel) query time is significantly less than the other three, and with the time continues growth, advantage is more apparent. This is because UPBI (parallel) query remain single query task scale stable to suppress the query time by dynamic adjustment of Map and Reduce task, at the same time it may be fully use of possible path between boundary vertex as the query time growth, but UPBI (parallel) query startup of MapReduce task requires more time.

6.2.2. Vertex Number in Leaf Node of Spatial Index: In Figure 9, the sampling interval is 180 seconds, the probability threshold is 0.7, and the query time with vertex number of leaf node in spatial index is decrease then increasing. Analyzing the reasons, when the number of vertices in leaf node is less, it needs to query the possible paths between boundary vertices in the adjacent layer from the leaf node to their first ancestors. With the vertices number increase, possible path query types change to the same leaf nodes, this is mainly the breadth first search. When the vertices number in the leaf nodes is larger, the query overhead is large too. We can see from Figure 9, when the number of vertices in leaf node is 128, the queries in the same and different leaf node optimal.

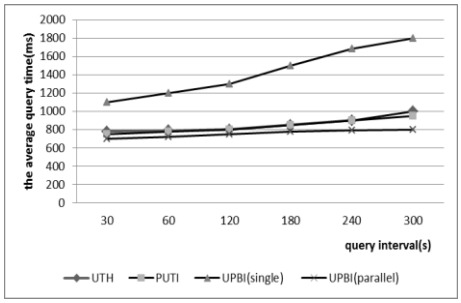

Figure 7. Effect of Date Set Size on the Average Query Time

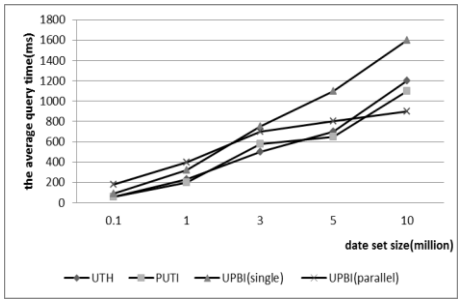

Figure 8. Effect of Query Interval on the Average Query Time

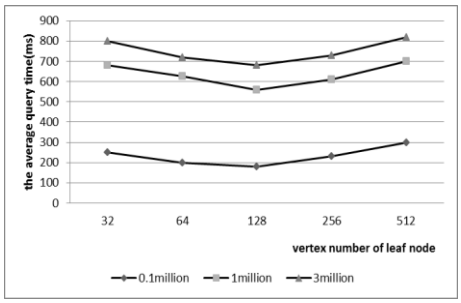

Figure 9. Effect of Vertex Number in Leaf Node on the Average Query Time

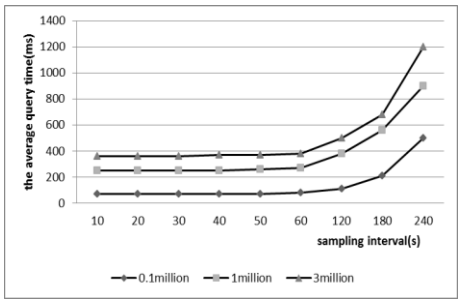

Figure 10. Effect of Sampling Interval on the Average Query Time

6.2.3. Sampling Interval: In Figure 10, the probability threshold is 0.7, the number of vertices in leaf node is 128. With the increase of the sampling interval, the query time increased, and before 50 seconds it increases gently, after 50 seconds it grows rapidly. Analyzing the reasons, when the sampling interval is longer, there are more possible path between two adjacent samples, namely moving objects' possible position may be more, the uncertainty is greater, and the probability range query time increases.

7. Conclusions and Future Work

In this paper, the uncertainty of the moving objects' position affected by the sampling frequency has been considered. An efficient UPBI index for continues probabilistic range query of moving objects on road network has been proposed. The experiment prove that index and query algorithm proposed effectively enhance query efficiency. In recent years, the spatial probabilistic temporal database (SPOT database) [14,15] attracted more and more attention. It will be considered the uncertainty of probability in the field of moving objects database (MOD).

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61370091), Natural Science Foundation of Jiangsu Province of China (No.BK2012059), and Advance Research Foundation of PLA University of Science and Technology (Research on Moving Object Indexing and Search Methods based on Spatial Networks).

References

[1] S. Liu, L. Chen and G. Chen, "Voronoi-based Range Query for Trajectory Data in Spatial Networks", Proceedings of the 2011 ACM Symposium on Applied Computing, (**2011**) March 21–24; Taiwan.
[2] K. Xuan, K. Zhao and D. Taniar, "Voronoi-based Range and Continuous Range Query Processing in Mobile Databases", Journal of Computer and System Sciences, vol. 77, no. 4), (**2011**), pp. 637–651.

[3] K. Xuan, G. Zhao, D. Taniar, M. Safar and B. Srinivasan, "Voronoi-based Multi-level Range Search in Mobile Navigation", Multimedia Tools and Applications, vol. 53, no. 2, (**2011**), pp.459-479.
[4] H. Wang and R. Zimmermann, "Processing of Continuous Location-Based Range Queries on Moving Objects in Road Network", Knowledge and Data Engineering, vol. 23, no. 7, (**2011**), pp.1065-1078.
[5] M. Hendawi, J. Bao and F. Mokbel, "iRoad: a Framework for Scalable Predictive Query Processing on Road Networks", Proceedings of the 39th International Conference on Very Large Data Bases, (**2013**), August; Italy.
[6] F. Y. Liu, T. T. Do and K. A. Hua, "Dynamic Range Query in SpatialNetwork Environments", Proceedings of the 17th International Congress of Database and Expert Systems Applications, (**2006**) September 4-8; Kraków, Poland.
[7] S. Chen, B. Ooi and Z. Zhang, "An adaptive updating protocol for reducing moving object database workload", Proceedings of the 36th International Conference on Very Large Data Bases, (**2010**) September 13-17; Singapore.
[8] K. Zheng, G. Trajcevski and X. Zhou, "Probabilistic Range Queries for Uncertain Trajectories on Road Networks", Proceedings of the 14th International Conference on Extending Database Technology, (**2011**) March 22-24; Uppsala, Sweden.
[9] L. Chen, Y. Tang, M. Lv and G. Chen, "Partition-based Range Query for Uncertain Trajectories in Road Networks", Geoinformatica, Published online: (**2014**) February 21.
[10] J. Feng, "A Study on Multi-scale/Multi-theme Map Information Model and Nearest Neighbor Search Method", Doctor paper, (**2004**).
[11] http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm
[12] C. Düntgen, T. Behr and R. H. Güting, "BerlinMOD: a Benchmark for Moving Object Databases", The VLDB Journal, (**2009**), vol. 18, pp. 1335-1368.
[13] P. Fan, G. Li, L. Yuan and Y. Li, "Vague Continuous K-nearest Neighbor Queries over Moving Objects with Uncertain Velocity in Road Networks", Information System, vol. 37, (**2012**), pp. 13-32.
[14] A. Parker, G. Infantes, J. Grant and V. S. Subrahmanian, "SPOT Databases: Efficient Consistency Checking and Optimistic Selection in Probabilistic Spatial Databases", IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 21, no. 1, (**2009**), pp. 92-107.
[15] J. Grant, C. Molinaro and F. Parisi, "Aggregate Count Queries in Probabilistic Spatio-temporal Databases", SUM 2013, LNAI 8078, (**2013**), pp. 255-268.

Authors

Yaqing Shi, she received the M.S degree in Computer Application Technology from Hohai University, China, in 2007. Now, she is a Ph.D. candidate in College of Computer and information, Hohai University. Her research interests include spatiotemporal indexing and searching methods and ITS. E-mail: yqshi_nanjing@163.com

Jun Feng, she received the Ph.D. degree in Information Engineering from University of Nagoya, Japan, 2004. She is currently a professor in College of Computer and information, Hohai University, Nanjing, China. She has been worked as a visiting scholar twice in University of Nagoya, from March 2005 to January 2006 and from December 2011 to February 2012, respectively. Her research interests include data management, spatiotemporal indexing and search methods, ITS and domain data mining. E-mail: fengjunhhu@gmail.com.

Zhixian Tang, he received the B.S. degree and M.S degree in Computer Science and Technology from Hohai University, China, in 2007 and 2010, respectively. Now, he is a Ph.D. candidate in College of Computer and information, Hohai University. His research interests include spatiotemporal indexing and searching methods. E-mail: hohaitangzx@gmail.com.