

Research Article

A Hybrid Aggregate Index Method for Trajectory Data

Yaqing Shi , Song Huang , Changyou Zheng , and Haijin Ji 

Command & Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China

Correspondence should be addressed to Yaqing Shi; 77745227@qq.com and Song Huang; hs0317@163.com

Received 5 February 2019; Revised 30 June 2019; Accepted 18 July 2019; Published 18 August 2019

Academic Editor: Paolo Boscariol

Copyright © 2019 Yaqing Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The aggregate query of moving objects on road network keeps being popular in the ITS research community. The existing methods often assume that the sampling frequency of the positioning devices like GPS or roadside radar is dense enough, making the result's uncertainty negligible. However, such assumption is not always tenable, especially in the extreme occasions like wartime. Regarding this issue, a hybrid aggregate index framework is proposed in this paper, in order to perform aggregate queries on massive trajectories that are sampled sparsely. Firstly, this framework uses an offline batch processing component based on the UPBI-Sketch index to acquire each object's most likely position between two continuous sampling instants. Next, it introduces the AMH⁺-Sketch index to processing the aggregate operation online, making sure each object is counted only once in the result. The experimental results show that the hybrid framework can ensure the query accuracy by adjusting the parameters L and U of AMH⁺-Sketch index and its space storage advantage becomes more and more obvious when the data scale is very large.

1. Introduction

Along with the rapid development of intelligent transportation systems (ITSs), massive novel applications are proposed in the recent years. Using the trajectories collected by the positioning devices like roadside radars, the enemy's forces can be inferred for decision-making in the wartime, or estimate the possible congestions and tune up the signals' time to relieve regional traffic pressures based on the short-term forecasting results. The route plans are made by the electronic navigation devices based not only on the current but also the future possible traffic congestions, etc. All such applications need to monitor the movements of vehicles or pedestrians, collect their locations, and depict their moving trajectories, using aggregating methods to identify the interested regions by most users. In addition, such queries need to be processed as fast as possible, making the aggregate query of moving objects on road network continuously being a hot research topic in the ITS research community, and a lot of approaches are proposed in the recent years.

Most existing methods utilize the trajectories collected by GPS or roadside radars, which continuously come as a stream in real-time, in an unpredictable speed and with an unbounded size. As a result, these methods can only scan

data once in the run-time and get the aggregated results as precise as possible. Therefore, the approximate processing technology is often used in most methods. The core part of the approximate processing technology is using an approximate query processor on data stream and maintaining a synopsis data structure. This structure is much smaller than the data stream size itself. If the size of the data stream is n , the size of the synopsis data structure will not exceed $O(\log n)$ [1, 2], so the synopsis data structure can be resident in memory without accessing the disk, which can meet the requirements of real-time monitoring and scheduling in ITS. However, the query accuracy of the existing approximate query processor for data stream cannot be guaranteed. The number of sketch and the number of moving objects in the query region cannot achieve real-time dynamic coordination in the sketch technology [3–8]. The histogram technology [1] can simulate the uniform distributed data sets very well, but once the data are concentrated on some popular elements with large proportions, histogram representation may produce a lot of mistakes, making it unsuitable for solving the repetition counting in the aggregate queries.

In this case, the single processing stream cannot guarantee the accuracy of aggregate query. Figure 1 shows the

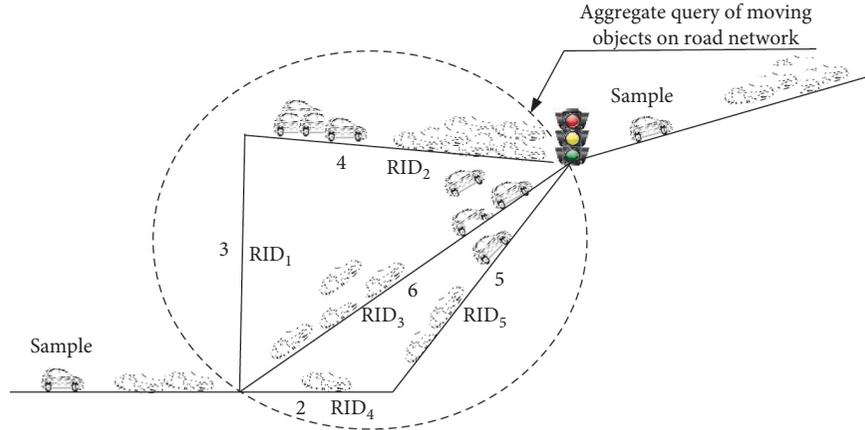


FIGURE 1: Uncertainty caused by sampling frequency.

uncertainty of a vehicle's movement during such a blind area, where the values indicate the possible passing time of the vehicle within this area, and RID is the road section identification. Assuming that the sampling frequency is 8 time units, the vehicle's location at a certain time between the two continuous samples is uncertain, that is to say, the aggregate value on a road section must have a probability attribute. As shown in Figure 1, the marked value in the figure is a time value. Assuming that the sampling frequency of the vehicle is 8 units of time, then the position of the moving vehicle at a certain time between the two consecutive sampling samples is uncertain, that is to say, if it is necessary to query the vehicle aggregate value of a certain road section between them, it must have a probability attribute.

In order to perform the aggregate queries as efficient as possible, in such an uncertain environment, a hybrid aggregate index framework for moving objects on road network is proposed. This framework combines the stream processing strategy with the batch processing components. The stream processing is established based on a synopsis data structure, while the batch processing is set up based on the data storage. Our approach optimizes the massive trajectory data-accessing method while concerning the uncertainty issue at the same time. Afterwards, it introduces an index structure to fetch the accurate and real-time aggregation results. We also combine the sketch technology and the histogram technology together to propose a hybrid aggregate index method, which includes UPBI-Sketch (UPA-tree and B^+ -tree index of Sketch) and AMH-Sketch (Adaptive Multidimensional Histogram of Sketch) index at the same time. The UPBI-Sketch supports the probability range query of moving objects on road network to solve the acquisition of uncertain data, and the AMH-Sketch index supports the aggregate query of moving objects on road network to solve the spatiotemporal repetition counting and nonuniform distribution.

The main contributions of the paper include:

Firstly, a hybrid aggregate index framework for trajectory data is constructed. It is divided into batch processing and stream processing. The former part deals with the acquisition of uncertain data, and the

latter part performs the aggregate query of trajectory data.

Secondly, the UPBI-Sketch index is designed and implemented to support the probability range query. It uses the UPA-tree (uncertain path based on the assembly method) to solve the non-Euclidean spatial indexing issue and the B^+ -tree for the temporal indexing.

Thirdly, the AMH⁺-Sketch index is designed and implemented to support the aggregate query for trajectory data. The sketch [9] technology has been introduced into the AMH⁺ [10] to improve the sketch calculation's precision directly.

Fourthly, an aggregate query algorithm is proposed, which divides the spatial area by the sketch value of the samplings. According to the overlapping ratio of the intersecting area within the query region, the suitable number of the sketches inside the query area is determined to improve its precision.

The remaining part of this paper is organized as follows. Section 2 presents the related work about the aggregate index methods. Section 3 introduces the basic definitions and the uncertain processing model. Section 4 mainly describes the hybrid index structure with more details, and then the aggregate query algorithm is introduced in Sections 5. Section 6 presents the experimental evaluations. At last, the conclusions and the future work are given in Section 7.

2. Related Work

A hybrid aggregate index method for the trajectory data involves two kinds of indexing strategies. One is the aggregate index based on the processing stream, and the other is the index based on the batch processing to deal with the data acquisition.

2.1. Aggregate Index Based on the Processing Stream. The characteristics of the aggregate query of moving objects in non-Euclidean spatial is massive data and real-time. It is usually solved by using the data stream approximation

processing technology in maintaining the synopsis data structure in memory. Sketch technology [3–8] is a common approximate processing technology. Tao et al. [9] combined a RB-tree [11] with sketch technology and proposed using OR operation to solve the spatiotemporal repetition counting. Lochert et al. [12] proposed an algorithm based on soft-state sketches to solve aggregation value-merging problem in the same area by hierarchical aggregate query. Garofalakis et al. [4] proposed an algorithm that abstracts sketch geometric connections from distributed data stream. The AMS Stream sketches also stores the high update data in a synopsis data structure. Papapetrou et al. [13, 14] proposed ECM sketches, which replace the counter of count-min sketch [15] with a sliding window to solve the distributed aggregate with uniform distribution. However, there are two problems in the existing index structure with the sketch technology. Firstly, road network is a kind of non-Euclidean space. The aggregate query on the moving objects in the Euclidean space cannot be used directly. Secondly, there are two cases in the process of repeated calculation in sketch technology, which will lead to large query error: When the actual number of vehicles in a spatiotemporal area is very large, the number of sketches used is very small. When the actual number of vehicles in a spatiotemporal area is very small, the number of sketches used is large. Therefore, the direct use of sketch calculation will bring considerable errors.

Sun et al. [16] proposed the adaptive multidimensional histogram (AMH) index and divided nonuniform distribution moving objects on road network into different buckets. It solved the problem of an uneven distribution of moving objects on road network. Jin et al. [17] studied the constraint conditions of bucket variance in AMH and proposed the AMH* index structure to solve the bucket maintenance operation. Jin et al. [10] put forward an AMH⁺ index for using low spatial complexity to get high-quality aggregate query results. The above three indexes also have the spatiotemporal repetition counting problem of moving objects. Dobra et al. [18] put forward a new sketch partitioning technique, which allows each part sketch to be assigned to the appropriate memory space and greatly improves the quality of the approximate query.

The research group of the author combined the sketch technology with an RR-tree and proposed the sketch RR-tree [19] for road network. On this basis, the DynSketch index [19] is proposed in combination with the AMH dynamic partition. Considering the query error caused by the maximum number of buckets AMH limited, the DS index [20] is proposed. The author proposed the DSD⁺ index [21, 22], which ensured that the storage space was significantly reduced on the basis of the time and the error advantages, and the direction aggregate was solved. The author realized the real-time dynamic coordination [23] between the number of sketches and the number of moving objects in the AMH⁺ buckets' division phase and the aggregate query phase. It improved the aggregate query accuracy of moving objects on road network.

2.2. Index for Range Query Based on Processing Batch. In the recent years, more and more studies have focused on the

probability range query of moving objects in uncertain models, but most of them are concentrated on the Euclidean space [24–29]. Kuijpers and Othman [30] put forward the uncertain trajectory model on road network—the space-time prism. This model only involved the alibi query, which asks whether two moving objects could have possibly met or not. The query complexity is relatively high for others. Hua and Pei [31] considered the spatial query problem based on the assumption of uncertain network edge weights on road network. On the basis of the general probability distribution function, Gu et al. [32] proposed a probabilistic query incremental processing model and a query optimization method based on the segmentation interval. The above uncertainty of moving objects is a kind of location information uncertainty. It is based on the accuracy of positioning devices, the location technology, the network delay, and the weight of the network edge. This paper is about the location uncertainty of moving objects between two continuous samples caused by sampling frequency. There are differences in semantics, model, and application background. The papers [33, 34] are consistent with the probability range query background of the subject. Consider that the actual moving object does not travel at a uniform speed on road network. It is impossible to determine the location of the moving object at a specific time t or a certain time period Δt . Consider that each road section has attributes of the length of the section $l(e)$ and the maximum speed limit $s(e)$, from which the shortest time $t_m(e) = l(e)/s(e)$ can be obtained, which can limit the time range of the moving object on each road section. Zheng et al. [33] used the probability distribution function of dependent time to express the uncertainty of moving objects on road network and proposed an index mechanism and a spatiotemporal range query algorithm for uncertain trajectory. The actual sample locations are recorded on the trajectory list of the UTH index, and the earliest arrival time and the latest departure time of the vertex of all moving objects in all possible paths are also recorded on the disk. Frequent disk reading and writing are needed in the process of index creation. It cannot meet the real-time performance of massive moving object data processing in large-scale road network. Zheng et al. [35] proposed a historical route-based inference system (HRIS), which made full use of the historical trajectory information of moving objects on road network to reduce uncertainty. It did not involve related query problems. Chen et al. [34] proposed a partition based on uncertain trajectory index (PUTI) in which the graph representing road network is partitioned according to the network distance of moving objects. However, the frequent uncertain trajectory insertion in the index-creation processing leads to the huge system burden.

Researchers have carried out a wide study [36–40] for the prediction range query on road network. To go beyond short-term prediction, Jeung et al. [36] proposed a network mobility model which offers a concise representation of mobility statistics extracted from massive collections of historical object trajectories. Based on the model, a maximum likelihood and a greedy algorithm for predicting the travel path of an object are presented. At the same time, an

effective index mechanism is proposed to support predictive range queries. Abdeltawab et al. [37] put forward the iROAD framework. By pruning the related space of each moving object to reduce the computing time significantly, iROAD can handle the large-scale road network and mass moving objects. All the above studies assume that the motion line of moving objects is certainty and the query accuracy cannot be guaranteed. The uncertainty of environment or trajectory is considered in the prediction processing. Zhang et al. [38] considered the dynamic and uncertain environment and proposed an effective reasoning method to predict the future locations of moving objects. The prediction can be integrated into the index design of uncertain moving objects. The recursive motion function (RMF) is proposed to model different types of uncertain motion patterns, and the spatial prediction tree (STP-tree) is introduced to index uncertain motion patterns by Tao et al. [39]. Qiao et al. [40] proposed an uncertain trajectory prediction algorithm of moving objects called PutMode. The algorithm constructed the trajectory continuous time Bayesian networks (TCTBNs) and then obtained the possible trajectory based on the TCTBNs by predicting the moving behavior of moving objects. The premise is to consider the individual information of moving objects, such as speed and moving direction.

3. Research Foundation

3.1. Trajectory Data. The trajectory data of moving objects on road network consist of a series of discrete location points, which is divided into sampling trajectory and nonsampling trajectory.

Definition 1 (moving object's sampling location). The sampling location of moving object OID on road section RID at time t_i is represented as (OID, RID, v_i , l_i , t_i), where v_i is RID's vertex which OID enters and l_i is the distance from v_i to OID, $i = 1, 2, \dots, n$. The combination of the four aforementioned characteristics is simply represented as sample_i .

Definition 2 (moving object's sampling trajectory). The sampling trajectory of moving object OID is a set of moving objects' sampling locations in continuous time $\{(OID, RID_0, v_0, l_0, t_0), (OID, RID_1, v_1, l_1, t_1), (OID, RID_2, v_2, l_2, t_2), \dots, (OID, RID_n, v_n, l_n, t_n)\}$, simply expressed as $\{\text{sample}_0, \text{sample}_1, \text{sample}_2, \dots, \text{sample}_n\}$.

The sampling trajectory is obtained by the positioning device. This paper does not consider the uncertain trajectory caused by the device accuracy or the sampling algorithm; that is, it is assumed that the location of moving objects returned by all devices is reliable.

The nonsampling trajectory of moving objects on road network should be the location set in all continuous samples $\langle \text{sample}_i, \text{sample}_{i+1} \rangle$. It is represented as $\{u_{0,1}^1, u_{0,1}^2, \dots, u_{0,1}^m, \{u_{1,2}^1, u_{1,2}^2, \dots, u_{1,2}^m\}, \dots, \{u_{(n-1),n}^1, u_{(n-1),n}^2, \dots, u_{(n-1),n}^m\}\}$, where $u_{i,i+1}^j$ is the j th uncertain position between $\langle \text{sample}_i, \text{sample}_{i+1} \rangle$. In fact, because the speed of moving objects is not constant on road network, it cannot be used

to express the nonsampling trajectory of moving objects by the discrete location points.

Definition 3 (moving object's possible path between continuous samples). The possible path PPH of moving object OID between the two continuous samples $\langle \text{sample}_i, \text{sample}_{i+1} \rangle$ is a set of path ph_j , $j = 1, 2, \dots, n$. ph_j meet $t_m(\text{ph}_j) \leq t_{i+1} - t_i$, where $t_m(\text{ph}_j)$ is the sum of each road section's shortest time in a possible path ph_j .

3.2. Aggregate Query Definition. The purpose of the aggregate query in the traffic monitoring system is analyzing the history and current congestion to predict the future traffic flow. Therefore, the aggregate query of moving objects on road network is divided into three categories based on the time range. The definition of the query is given as follows:

Definition 4 (aggregate query of moving objects on road network). The aggregate query of moving objects on road network $q(q_R, q_T, \alpha)$ returns the number of moving objects with a probability of higher than probability value α at query time q_T on query road sections contained in q_R :

- (i) if $q_T < 0$, then $q(q_R, q_T, \alpha)$ is the historical aggregate query
- (ii) if $q_T = 0$, then $q(q_R, q_T, \alpha)$ is the current aggregate query
- (iii) if $q_T > 0$, then $q(q_R, q_T, \alpha)$ is the prediction aggregate query

3.3. Processing Model. The traditional aggregate query of moving objects on road network usually adopts the processing stream, that is, to maintain a synopsis data structure which is much smaller than the data stream. As shown in Figure 2, the premise of this model is that the sampling frequency of GPS and roadside radar can reach intensive to exclude the influence of uncertain data. It is considered that they are all sample data. However, the positioning devices usually cannot achieve such a high sampling frequency in the practical ITS contexts. The lower the sampling frequency is, the greater the uncertainty generated between two continuous samples. As shown in Figure 3, the data stream processing model takes the sample data collected by the positioning devices and the uncertain data which must be derived from the sample data as the study subjects. A hybrid aggregate index for trajectory data is designed in this paper. Under the premise of low sampling frequency, the sample data are stored and the uncertain data are obtained by the probability range query. Both of them together get the synopsis information by the approximate processing technology. At last, the approximate aggregate query results which meet the requirements of accuracy are obtained.

4. Hybrid Aggregate Index Method

4.1. Index Framework. The hybrid aggregate index framework for trajectory data is divided into two modules: batch processing and stream processing. The batch processing module focuses on how to obtain uncertain data, that is, to

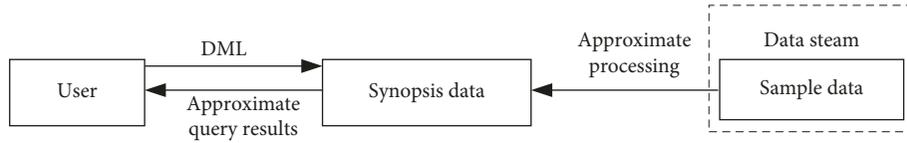


FIGURE 2: Traditional data stream processing model.

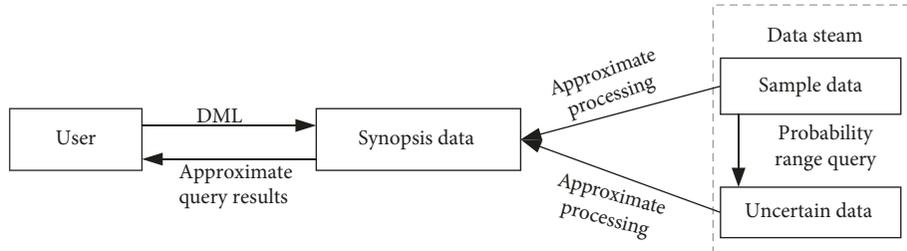


FIGURE 3: Data stream processing model considering trajectory uncertainty.

obtain moving objects satisfying the spatiotemporal conditions. The stream processing module focuses on the spatiotemporal repetition counting and nonuniform distribution of moving objects on road network. The index framework is designed as shown in Figure 4.

When the sample data $sample_i$ collected by using the positioning devices arrive to the system, the road section sketch generator generates the sketch value at each time of each section, and the batch processing module establishes the UPBI-Sketch index according to the spatiotemporal attributes of $sample_i$ and records the sketch value in the database. In the stream processing module, the AMH^+ -Sketch index structure is constructed, buckets are divided according to the sketch value of the sample data $sample_i$, and the binary tree (BPT) index is constructed. When the client sends the query request, the system searches the bucket b_k from the BPT index and UPBI-Sketch index to process the obtained sketch information (sketch adaptation, OR operation), in result to form the final query result.

4.2. UPBI-Sketch Index. UPBI-Sketch (UPA-tree and B^+ -tree index of Sketch) is proposed to support the aggregate query of moving objects on road network. The UPBI-Sketch index structure is divided into the spatial index and the temporal index. The UPA-tree (uncertain path based on the assembly method) is designed in the spatial index, and the B^+ -tree is used in the temporal index. The advantage of UPBI-Sketch is that it can solve non-Euclidean spatial index problem by using UPA-tree. It can also deal with the spatiotemporal repetition counting in the aggregate query of moving objects on road network by replacing the SUM operation by OR operation.

4.2.1. UPA-Tree Index. The UPA-tree considers that moving objects may choose any path that conforms to $t_m(ph_j) \leq t_{i+1} - t_i$ in the two continuous samples $\langle sample_i, sample_{i+1} \rangle$. This index solves the uncertain selection of sections between the two continuous samples. The UPA-tree indexes the

sample location directly through the subgraph expressed by a leaf node. Road network relationship between $\langle sample_i, sample_{i+1} \rangle$ is expressed by the boundary vertex of all sub-graphs, the adjacency matrix, and the time constraint $t_m(ph_j) \leq t_{i+1} - t_i$. UPA-tree design has been introduced with more details in [41, 42].

4.2.2. B^+ -Tree Index. This system designs a B^+ -tree index for each section in UPA-tree leaf node and adopts that the upper level time granularity T_i indexes lower time granularity t_i . This design mainly considers that the section is the basic unit of cells in AMH^+ -Sketch index. The value of each cell is expressed as *sketch* with per unit section RID, per unit time t , and the given probability threshold α . The index's design can directly obtain the sketch value in each cell and improve the query efficiency. Each B^+ -tree's middle node includes *the node TID, the parent node ID, the left and the right node ID, the child node ID*, and *sam_Sketch*, where *sam_Sketch* is the sketch of *sample*. Each leaf node includes *the node TID, the parent node ID, the left and the right node ID, the Region ID, OID, the RowKey_TimeStamp, and sam_Sketch*, where *bucketID* is the ID of the historical bucket indexed by l_e in the *lifespan* $[l_s, l_e]$.

4.2.3. Hash Table. The hash table connects the spatial index and the temporal index. The leaf node of UPA-tree represents the subgraph of road network, which contains a number of edges and vertexes. The B^+ -tree indexes moving objects in a single section, so the hash table is designed to connect the spatial index and the temporal index. As shown in Figure 5, the edge in the leaf nodes of UPA-tree is mapped with the RID and then designs a B^+ -tree index for each *edge*.

4.3. AMH^+ -Sketch Index. The AMH^+ -Sketch index uses the idea of AMH^+ [17]. As shown in Figure 6(a), the space is divided into ω . ω cells are in the form of the two-dimensional grid, and the width of each cell $c(1 \leq c \leq \omega^2)$ is $1/\omega$ on the X- and Y-axis. F_c represents the number of moving objects in

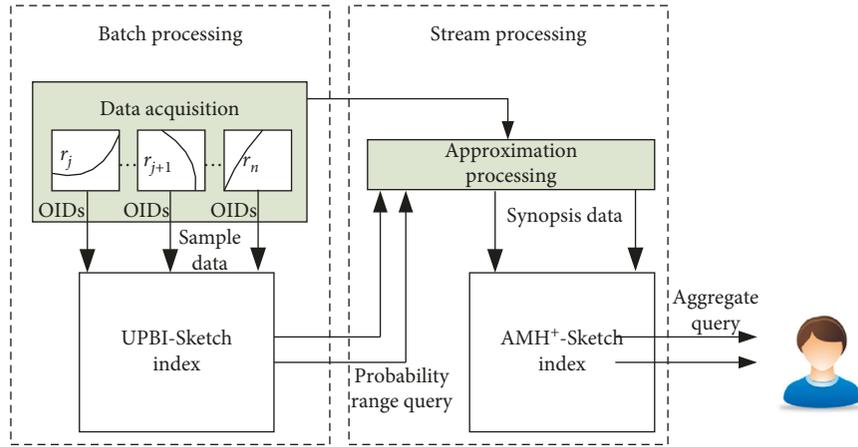


FIGURE 4: Hybrid aggregate index framework for trajectory data.

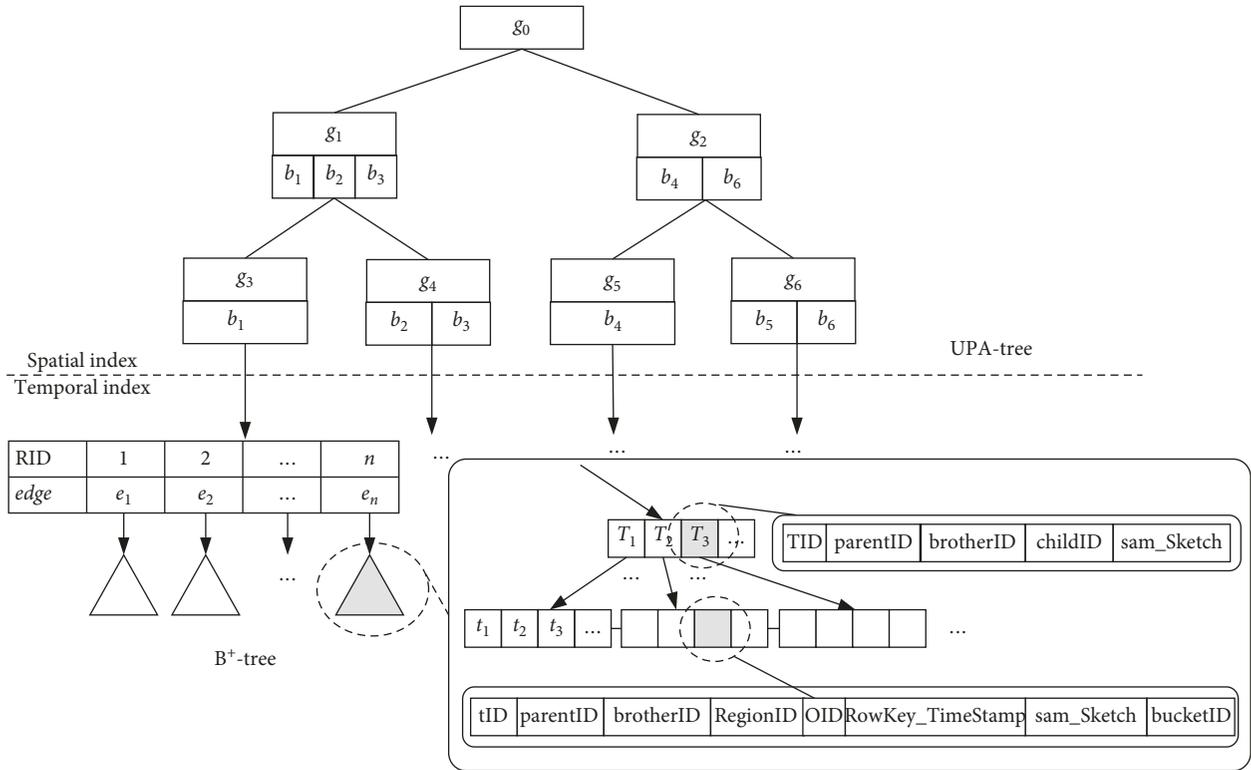


FIGURE 5: UPBI-Sketch index.

cell c . In this case, the network environment is instantiated: the two-dimensional grid represents road network. Each cell c corresponds to each section. F_c is expressed as *sketch* of the corresponding section to solve the spatiotemporal repetition counting problem.

When the sample data arrive, AMH^+ -Sketch index gets the corresponding section's *sketch* from the leaf node of UPBI-Sketch index and then puts the units close to F_c into the same bucket. The bucket is defined as a regular rectangle. Cell c does not overlap in each bucket b_k , and the bucket does not overlap too as shown in Figure 6(b). The number of buckets n is much smaller than that of the previous cells'

number ω^2 . As shown in Figure 6(d), any bucket $b_k(1 \leq k \leq n)$ can be represented as an eight-vector $(R_k, n_k, f_k, g_k, v_k, m_k, sk_k, lifespan[t_s, t_e])$, as shown in Table 1. The goal of AMH^+ -Sketch index is to minimize the sum of all buckets' v_k (weighted variance sum, WVS), $WVS = \sum_{i=1}^n (n_i \cdot v_i)$. Because the cell's number n_k covered by the bucket can be obtained by the area ratio of $R_k \omega^2$, $v_k = g_k - f_k^2$, the WVS of all buckets can be calculated using R_k, f_k and g_k .

The divided buckets at the current moment is indexed by establishing a binary partition tree (BPT), as shown in Figure 6(c), and each leaf node corresponds to a bucket b_k . The middle node is a rectangular area containing buckets

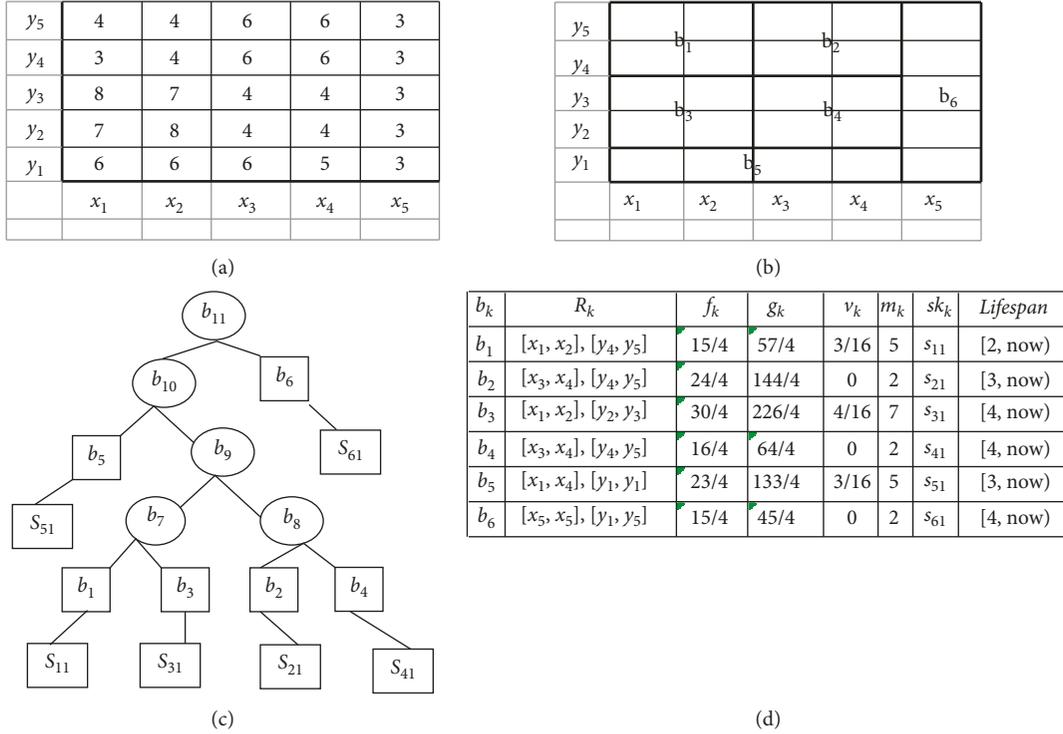

 FIGURE 6: AMH⁺-Sketch index at the current moment. (a) Sketch of cells. (b) Partition of cells. (c) BPT. (d) Sketch information of bucket.

 TABLE 1: Bucket b_k eight-vector representation.

Vector	Description	Calculation/representation
R_k	The rectangular range of bucket b_k	The coordinates of the upper left and right lower corners of bucket b_k
n_k	The number of cells in bucket b_k	Integer representation
f_k	Average frequency of cells in bucket b_k	$f_k = (1/n_k) \sum_{c \in b_k} F_c$
g_k	Average square sum of cells' frequency in bucket b_k	$g_k = (1/n_k) \sum_{c \in b_k} F_c^2$
v_k	Average variance of frequency in bucket b_k	$v_k = (1/n_k) \sum_{c \in b_k} (F_c - f_k)^2$
m_k	The number of sketches used in bucket b_k	Calculate through the appropriate sketches method
sk_k	The number of moving objects in bucket b_k	Exact value or sketch representation
$lifespan[t_s, t_e)$	Lifespan of bucket b_k	t_s : the creation time of bucket b_k t_e : the update time of bucket b_k

corresponding left and right child nodes. When dealing with the current aggregate query, BPT can be traversed to find the buckets intersecting the query area. t_e of the current bucket is represented as now, indicating that the bucket is indexed by BPT. Because of the constant updating of sample data, the sketch of c is constantly changing, resulting in buckets' updating, merging, and splitting and constantly forming new bucket.

The purpose of the aggregate query of moving objects on road network is to support the prediction of traffic congestion in the future, so the current buckets are directly stored in the memory to realize frequent real-time queries. For those history buckets, the system keeps only a four-vector $(R_k, m_k, sk_k, lifespan[t_s, t_e))$. The spatial dimension is set based on the cell of the most left upper corner of R_k . The temporal dimension is set based on t_e which is the bucket's end time. The index is created at the corresponding time of B⁺-tree index. When the historical probability aggregates the

query, it is only necessary to find the bucket that intersects the query area according to the time space condition and use its quaternion vector value to determine the sketch of the query area.

5. Aggregate Query Processing

In the aggregate query stage, the client first sends out the query request. The system uses the probability range query algorithm to acquire moving objects quickly. The specific query algorithm is in the author's paper [41, 42]. The result is also converted to the synopsis data. The system makes use of the nonrepeating counting aggregate query method to obtain the historical aggregate value.

The historical aggregate query algorithm, as shown in Algorithm 1, obtains all buckets intersected from q_R in the UPBI-Sketch and AMH⁺-Sketch index according to q_R and q_T firstly (line 1) and then obtains m_k of each intersected b_i

Algorithm Aggregate_Query (q_R, q_T, α)Input: q_R : query area, q_T : query time, α : probability thresholdOutput: *allnumber*: the approximate number of moving objects under this query condition

- (1) Obtain all buckets b_i intersected from q_R in the UPBI-Sketch and AMH⁺-Sketch index according to q_R and q_T
 - (2) The appropriate sketch number M about q_R is calculated according to the sketch number m_k of each intersected b_i ($R_k, m_k, sk_k, lifespan(l_s, l_e)$)
 - (3) **For** RID in q_R
 - (4) Query all moving objects OID at q_T time to generate sketch
//The probabilistic range query algorithm [42] is combined with the FM_PCSA algorithm [9]
 - (5) Use *OR* operation [9] and replace *ADD* operation to get *all_Sketch*
 - (6) **End for**
 - (7) Convert *all_Sketch* to the approximate value *allnumber* of moving objects by fitting curve $y = 0.747978x^{0.708563}$ [19]
- End Aggregate_Query**

ALGORITHM 1: Aggregate query algorithm.

($R_k, m_k, sk_k, lifespan(l_s, l_e)$) from the intersection buckets' vector. Combined the ratio with $R_{q_{Ri}}$ (q_R in part b_i) and bucket's area R_i , the appropriate sketch number M about q_R is calculated (line 2). The probabilistic range query algorithm [42] is combined with the FM_PCSA algorithm [9] to obtain the uncertain data, each query result OID does not record but directly uses the sketch generator to generate *sketch* (line 4) and uses the *OR* operation to get *all_Sketch* (line 5). At last, *all_Sketch* is converted to the approximate value *allnumber* of moving objects (line 7) under the query condition by fitting curve $y = 0.747978x^{0.708563}$, where x is the number of sketches used in the aggregate query and y is the number of corresponding moving objects. The fitting curve has been obtained through experiments in our previous research [19].

The complexity of the aggregate query algorithm mainly focuses on line 3 to line 6; the analysis is as follows: For the determined sample data query, the time complexity is $O(n)$ and n is the number of the total sample location. MapReduce-based parallel processing is used for the possible object queries of uncertain sample pairs and M and R are the number of Map and Reduce, respectively. The calculation of time complexity is divided into two parts: one is to divide and solve the Map part of spatial pruning; then, in the worst case, the spatial pruning calculates all sample pairs, that is, $O(n^2)$; the other is the Reduce part of possible path query, probability pruning, and position probability calculation, and the key of the latter two time consumption lies in the possible path query. The time complexity is $O(1/\omega + \chi^2 + \log_2(V/\chi) \cdot V)$, ω is the sampling frequency of the sample, χ is the number of nodes in the leaf of UPBI-Sketch index, and V is the number of nodes in road network.

The difference between the current and the historical aggregate queries is mainly on finding buckets that intersects q_R . The historical aggregate query finds the relevant buckets through the UPBI-Sketch and AMH⁺-Sketch index. However, the current bucket is indexed by BPT directly in memory, and the current aggregate query can quickly traverse BPT through the in-order traversal to find the related buckets. It satisfies the current real-time aggregate query's requirement on road network.

6. Experimental Evaluation

6.1. Data Sets. The processor used in the experiment is Intel Core i5-2450M, 2.5 GHz dual-core, with 4 GB memory and a 500 GB hard disk of 7200 R/Min, 2 MB cache. The operating system uses Ubuntu Linux. The development environment is Eclipse3.2, and the version of JDK is jdk1.6. The UPBI-Sketch index, the AMH⁺-Sketch index, and the aggregate query of moving objects on road network are implemented in Java.

The data set is divided into two parts: road network and the mobile vehicle records. In the experiment, the US Colorado traffic network data [43] (435666 intersection and 1057066 sections) were used to intercept 10000 sections of road network as road network data of the aggregate query. This paper uses the mobile vehicle generator [44] to simulate 10000 mobile vehicles on the Colorado road network, then records the location information of the vehicle every 180 seconds, and records 180 times in a row.

6.2. Results and Discussion. The experiment mainly analyzes the performance of historical probability aggregation query based on UPBI-Sketch and AMH-Sketch index. The performance of aggregation query is mainly the relative error. The approximate result of probability aggregation query in the experiment is Q_{obtain} , and the actual exact value is g_{real} . Then, the relative error will be calculated by formula $|g_{\text{real}} - Q_{\text{obtain}}|/g_{\text{real}}$, and the accuracy of the query is analyzed.

The experimental processing is mainly divided into the following steps:

- (1) The space is divided into $n \cdot n$ cells; each cell corresponds to a section. Because of a total of 10000 sections, the n takes 100. When the sample data of moving objects are updated in the section, the sketch technology is called and the sketch value of the mobile vehicle in the section is recorded in the UPBI-Sketch node item, and the F_c in the corresponding cell is updated.

- (2) The approximate number of vehicles is calculated according to the sketch values in the cell, and the adjacent cells are divided into a bucket. The lifespan of bucket is recorded.
- (3) The approximate number of vehicles in each bucket is calculated to get the number of sketch. BPT is constructed according to buckets' information.
- (4) If buckets need to be maintained in the returning location data processing, they are updated, split, and merged by combining with the bucket constraints of AMH⁺-Sketch index. The lifespan is modified. The historical bucket is converted to UPBI-Sketch index and stored in HBase.
- (5) The aggregate query is carried out, and results are analyzed.

The purpose of this part of the experiment is to analyze the performance of historical and current probabilistic aggregate queries based on UPBI-Sketch and AMH⁺-Sketch indexes. The focus of the experiment is on the construction and maintenance of AMH⁺-Sketch indexes. Because based on AMH⁺, the AMH⁺-Sketch index intelligently divides the cell units of road sections as buckets with similar frequencies, and AMH⁺, like AMH*, is an extended version of AMH. The previous research in our laboratory [20] has verified that the DS index based on AMH* is obviously superior to the DynSketch [19] index based on AMH in terms of query time and average error. Therefore, the latest research techniques involved in the experiment are mainly focused on the comparative analysis of the performance of AMH⁺-Sketch index and DS index queries and set $\rho = 0.95$, $\eta(\rho) = 1.96$.

6.2.1. Query Range Analysis. It examines the variation trend of the query's maximum relative error (expressed as `maxrel_err` in figures) and average relative error (expressed as `avgrel_err` in figures) with the change of regional interval or time interval, and the appropriate range of query is also obtained. $\varepsilon_q = 5$ is set up temporarily, and the value of ε_q will be analyzed in the following experiments.

In the regional interval, the square query region in AMH⁺-Sketch index is represented by $n * n$. n is used to represent the number of cells crossed by the edge's length of a square query area, such as the interval value of 5, indicating the query area is a $5 * 5$ road section. A time interval represents a continuous time unit, for example, the time interval value is 5, indicating the query time: $5 * \text{unit time}$, if the *unit time* is 10 seconds, then 5 indicates 50 seconds.

As shown in Figures 7 and 8, the regional interval is 3–5, the time interval is 5–10, and the maximum relative error and the average relative error of the aggregate query are both small. The experiment shows that the AMH⁺-Sketch index is suitable for the query in a certain space. The query results conform to the long-time vehicle information can be more true to the actual traffic situation in the query area. The regional interval is 3, and the time interval is 5 in the follow experiments.

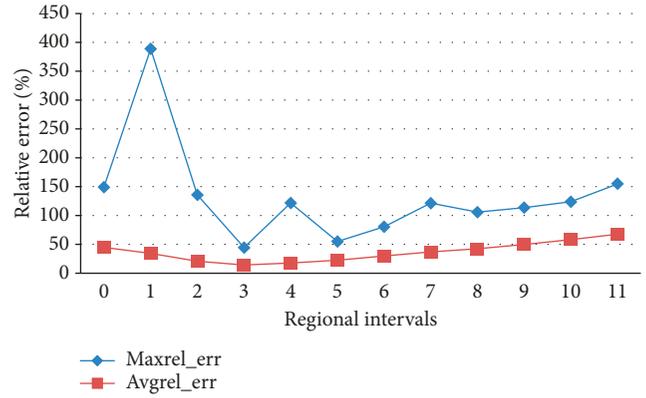


FIGURE 7: Relative error comparison among different regional intervals.

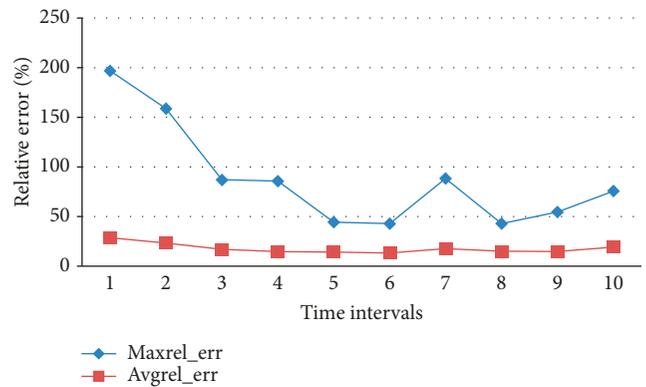


FIGURE 8: Relative error comparison among different time intervals.

6.2.2. Query Performance Comparison with DS. The purpose of these experiments is to compare the performance of AMH⁺-Sketch index with DS which also deals with aggregate query of moving objects on road network. DS does not consider the uncertainty caused by sampling frequency, so the assumption that the sampling frequency is 180 seconds is not meaningful. DS is bound to lose a large number of uncertain objects, and the AMH⁺-Sketch index costs a large amount of time for probabilistic range query. Therefore, the core AMH* technology in DS is replaced by the AMH⁺ in AMH⁺-Sketch index, which is expressed as AMH*-Sketch. It can compare the performance of them in the same problem environment. In follow-up experiments, AMH⁺ and AMH* are used to represent AMH⁺-Sketch index and AMH*-Sketch index, respectively.

(1) Average Relative Error. In this experiment, the average relative error of AMH⁺-Sketch index and AMH*-Sketch index queries is compared. According to the theorem [17], if the variance v_i of any bucket b_i satisfies $v_i \leq (\varepsilon_q^2) / (\eta^2(\rho) \cdot U)$, then for the query that conforms to $(1 \leq L \leq n(q_R) \leq U \leq n^2)$, the query's absolute error is less than ε_q under the condition of probability ρ , where $n(q_R)$ represents the number of cells in the query area q_R and L and U represent the lower and upper limits of the query range, respectively. The experiment's purpose is to analyze the accuracy of queries and

examine the effects of the parameters L and U in the AMH^+ -Sketch index.

In this case, 20–80 queries are randomly selected, with a temporary setting $\varepsilon_q = 5$. Figure 9 compares the AMH^+ constraints [17] $v_i \leq (\varepsilon_q^2)/(\eta^2(\rho) \cdot U)$ and AMH^* constraints [16] $v_i \leq (\varepsilon_q^2)/(\eta^2(\rho) \cdot n^2)$ in the AMH^+ -Sketch index, where U is set to 50, 100, and 250, respectively, in AMH^+ constraints. Figure 10 compares the AMH^+ constraint $v_i \leq (L \cdot \varepsilon_q^2)/\eta^2(\rho)$ with the AMH^* constraints $v_i \leq (\varepsilon_q^2)/\eta^2(\rho)$, which are aimed at the average aggregate query. The purpose of this comparison is to investigate the impact of L on the query. The specific meaning is no longer described, and the values of L are set to 5, 8, and 15, respectively.

As shown in Figures 9 and 10, the average relative error of AMH^+ -Sketch index decreases with the increase of U and increases with the increase of L . The average relative error of AMH^* -Sketch index is smaller than that of AMH^+ -Sketch index, but it is not very different with the smallest AMH^+ -Sketch index, indicating that the accuracy of AMH^* -Sketch and AMH^+ -Sketch index is less than 15%. The accuracy of AMH^* -Sketch is basically close to AMH^+ -Sketch index's at $L = 5$; the value of L in the following experiments is 5.

(2) *Storage Space*. Although the average relative error is relatively close between the AMH^+ -Sketch index and AMH^* -Sketch, the former has obvious advantages over the number of buckets. The constraint conditions can be found that the AMH^+ -Sketch index can dynamically adjust L and U to ensure that the average relative error can be controlled in a certain range. In Figure 11, ε_q sets different values to examine the changes in the number of buckets, in which the L takes 5, the regional intervals are 3 (the query area is a 3×3 road section), the time range is 5, and 50 queries are performed randomly.

As shown in Figure 11, the number of buckets required for AMH^+ -Sketch index and AMH^* -Sketch index decreases with the increase of ε_q . Starting from $\varepsilon_q > 1$, AMH^+ -Sketch index is obviously less than AMH^* -Sketch and the trend is more and more obvious with the increase of ε_q . Under normal conditions, the larger the number of bucket is, the more the storage space is, but the higher the query accuracy is. Figures 9 and 10 show that AMH^+ -Sketch index can adjust L and U to ensure that the query accuracy is close to AMH^* -Sketch. That is to say, when the number of bucket is obviously smaller than that of AMH^* -Sketch index, the accuracy of AMH^+ -Sketch index is basically the same. When the amount of data is increasing, the advantage of AMH^+ -Sketch index in space will become more and more obvious and query time will also be more and more superior to the AMH^* -Sketch index.

(3) *Query Time*. The purpose of this experiment is to compare the query time between AMH^+ -Sketch index and AMH^* -Sketch index under their constraint conditions $v_i \leq (\varepsilon_q^2)/(\eta^2(\rho) \cdot U)$ and $v_i \leq (\varepsilon_q^2)/(\eta^2(\rho) \cdot n^2)$. $L = 5$ is set, the regional intervals is 3, the time interval is 5, and $\varepsilon_q = 5$ is set for 20–80 queries at random.

Figure 12 shows that the average query time with AMH^+ -Sketch index is slightly lower than with AMH^* -Sketch index

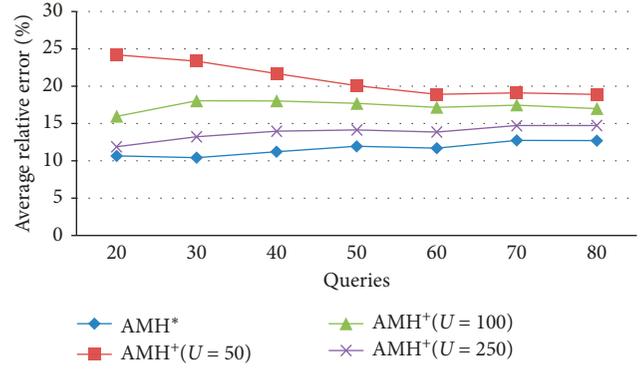


FIGURE 9: Influence of U on the average relative error.

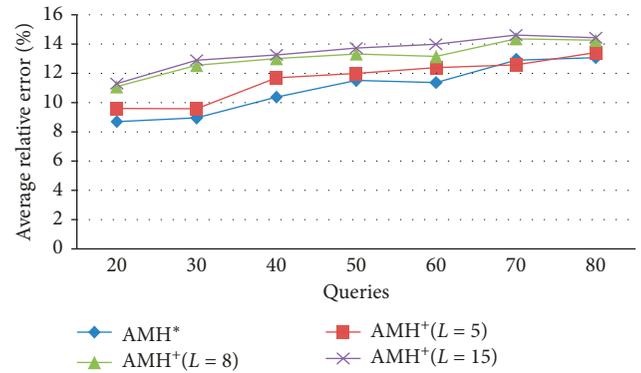


FIGURE 10: Influence of L on the average relative error.

at query time. The reason is that query time consumption based on AMH^+ -Sketch index and AMH^* -Sketch index is mainly concentrated on probability range query. With the same query conditions, the time consumption in the probability range query is close, and the difference of time consumption is not significant in the aggregate query. AMH^+ -Sketch index is superior in the number of bucket. Therefore, the query time is slightly lower than the AMH^* -Sketch index, but the difference is not obvious when the data size is small.

6.2.3. ε_q Analysis. The purpose of this experiment is to observe the maximum relative error and average relative error through change ε_q . ε_q is changed from 0.000001 to 70, $L = 5$, with the regional intervals 3 and the time interval 5. As shown in Figure 13, with the increase of ε_q , the maximum relative errors and the average relative errors of AMH^+ -Sketch index and AMH^* -Sketch index increase. The reason is that the larger ε_q is, the larger the scope of the confining condition $v_i \leq (\varepsilon_q^2)/(\eta^2(\rho) \cdot U)$ is. As a result, the smaller the constraint on the bucket, the greater the relative error. Figure 13 combined with Figure 11 shows that when ε_q is 5, the query accuracy is guaranteed by AMH^+ -Sketch index, and the number of bucket is also smaller than the AMH^* -Sketch. The spatial storage and query time also become smaller. The subsequent experimental ε_q is 5.

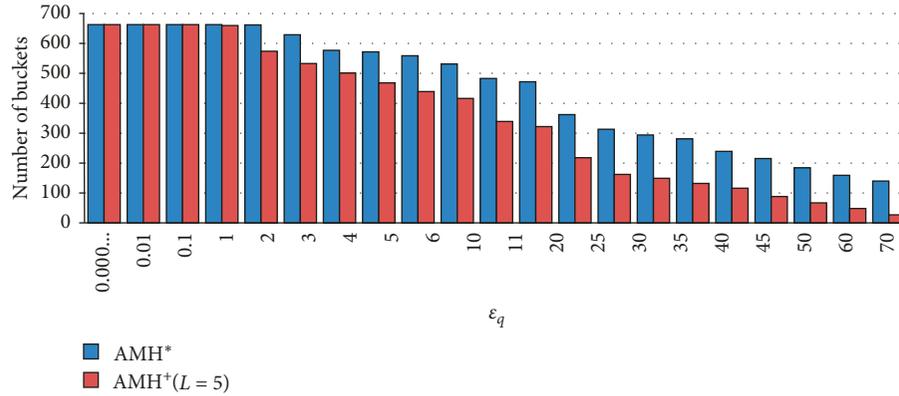


FIGURE 11: Influence of the change of ϵ_q on the number of buckets.

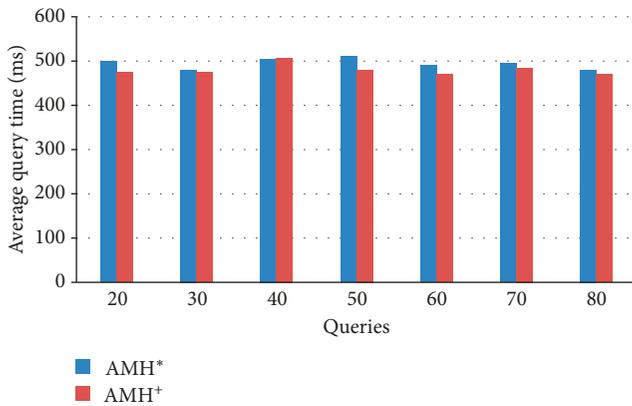


FIGURE 12: Average query time comparison between AMH+-Sketch index and AMH*-Sketch index.

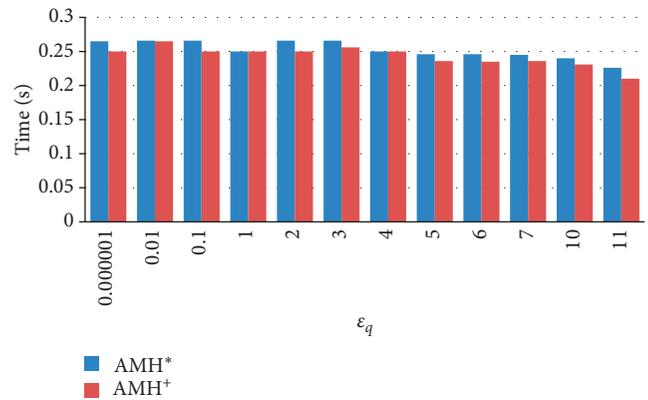


FIGURE 14: Comparison of sketch generation time.

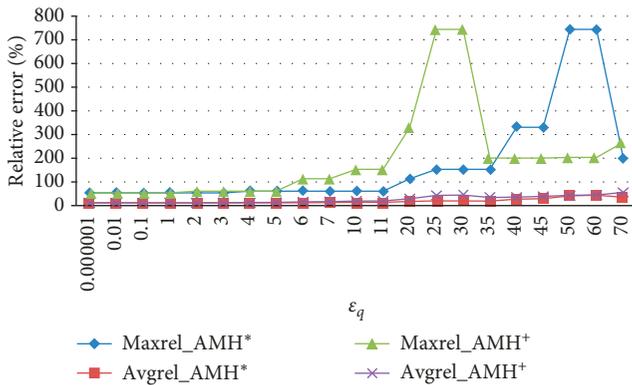


FIGURE 13: Influence of ϵ_q on relative error.

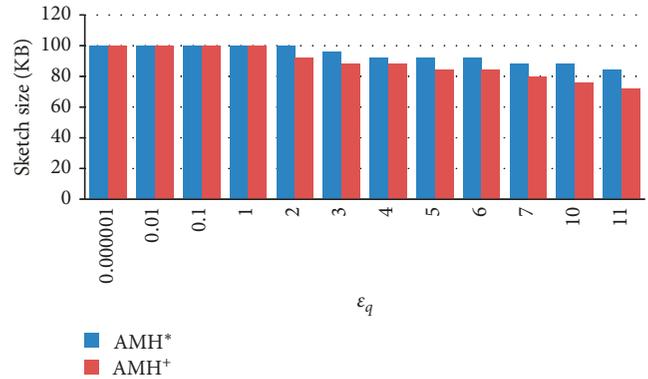


FIGURE 15: Comparison of sketch size.

6.2.4. Comparison of the Generation Time and the Size of Sketch. The purpose of this experiment is to compare AMH+-Sketch index and AMH*-Sketch index in sketch generation time and the file size under different bucket conditions, respectively. In Figure 13, the maximum relative error began to increase sharply and the average relative error began to increase gradually from $\epsilon_q > 11$. Therefore, only the case of $\epsilon_q \leq 11$ was analyzed in this experiment.

As shown in Figure 14, the AMH+-Sketch index has less sketch generation time than AMH*-Sketch index, and the difference is not large, basically around 6%~7%, about tens of milliseconds. But, in the case of large-scale data, the advantage of AMH+-Sketch index in time will become more and more obvious. In combination with Figure 11, the number of bucket of AMH+-Sketch index is basically the same as AMH*-Sketch index when $\epsilon_q \leq 1$. The number of bucket has a distinct gap from $\epsilon_q = 2$, and the gap in sketch size in Figure 15 also increases accordingly from $\epsilon_q = 2$. Considering the balance of the

number of bucket and the query precision, $\varepsilon_q = 5$. In Figure 15, the sketch size of the AMH⁺-Sketch index is 84 kb, while the sketch size in the AMH*-Sketch is 92 kb, and the gap is 9.524%. When the data are large, the memory gap will be very large too. AMH⁺-Sketch index is obviously better than AMH*-Sketch index in the space storage, and Figures 9 and 10 also show that AMH⁺-Sketch index ensures that the query accuracy is close to AMH*-Sketch index by adjusting L and U .

7. Conclusions and Future Work

The aggregate query of moving objects on road network is a hot topic in the field of moving object database. The existing methods adopt the traditional data stream approximation processing technology. However, these methods do not take the uncertainty of the trajectory caused by the sampling frequency into account, resulting in low query accuracy. This paper constructs a hybrid aggregate index framework for trajectory data considering batch processing and stream processing. The batch processing deals with the acquisition of uncertain data by the UPBI-Sketch index, and the stream processing performs the aggregate query of trajectory data by the AMH⁺-Sketch index. Based on the hybrid aggregate index framework, an aggregate query algorithm is proposed. The suitable number of the sketches inside the query area is determined to improve its precision according to the overlapping ratio of the intersecting area within the query region.

The research on the trajectory uncertainty of moving objects on road network has practical application value and brings opportunities and challenges to the field of moving object database. In recent years, many new methods based on sketch technology [45] have been proposed. At the same time, many new index structures and query methods have been designed for the stream processing mode [46]. On this basis, the index structure and query methods proposed in this paper can be improved to promote the efficiency and accuracy of the aggregate query for moving objects on road network.

This paper concentrates on the trajectory uncertainty of moving objects caused by the low sampling frequency. For each moving object, its trajectory is expressed as a probability density function $f(t)$ related to time t , making the specific time query a definite probability value. In the recent years, the spatial probabilistic temporal database (SPOT database) has begun to pay attention on the uncertainty of probability [47, 48], which will also be integrated into the work of this paper.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the General Financial Grant from the China Postdoctoral Science Foundation (no. 2016M603030) and National Natural Science Foundation of China (no. 61602151).

References

- [1] Y. Yang, Z. M. Han, and L. Yang, "Survey on key technology and application development for data streams," *Application Research of Computers*, vol. 22, no. 11, pp. 4–7, 2005.
- [2] X. F. Meng and X. Ci, "Big data management: concepts, techniques and challenges," *Journal of Computer Research and Development*, vol. 50, no. 1, pp. 1000–1239, 2013.
- [3] Y. Li, H. L. Nguyen, and D. P. Woodruff, "Turnstile streaming algorithms might as well be linear sketches," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing—STOC '14*, pp. 174–183, New York, NY, USA, June 2014.
- [4] M. Garofalakis, D. Keren, and V. Samoladas, "Sketch-based geometric monitoring of distributed stream queries," *Proceedings of the VLDB Endowment*, vol. 6, no. 10, pp. 937–948, 2013.
- [5] Z. Wei, G. Luo, K. Yi, X. Du, and J.-R. Wen, "Persistent data sketching," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data—SIGMOD '15*, pp. 795–810, Melbourne, Victoria, Australia, May 2015.
- [6] G. Cormode, "Count-min sketch," in *Encyclopedia of Algorithms*, vol. 29, no. 1, pp. 64–69, Springer, Boston, MA, USA, 2009.
- [7] T. Wellem and Y.-K. Lai, "An OpenCL implementation of sketch-based network traffic change detection on GPU," in *Proceedings of the 2012 Fifth International Symposium on Parallel Architectures, Algorithms and Programming*, pp. 279–286, Taipei, Taiwan, December 2012.
- [8] L. Chen and A. Dobra, "Histograms as statistical estimators for aggregate queries," *Information Systems*, vol. 38, no. 2, pp. 213–230, 2013.
- [9] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias, "Spatio-temporal aggregation using sketches," in *Proceedings of the International Conference on Data Engineering*, vol. 20, pp. 214–226, Boston, MA, USA, April 2004.
- [10] C. Q. Jin, F. T. Zhao, and W. B. Guo, "Towards processing aggregate queries upon spatial data," *Journal of East China University of Science and Technology (Natural Science Edition)*, vol. 35, no. 1, pp. 220–227, 2009.
- [11] D. Papadias, Y. Tao, P. Kanis, and J. Zhang, "Indexing spatio-temporal data warehouses," in *Proceedings 18th International Conference on Data Engineering*, pp. 166–175, San Jose, CA, USA, February 2002.
- [12] C. Lochert, B. Scheuermann, and M. Mauve, "A probabilistic method for cooperative hierarchical aggregation of data in VANETs," *Ad Hoc Networks*, vol. 8, no. 5, pp. 518–530, 2010.
- [13] O. Papapetrou, M. Garofalakis, and A. Deligiannakis, "Sketch-based querying of distributed sliding-window data streams," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 992–1003, 2012.
- [14] O. Papapetrou, M. Garofalakis, and A. Deligiannakis, "Sketching distributed sliding-window data streams," *The VLDB Journal*, vol. 24, no. 3, pp. 345–368, 2015.
- [15] G. Cormode, "Countin sketch," in *Encyclopedia of Algorithms*, vol. 29, no. 1, pp. 1–6, Springer, New York, NY, USA, 2014.

- [16] J. Sun, D. Papadias, Y. Tao, and B. Liu, "Querying about the past, the present, and the future in spatio-temporal databases," in *Proceedings of the 23rd International Conference on Data Engineering*, vol. 20, no. 2, pp. 202–213, Boston, MA, USA, April 2004.
- [17] C. Jin, W. Guo, and F. Zhao, "Getting qualified answers for aggregate queries in spatio-temporal databases," in *Advances in Data and Web Management*, vol. 4505, pp. 220–227, Springer, Berlin, Germany, 2007.
- [18] A. Dobra, M. Garofalakis, J. Gehrke, and R. Rastogi, "Processing complex aggregate queries over data streams," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data—SIGMOD '02*, pp. 61–72, Madison, WI, USA, June 2002.
- [19] J. Feng, C. Y. Lu, Y. Wang, and T. Watanabe, "Sketch RR-tree: a spatio-temporal aggregation index for network-constrained moving objects," in *Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control*, pp. 4–7, IEEE, Dalian, China, June 2008.
- [20] J. Feng and Z. Zhu, "Modified histogram: a spatio-temporal aggregate index for moving objects in road networks," *Procedia Engineering*, vol. 29, pp. 4135–4139, 2012.
- [21] J. Feng, Y. Q. Shi, Z. X. Tang, and C. H. Rui, "Aggregation index technique of moving objects in road networks," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 44, no. 6, pp. 1799–1805, 2014.
- [22] J. Feng, Y. Q. Shi, Z. X. Tang, C. H. Rui, and X. Min, "A novel method for predictive aggregate queries over data streams in road networks based on STES methods," in *Modern Advances in Applied Intelligence*, vol. 8482, pp. 130–139, Springer International Publishing, Berlin, Germany, 2014.
- [23] Y. Q. Shi, "A study on the complete temporal probabilistic aggregate query over moving objects on road networks," Doctoral dissertations, Hohai University, Nanjing, China, 2015.
- [24] H. Jeung, H. Lu, S. Sathé, and L. Y. Man, "Managing evolving uncertainty in trajectory databases," *IEEE Transactions on Knowledge & Data Engineering*, vol. 26, no. 7, pp. 1692–1705, 2014.
- [25] Y. Zhang, X. Lin, Y. Tao, W. Zhang, and H. Wang, "Efficient computation of range aggregates against uncertain location-based queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 7, pp. 1244–1258, 2012.
- [26] Y. Zhang, W. Zhang, Q. Lin, and X. Lin, "Effectively indexing the multidimensional uncertain objects for range searching," in *Proceedings of the 15th International Conference on Extending Database Technology (EDBT '12)*, pp. 504–515, ACM, Berlin, Germany, March 2012.
- [27] S. Liu, L. Chen, and G. Chen, "Voronoi-based range query for trajectory data in spatial networks," in *Proceedings of the 2011 ACM Symposium on Applied Computing—SAC '11*, pp. 1022–1026, Taichung, Taiwan, March 2011.
- [28] P. I. Sandu, K. Zeitouni, V. Oria, D. Barth, and S. Vial, "PARINET: a tunable access method for in-network trajectories," in *Proceedings of the 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pp. 177–188, Long Beach, CA, USA, March 2010.
- [29] P. I. Sandu, K. Zeitouni, V. Oria, D. Barth, and S. Vial, "Indexing in-network trajectory flows," *The VLDB Journal*, vol. 20, no. 5, pp. 643–669, 2011.
- [30] B. Kuijpers and W. Othman, "Modeling uncertainty of moving objects on road networks via space-time prisms," *International Journal of Geographical Information Science*, vol. 23, no. 9, pp. 1095–1117, 2009.
- [31] M. Hua and J. Pei, "Probabilistic path queries in road networks: traffic uncertainty aware path selection," in *Proceedings of the 13th International Conference on Extending Database Technology—EDBT '10*, pp. 347–358, ACM Press, Lausanne, Switzerland, March 2010.
- [32] Y. Gu, N. Guo, and G. Yu, "Uncertain moving range query techniques in road networks," *Journal of Software*, vol. 24, no. 6, pp. 1243–1262, 2013.
- [33] K. Zheng, G. Trajcevski, X. Zhou, and P. Scheuermann, "Probabilistic range queries for uncertain trajectories on road networks," in *Proceedings of the 14th International Conference on Extending Database Technology—EDBT/ICDT '11*, pp. 283–294, Uppsala, Sweden, March 2011.
- [34] 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.
- [35] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories," in *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering (ICDE)*, vol. 41, no. 4, pp. 1144–1155, Washington, DC, USA, April 2012.
- [36] H. Jeung, M. L. Yiu, X. Zhou, and C. S. Jensen, "Path prediction and predictive range querying in road network databases," *The VLDB Journal*, vol. 19, no. 4, pp. 585–602, 2010.
- [37] M. H. Abdeltawab, B. Jie, and F. M. Mohamed, "iRoad: a framework for scalable predictive query processing on road networks," *Proceedings of the VLDB Endowment*, vol. 6, no. 12, pp. 1262–1265, 2013.
- [38] M. Zhang, S. Chen, C. S. Jensen, B. C. Ooi, and Z. Zhang, "Effectively indexing uncertain moving objects for predictive queries," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 1198–1209, 2009.
- [39] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu, "Prediction and indexing of moving objects with unknown motion patterns," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data—SIGMOD '04*, pp. 611–622, Paris, France, June 2004.
- [40] S. Qiao, C. Tang, H. Jin et al., "Putmode: prediction of uncertain trajectories in moving objects databases," *Applied Intelligence*, vol. 33, no. 3, pp. 370–386, 2010.
- [41] S. Yaqing, F. Jun, and T. Zhixian, "UPBI: an efficient index for continuous probabilistic range query of moving objects on road network," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 5, no. 10, pp. 355–372, 2015.
- [42] S. Yaqing, F. Jun, R. Zhengping, and X. Wenjuan, "Hadoop-based probabilistic range queries of moving objects on road network," *International Journal of Smart Home*, vol. 10, no. 9, pp. 113–122, 2016.
- [43] FSU Computer Science, "SpatialDataset[EB/OL]," 2014, <http://www.cs.fsu.edu/lifeifei/SpatialDataset.htm>.
- [44] C. Düntgen, T. Behr, and R. H. Güting, "BerlinMOD: a benchmark for moving object databases," *The VLDB Journal*, vol. 18, no. 6, pp. 1335–1368, 2009.
- [45] G. Edward, D. Jialin, K. S. Tai, V. Sharan, and P. Bailis, "Moment-based quantile sketches for efficient high cardinality aggregation queries," *Proceedings of the VLDB Endowment*, vol. 11, no. 11, 2018.
- [46] M. Gorawski and R. Malczok, "CAM2S: An integrated indexing structure for spatial objects generating data streams," 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 33–40, Krakow, Poland, February 2010.
- [47] 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*, vol. 21, no. 1, pp. 92–107, 2009.

- [48] J. Grant, C. Molinaro, and F. Parisi, “Aggregate count queries in probabilistic spatio-temporal databases,” in *Lecture Notes in Computer Science*, vol. 8078, pp. 255–268, Springer, Berlin, Germany, 2013.

