International Journal of Smart Home
Vol. 10, No. 9 (2016), pp. 113-122
http://dx.doi.org/10.14257/ijsh.2016.10.9.11

Hadoop-based Probabilistic Range Queries of Moving Objects on Road Network

Yaqing Shi[1], Jun Feng[2], Zhengping Ren[3] and Wenjuan Xie[4]

[1]*PLA University of Science and Technology Nanjing, China*
[2]*Hohai University Nanjing, China*
[3, 4]*PLA University of Science and Technology Nanjing, China*

Abstract

With the continuous development of wireless communication and mobile positioning technologies, spatio-temporal queries of moving objects attract more and more attention. In practical application, affected by the sampling frequency of the devices, the position information of moving objects restricted to the road network is often with uncertainty. In this paper, on the basis of the distributed computing framework-Hadoop, it firstly constructs the UPBI index mixing certain and uncertain data. Secondly, it proposes the probabilistic range parallel queries algorithm and the probabilistic calculating method of moving objects on road network. Finally, it gives space constraint r-Restrict to reduce the query scope of the possible path, and simultaneously gives sample pair division to resolve the problem of repetitive calculation. The experiment proves that index and query algorithm proposed effectively solve the mass data problem about moving objects, and enhance query efficiency and precision.

***Keywords**: Hadoop; moving objects on road network; probabilistic range queries; uncertain data; sampling frequency*

1. Introduction

With the development of application, such as Geographic Information System, Mobile Communication System, Intelligent Transportation Systems and Location-Based Services, Moving Objects Databases (MOD) [1] has gradually become a hot research area in the field of spatio-temporal database. The space limited network like road network restricts moving objects can only move along the set direction and path. The location-based query of moving objects on road network, such as the range queries, can support ITS decision. Therefore, it has attracted more and more attention of researchers. Because the position information of moving objects on road network is collected by GPS or roadside sensor devices, the precision of related equipment, positioning technology and network delay could also cause the position information obtained with uncertainty. But there is another uncertainty widespread in moving objects on road network. The prerequisite is that all values of the moving object provided by the positioning devices are assumed to be available [2-7]. The positioning devices provide discrete time-based position value. The uncertainty of position between the values is caused by the low sampling frequency of positioning device. The sampling frequency is lower, and the uncertainty of position of the moving objects is greater. According to the analysis, we know that there are two kinds sampling. One kind is a uniform sampling frequency, and another is non-uniform sampling frequency because of the non-uniformity of damaged devices.

For the range queries of moving objects on road network about the two application examples, the traditional approach is clearly only for certain data, and loses uncertain data matching the query. The precision can't be guaranteed. As time going by, the moving objects' position continuous change. It causes sharp incensement of certain and uncertain

ISSN: 1975-4094 IJSH

data, and ultimately leads to sharp decline in the performance of the index and query. Therefore, this paper introduces distributed computing framework-Hadoop to the probabilistic range queries of moving objects on road network. The key is to solve the massive and uncertainty of data about moving objects.

The major contributions of this paper are summarized as follows:

- An index structure UPBI is proposed to efficiently index the certain data and help to obtain the uncertain data by indexing road network and possible paths between boundary vertices.
- A Hadoop-based probabilistic range parallel queries algorithm about uncertain data of moving object on road network is proposed. It effectively solves the mass data problem and position uncertainty problem of moving object. The query is also separated for the certain and uncertain data to ensure the precision of the query.
- A space constraint r-Restrict and sample pair division are proposed to reduce the query scope of the possible path and resolve the repetitive calculation problem.
- Simulation experiments are conducted to evaluate the performance of the UPBI index and the probabilistic range queries algorithm.

2. Related Work

In recent years, more and more researches about uncertain location of the moving objects have been developed [8-14]. Of course, there are researchers try to do road network direction, Kuijpers *et al*. [15] propose the space-time prism in the road networks with general maximal speeds of edges based on the model in Euclidean space. However, there is only an alibi query provided in the paper, which might be because the complexity of other query types is high. Considering the network edge weights uncertain on the road network, Ming *et al*. [16] proposed three novel types of probabilistic path queries using basic probability principles. The position uncertainty about moving objects above mentioned is caused by the precision of positioning equipment, positioning technology, network delay and network edge weights. But this paper is to study the position uncertainty between the two successive samples caused by sampling frequency of positioning equipment. Both have the difference in semantics, model and application background. Zheng *et al*. [17] proposed a History based Path Inference System (HRIS). It made full use of historical track information of moving objects on road network, reduced the uncertainty, but not involved in related queries.

In recent years, researches about the probabilistic range queries with the same study background to us include Zheng *et al*. [18] and Chen *et al*. [19]. Both assumed availability of a maximal speed on each road segment. Zheng *et al*. [18] proposed making use of vertices' earliest arrival time and latest departure time. It represented the uncertainty of the objects moving along road networks as time-dependent probability distribution functions. It proposed an indexing mechanism Uncertain Trajectories Hierarchy (UTH) and the spatio-temporal range query algorithm. The probability calculation is carried out in the query process, but the Trajectory List records both the actual sample location and all vertices' earliest arrival time and latest departure time in all possible paths about all moving objects. It needs frequent disk read and write during index creation processing. It is not suitable for large-scale processing about mass of moving object. Based on the literature [18], Chen *et al*. [19] constructed an 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 problem with this approach is also that frequent uncertain trajectory insertion causes a great system burden during index creation processing.

3. Problem Formulation

DEFINITION 1 A road network is represented by a graph G=(V, E). Vertices' set V is corresponding to the crossings of road network. Edges' set E is corresponding to the segments of the road network. Each edge e ($e \in E$) is associated with an attribute vector $<l(e), s(e)>$, corresponding to the length and maximum allowed speed of e. $t_m(e)$ is the minimum time that moving objects to drive in e with the maximum limit speed, $t_m(e)$= $l(e)/s(e)$.

DEFINITION 2 Given graph G= (V, E), V and E are the set of vertices and edges, respectively. $\{g_1,g_2,\ldots g_k\}$ is a partition of Graph G, g_i=(V_i, E_i), $i=1,2,\ldots,k$, which satisfies:

a) $V = \bigcup V_i$;

b) if $i \neq j$, $v_i \cap v_j = \varnothing$;

c) $\forall u, v \in V_i$, if $(u, v) \in E$, then $(u, v) \in E_i$.

DEFINITION 3 Given g_i, which is the subgraph of graph G, vertice $u \in V_i$, and there is at least one edge $(u, v) \in E$, $v \notin V_i$, then u is the boundary vertice of subgraph g_i. All boundary vertices of subgraph g_i are represented as set $B(g_i)$.

DEFINITION 4 Sample position of moving objects on road network at time t_i received by positioning devices is represented as $sample_i$, then the possible paths between two successive positions $<sample_i, sample_{i+1}>$ is represented by the path set $Ph(k)$, $k=1,2,\ldots,n$, and meet $t \leq t_{i+1}-t_i$.

DEFINITION 5 Given segment RID, query time t and probabilistic threshold α. The probabilistic value of OID possibly going through path RID at time t is represented as $p_{t,RID}(OID)$. Then the result of the probabilistic range query is a set of all moving objects meet $p_{t,RID}(OID) \geq \alpha$.

4. Index Structure

UPBI (*UPA-tree* and B^+*-tree* Index) index [20] is structurally divided into two layers: the *UPA-tree* and the B^+*-tree* (or the *Region table*). *Region table* records the Region's ID which the paths between each boundary vertices of different subgraph in *UPA-tree* belong to. The leaf nodes of B^+*-tree* point to certain position of moving object stored in *HBase*. *UPBI* index structure is shown in Figure 1.

4.1 Spatial Index

Firstly, the road network is partitioned into k subgraph using METIS 5.0[20]. 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.

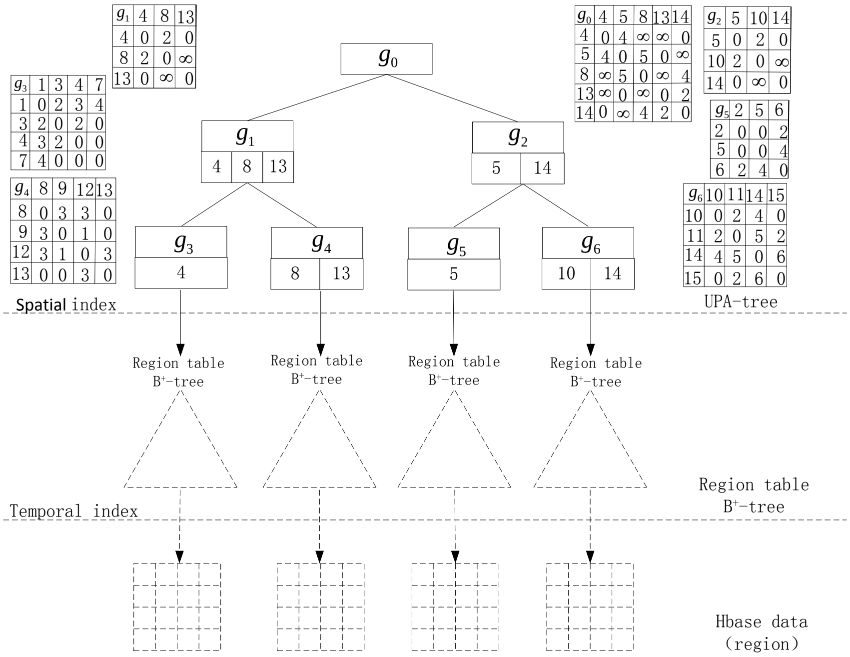


Figure 1. *UPBI* Index Structure

4.2 Temporal Index

In order to improve the utilization of the system and the query efficiency, initially the Region only stores certain sample data, and then stores the possible paths between the boundary vertices in the query processing step by step. The latter does not involve the temporal dimension, so it only need to index samples' time. *UPBI*'s temporal index uses B^+*-tree* index structure. As shown in Figure 1, each of the *UPA-tree*'s leaf nodes corresponds to a B^+*-tree* index. According to the sampling interval on the actual road network, the paper has solved two cases: one case, sampling frequency is consistent under common intelligent transportation system. We parallel create B^+*-tree* index based on time granularities. It can ensure a balance of B^+*-tree*. Another case, sampling frequency is not consistent in war or saving energy situation. We directly take traditional B^+*-tree* index based on the sampling time point which can ensure a balance of B^+*-tree* as well.

5. Probabilistic Range Queries

There are two cases in the result sets obtained from the probabilistic range queries. First, the moving object *OID* just has a certain $sample_i$ which meet at the moment t on the road *RID*, that is $t_i=t$, and it can be regarded as $p_{t,RID}(OID)=1$. In this case, the result can be obtained by combining the *UPA-tree* and the B^+*-tree* in *UPBI* index structure. Second, two adjacent samples $< sample_i , sample_{i+1}>$ of the moving object meet $t_i<t>t_{i+1}$, so that the probability value $p_{t,RID}(OID)$ of moving object *OID* passing by road *RID* at the moment t meets $0<p_{t,RID}(OID)<1$. In this case, samples and given road *RID* may belong to the same leaf node of *UPA-tree*, or may be one sample and given road *RID* belong to the same leaf node of *UPA-tree* and another sample belong to different leaf node. The worst case, the three belong to three different leaf nodes of UPA-tree. We use space constraints *r-Restrict* method, which pre-selects samples that meet the time condition and then specifically calculates to obtain the final results.

5.1 Query Algorithm

The parallel probabilistic range queries for uncertain data based on MapReduce is shown in Figure 2.

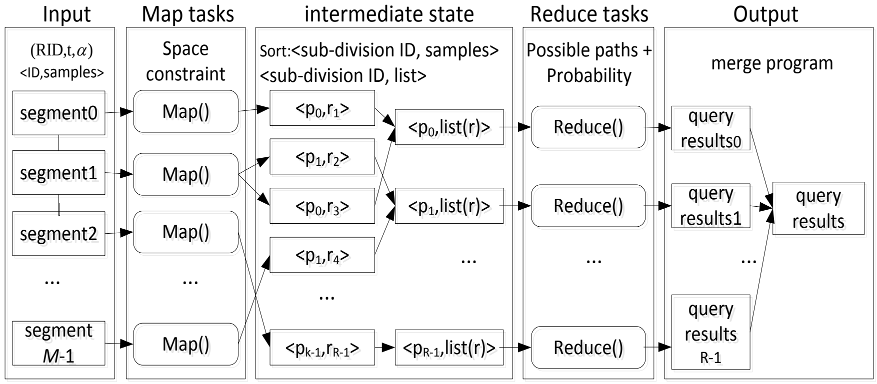


Figure 2. Process of the Parallel Probabilistic Range Queries for Uncertain Data

1) Map stage: The input samples data <ID, samples> are divided into *M* segments, which is corresponding to the same number of Map tasks. Input of each Map operation is corresponding to key-value pairs <ID, samples> in the segment. Map operation calls a user-defined space constraint function, determine whether the input *OID* of samples at the query moment *t* may be on the query road *RID*. If the conditions met, then the function judges *UPA-tree* sub-division according to the spatial location of $sample_i$. Output of Map operation is intermediate state key-value pairs <sub-division ID, samples>. Then, according to the sub-division ID the output data sets will be sorted and a new <sub-division ID, list> tuple has been generated. The purpose is to make the queries of samples which have the same beginning $sample_i$ be concentrated in the list of the same sub-division and effectively takes advantage of the calculated results to improve query efficiency. Then those tuples will be split into *R* fragments according to the scope of the sub-division ID which are corresponding to the number of Reduce tasks.

2) Reduce stage: Input of each Reduce operation is <sub-division ID, list>. Reduce operation calls functions of the possible paths queries and the probability calculation, and finally calls the merge program of sub-query results to combine all queries results into a complete result.

The space constraint r-Restrict and the probability calculation using vertices' earliest arrival time $t_{ea}(v_k)$ and latest departure time $t_{ld}(v_k)$ is the same to the continues probabilistic range query of moving objects on road network[20], possible paths calculation will be introduced as follow:

5.2 Possible Paths Calculation

For the possible paths queries we learn from the thought of processing shortest path query in paper [21]. We divide possible paths query respectively into two cases: samples belonging to the same leaf node and the different leaf nodes. For the former we directly using *Breadth-First-Search* strategy, and for the later we separately calculate possible paths from $sample_i$ to the boundary node of the leaf node which $sample_i$ belongs to, and possible paths between the boundary node of the leaf node which $sample_{i+1}$ belongs to and $sample_{i+1}$, and possible paths from above two kinds boundary nodes. Finally, possible paths of < $sample_i$, $sample_{i+1}$> can be obtained by using combination according to the time constraint. Possible paths of boundary nodes of the two leaf nodes can be acquired by using the following method: in accordance with the hierarchical relationship of *UPA*-

tree, it starts from the leaf node which $sample_i$ belongs to until the first public parent node layer by layer, and then from the first public parent node to the leaf node witch $sample_{i+1}$ belongs to layer by layer, respectively. It should separately calculate the possible paths of boundary vertices of adjacent layers nodes, and finally acquiring the result by using combination.

There are two types of repetitive computation in the processing of possible paths query. One is possible paths calculation of boundary vertices $B(g_i)$ of each adjacent layer node. Our solution is storing increased possible paths between $B(g_i)$ in each query. And possible paths stored in *Hbase* are divided into between boundary vertices of each node and their brothers or parents nodes. Obviously *Region table* scale is gradually expanded, but when possible paths of all boundary vertices in the space are recorded *Region table* does not change again. While in every query processing, as long as we find possible paths meeting the conditions among $B(g_i)$ in the region, and then on this basis we increase possible paths of new vertices obtained by using p-region [22].

Another repetitive computation is that the starting vertice of the two kind paths is the same, as well as the end and the starting vertices are not in the same *UPA-tree* leaf node, so we repetitively calculate the path from the starting vertice to the boundary vertices of the subgraph which the starting vertice belongs to. This occurs in a high probability in actual road network. The repetitive calculation also will appear on the road which takes the same vertice as the end point and each pair $sample_i$ and $sample_{i+1}$ is not in the same leaf node.

For this repetitive calculation, we divide samples according to $sample_i$ belonging to different *UPA-tree* leaf nodes, and then judge whether $sample_i$ and $sample_{i+1}$ of each samples are in the same *UPA-tree* leaf node. If the samples are not in the same leaf node, we continually divide them according to $sample_i$ position (the end vertice of the road which $sample_i$ is on), and then for each division we only need to calculate possible paths once from $sample_i$ to boundary vertices of the leaf node which $sample_i$ belongs to.

6. Experimental Evaluation

All experiments are implemented in JAVA language based on Hbase-0.90.4 and Hadoop-1.0.4. There are four Datanodes and one Namenode, and the configuration is: CPU: Intel Core i5-2450M, 2.5 GHz dual-core, memory: 4GB, disk: 500G, os: Ubuntu Linux.

The road network data is the traffic network of Colorado [23] which has 435666 vertices, 1057066 edges. We use the generator proposed in paper [24] to synthetic 10000 vehicles on the Colorado road network, and then continuously record the location of these vehicles with the same sampling interval. We respectively get 0.1, 1, 3, 5 and 10 million records.

6.1 Performance of the Query Time

6.1.1 Effect of the Data Set Size: As shown in Figure 3, the query time for the four index structures presents increasing trend with the increase of data set size. While the data set size bellowing 2 million, the query time of the parallel query based on UPBI is higher than the single query, and while the data set size exceeding 2 million query time is significantly lower than the single query. And the main reason is the same to the index construction. It must cost time in starting MapReduce task. And when data set is more than 2 million, the parallel query can dynamically adjust the number of Map and Reduce tasks to suppress query time. As shown in Figure 3, the query time of query technology based on *UTH* and *PUTI* has been superior to query technology based on *UPBI*. The main reason is that the latter two needs to deal with uncertain path queries and probability calculation.

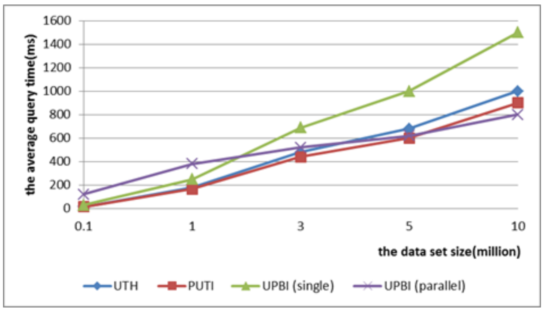


Figure 3. Average Query Time

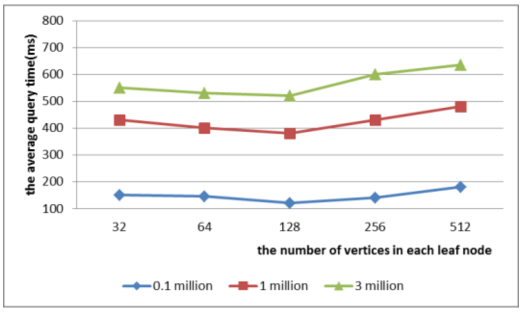


Figure 4. Average Query Time

6.1.2 Effect of the Number of Vertices in each Leaf Node: In this experiment we take the sampling interval as 180 seconds and take the probability threshold as 0.7. As shown in Figure 4, with the increase of the number of vertices in each leaf node, the query time is decreasing first and then increasing. The reason is that with the increase of vertices' number, the possible path queries have gradually transferred from different leaf nodes to the same leaf node. When the number of vertices of leaf nodes is bigger, the query cost is also greater. As shown in Figure 8, when the number of vertices in each leaf node is 128, the query of different leaf nodes and the same leaf node achieves the optimal.

6.1.3 Effect of the Sampling Interval: Taking the sampling interval as 10, 20, 30, 40, 50, 60, 120, 180 and 240 seconds, the probability threshold as 0.7, the number of vertices in each leaf node as 128, as shown in Figure 5, with the increase of sampling interval, the query time presents increasing trend, and the growth of query time is flat before 50 seconds and rapid after 50 seconds. The reason is that when the sampling interval is bigger, the possible path between two adjacent sampling points of the same moving object is more, and eventually makes query time increases. Considering practical application and data storage factors we take sampling interval as 180 seconds in the follower.

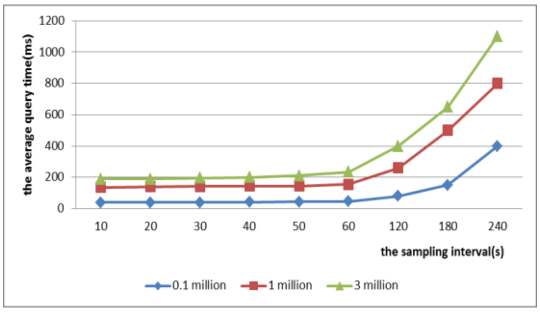


Figure 5. Average Query Time

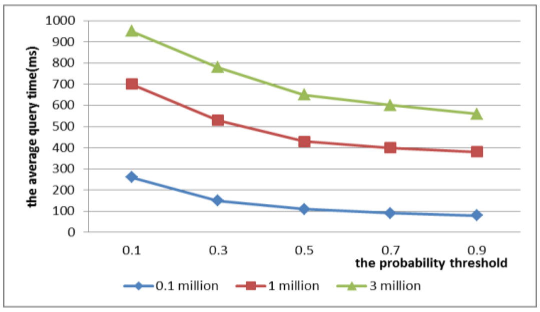


Figure 6. Average Query Time

6.1.4 Effect of the Probability Threshold: As shown in Figure 6, with the probability threshold increasing, the query time has showed a trend of decline. The reason is that the bigger the probability threshold is, the less the query candidate sets is. We can further observe through Figure 6 that with the increasing of data size, query time also increases. This is mainly because large amount of data causes large query candidate set, and the time used in time constraint and space pruning also will increase. Considering the decline trend greater when probability threshold is 0.7 in Figure 6, we take probability threshold as 0.7 in the follower.

6.2 Performance of the Query Precision

Query precision is expressed as $\#(Q^{i}_{obtain} \cap g^{10}_{real}) \,/\, \#(Q^{i}_{obtain}) * 100\%$, Q^{i}_{obtain} presents the probabilistic range queries results while sampling interval is *i* which respectively takes 20, 40, 60, 120, 180 and 240 seconds, g^{10}_{real} presents the sampling results of the actual segment in every 10 seconds. We design false negative ratio $(\#(g^{10}_{real}) - \#(Q^{i}_{obtain} \cap g^{10}_{real})) \,/\, \#(Q^{i}_{obtain}) * 100\%$ to resolve the missed real value in query processing problem.

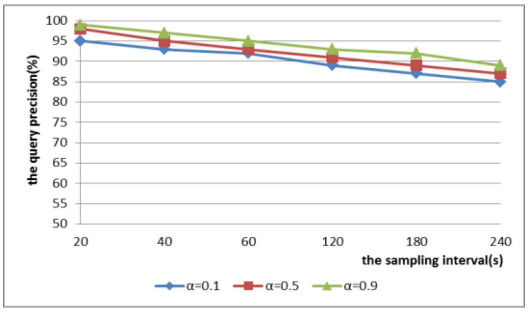


Figure 7. Query Precision

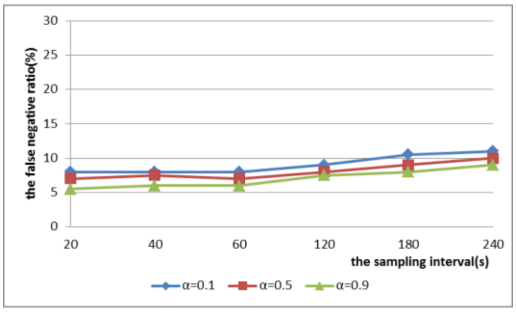


Figure 8. False Negative Ratio

As shown in Figure 7 and Figure 8, with the increase of sampling interval, the precision of the query reduces, and the false negative ratio rises. Query precision of all sampling interval is over 85%, and when the sampling interval is 180 seconds and the probability threshold is 0.5 the precision of the query can achieve 90% above, and the overall value is between 6%~11%. The reason is that when the sampling interval is bigger, the possible path between two adjacent sampling points is more.

7. Conclusion

In recent years, the uncertain location of the moving objects affected by the sampling frequency attracted more and more attention in the field of MOD. In this paper, the UPBI index and the probabilistic range parallel queries of moving objects on road network based on Hadoop has been proposed. The experiment prove that index and query algorithm proposed effectively enhance query efficiency and precision. It will be considered to use the queries results to improve the precision of the probabilistic aggregation queries considering location uncertain on road network.

Acknowledgment

This work is funded by the National Natural Science Foundation of China (No.61370091, No.61170200), Natural Science Foundation of Jiangsu Province of China (No.BK2012059), and Advance Research Foundation of PLA University of Science and Technology (Retrieval Method for moving objects based on Space Network).

References

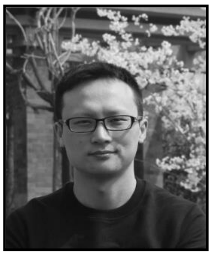
[1] R. Güting and M. Schneider. Moving Objects Databases. Morgan Kaufmann, (2005).
[2] A.V. De and R. Güting. Supporting uncertainty in moving objects in network databases. Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems, (2005), pp.31–40.
[3] X. Li, H. Lin. Indexing network-constrained trajectories for connectivity-based queries. International Journal of Geographical Information Science. 20, 3(2006), pp.303–328.

[4] 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), pp. 1022–1026.
[5] D. Pfoser, C. Jensen. Trajectory indexing using movement constraints. GeoInformatica. 9(2005), pp. 93–115.
[6] P.I. Sandu, K. Zeitouni and V. Oria. PARINET: A tunable access method for in-network trajectories. Proceedings of the 2010 IEEE 26th International Conference on Data Engineering, (2010) , pp. 177–188.
[7] I.P. Sandu, K. Zeitouni, V. Oria. Indexing in-network trajectory flows. Vldb Journal. 20(2011), pp. 643–669.
[8] G. Trajcevski, O. Wolfson, K. Hinrichs, S. Chamberlain. Managing uncertainty in moving objects databases. Acm Transactions on Database Systems, 29, 3(2004), pp. 463−507.
[9] T. Emrich, H.P. Kriegel, M. Nikos, M. Renz, A. Züfle. Querying Uncertain Spatio-Temporal Data. Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, (2012), pp. 354–365.
[10] Y. Zhang, X. Lin, Y. Tao, W. Zhang, H. Wang. Efficient Computation of Range Aggregates against Uncertain Location Based Queries. Knowledge and Data Engineering, IEEE Transactions, 24(2012), pp. 1244–1258.
[11] G.Trajcevski, R. Tamassia, H. Ding, P. Scheuermann, I. Cruz. Continuous probabilistic nearest-neighbor queries for uncertain trajectories. Proceedings of the 12th International Conference on Extending Database Technology. Saint-Petersburg: ACM Press, (2009), pp. 874−885.
[12] R. Cheng, J.C.Chen, M. Mokbel, C.Y. Chow. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. Proceedings of the 24th International Conference on Data Engineering. Cancun: IEEE Computer Society, (2008), pp 973−982.
[13] R. Cheng, L. Chen, J.C. Chen, X.K. Xie. Evaluating probabilistic threshold k-nearest-neighbor queries over uncertain data. Proceedings of the 12th International Conference on Extending Database Technology. Saint-Petersburg: ACM Press, (2009), pp. 672-683.
[14] Y.K. Huang, C.C. Chen, C. Lee. Continuous K-nearest neighbor query for moving objects with uncertain velocity. Geoinformatica. 13(2009), pp. 1−25.
[15] B. Kuijpers, W. Othman. Modeling uncertainty of moving objects on road networks via space-time prisms. International Journal of Geographical Information Science. 23(2009), pp. 1095–1117.
[16] M. Hua, J. Pei. Probabilistic path queries in road networks: Traffic uncertainty aware path selection. Proceedings of the 13th International Conference on Extending Database Technology. Lausanne: ACM Press, (2010), pp. 347−358.
[17] K. Zheng, Y. Zheng, X. Xie, X. Zhou. Reducing Uncertainty of Low-Sampling-Rate Trajectories. Data Engineering (ICDE), Proceedings of the IEEE 28th International Conference on Washington, DC, (2012), pp.1144– 1155.
[18] 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), pp. 283–294.
[19] L. Chen , Y. Tang ,M. Lv , G. Chen. Partition-based range query for uncertain trajectories in road networks. Geoinformatica. 19, 1 (2015), pp. 61-84.
[20] Y.Q. Shi, J. Feng and Z.X. Tang. UPBI: An Efficient Index for Continues Probabilistic Range Query of Moving Objects on Road Network. International Journal Multimedia and Ubiquitous Engineering. 10,5(2015), pp.355-372.
[21] R. Zhong, G. Li, K. Tan, L. Zhou. G-tree: an efficient index for KNN search on road networks. Proceedings of the Acm International Conference on Conference on Information & Knowledge Management, (2012), pp. 385-394.
[22] J. Feng. A Study on Multi-scale/Multi-theme Map Information Model and Nearest Neighbor Search Method. Doctoral dissertation, Nagoya University(2004).
[23] http://www.dis.uniroma1.it/challenge9/index.shtml.
[24] D. Christian, B. Thomas, R. Güting. BerlinMOD: a benchmark for moving object databases. The VLDB Journal, 18(2009), pp.1335–1368.

Authors

Yaqing Shi, She received the Ph.D. degree in Computer Application Technology from Hohai University, China, in 2015. Now, she is a associate professor in Command Information System Institute, PLA University of Science and Technology, Nanjing, China. 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. Her research interests include data management, spatiotemporal indexing and search methods, ITS and domain data mining. E-mail: fengjunhhu@gmail.com.



Zhengping Ren, He received the Ph.D degree in Software Engineering from PLA University of Science and Technology, China, in 2008. Now, he is an associate professor in Command Information System Institute, PLAUST, Nanjing, China. His research interests include software security testing, software quality evaluation and performance modeling. E-mail: zhengpingren@gmail.com

Wenjuan Xie, She works in Command Information System Institute, PLA University of Science and Technology, Nanjing, China. Her research interests include spatiotemporal indexing and searching methods and ITS.

