A Correlation-Aware Partial Materialization Scheme for Near Real-Time Automotive Queries

Yu Hua
Wuhan National Lab for Optoelectronics, School of Computer
Huazhong University of Science and Technology
Wuhan, China
csyhua@hust.edu.cn

Dan Feng
Wuhan National Lab for Optoelectronics, School of Computer
Huazhong University of Science and Technology
Wuhan, China
dfeng@hust.edu.cn

***Abstract*—Real-time aggregate queries can help obtain interested summary of traffic information on the road. However, due to unreliable connection and limited duration in Vehicular Ad hoc Networks (VANETs), it is difficult to carry out the on-line computation over all received traffic messages. In order to improve query accuracy and provide quick query response, we propose a novel scheme for real-time aggregate queries, called *Road Cube*, which essentially makes use of precomputation on interested traffic messages. We utilize Information Retrieval (IR) technique to identify interested information that potentially shows semantic correlation and can be indexed in future with high probability. The Road Cube improves upon conventional data cube by exploiting semantic correlation of multi-dimensional attributes existing in received traffic information so as to obtain partial materialization. The partial materialization usually satisfies real-time and space requirements in VANETs. Extensive performance evaluation based on real-world map and traffic models shows that the Road Cube obtains significant performance improvements, compared with the conventional approaches.**

I. Introduction

Vehicular Ad hoc Networks (VANETs) serve as the infrastructure of intelligent transportation system through using no fixed devices, instead depending upon the inter-vehicle communication to support a wide of applications, such as interested content sharing, collision avoidance and emergency information dissemination. VANETs leverage short-range wireless communication, including IEEE 802.11 and Dedicated Short Range Communication (DSRC), to decrease vehicle-to-vehicle and vehicle-to-infrastructure transmission latency and facilitate data dissemination [1], [2]. Each vehicle can hence send and receive observed information, such as traffic density and free parking places, which are extracted from transmitted messages based on pre-defined format and temporarily stored into local memory to support real-time queries.

Real-time queries are much important to carry out data dissemination and reduce stale information in the VANETs since vehicles usually construct the ad-hoc network with probabilistic and unreliable connection. Specifically, a vehicle obtains the queried data only with probabilistic guarantee, i.e., when it meets with another vehicle that happens to contain the queried data, meanwhile allowing enough time to execute local indexing. Unfortunately, connection duration in the VANETs is actually rather limited and requires prohibitively high maintenance costs. Each vehicle thus has no enough time to complete instantaneous indexing and on-line computation over received messages. On the other hand, the characteristics of VANETs, such as dynamic and opportunistic configuration of entire network topology, make time-consuming on-line computation inefficient and inaccurate. Modern vehicles often contain certain amounts of local storage space, thus making precomputation and aggregation possible. In addition, VANET applications often require summarized statistic information that further makes real-time queries more challenging [3], [4].

This paper takes into account a *cube* as precomputation model due to its easy implementations and wide applications to efficiently support real-time aggregate queries. A data cube [5] makes use of multi-dimensional aggregates based on the results of off-line precomputation to satisfy the requirements for real-time queries in terms of runtime performance and summarized statistics. The data cube comes from a fact table that includes a measure attribute and a set of dimensional attributes that are associated with dimension tables. A cube consists of a lattice of cuboids, where each cuboid is associated with an aggregate of measure attribute according to a *group-by* operation on a subset of the dimensional attributes. Data cube plays an essential role in supporting fast queries and substantially increases the power of multi-dimensional data analysis.

However, performing the construction of a practical cube is non-trivial since a cube often requires too much storage space to maintain precomputed aggregates as many as possible, thus allowing most queries to obtain results within the precomputed cube and improving query accuracy. For instance, we can precompute all possible aggregates, called *full materialization*, which are certainly able to satisfy any query request, unfortunately requiring the maximum storage space. In fact, the cube-based model can be understood as a tradeoff between storage space and query performance (e.g., latency and accuracy). On the other hand, although a vehicle has certain storage space, the limited space obviously can not contain all the results of full materialization and meanwhile it is unnecessary to precompute all possible results due to dynamic configuration of network topology and unreliable connection. We thus need to precompute some limited aggregates, i.e., carrying out *partial materialization*, to allow local space to contain the precomputed results. Although some schemes of partial materialization exist [6], [7], they can not be directly applied in VANETs due to adaptive changes of available space and probabilistic queries. Therefore, we construct a *Road Cube* in VANETs to satisfy the requirements for available space, query latency and query accuracy by exploiting the semantics from received data.

The evidence of semantic correlation widely exists within transmitted data in VANETs to help identify correlated information. For example, a "`Rush Hour`" often determines the values of "`Vehicle Number`", and "`Street ID`" is always correlated with "Longitude" and "Latitude". On the other hand, due to the opportunistic and unreliable connection in VANETs, a query request may be more willing to obtain roughly correlated results with shorter latency than spend more waiting time to get perfectly accurate results. Therefore, we can store semantically correlated results from precomputation to carry out partial materialization in the *Road Cube*.

The closest work to Road Cube includes Roadcast [8], TrafficView [9], MobEyes [10], Dynamic Travel Time Map [11] and ARCube [6]. Roadcast provides a novel keyword-based P2P content sharing scheme for VANETs through relaxing user's query requirement a little and utilizing information retrieval technique to identify the relevant data, while also requiring on-line computation. TrafficView makes use of data semantics to facilitate decision making and execute data aggregation for real-time processing on limited attributes. MobEyes design serves for proactive urban monitoring by exploiting node mobility to opportunistically diffuse sensed data summaries among neighbor vehicles with the aid of low-cost index. Dynamic travel time map is a dynamic weight database to manage travel time by exploiting only spatio-temporal localities to construct data warehouse. Cube-based methods generally can support fast query services, however require prohibitively expensive costs in terms of storage and computation. ARCube consists of guiding cuboid and supporting cuboid for partial materialization to carry out a candidate generation and verification scheme.

The above work motivates our research and the proposed Road Cube improves upon them by exploiting traffic characteristics in VANETs and thus providing correlation-based partial materialization scheme. Road Cube precomputes cube measures that are stored in local memory of each vehicle to support on-line data processing and reduce query delays. We further use semantic indexing tool to identify semantically correlated data to simplify cube materialization and reduce the number of locally stored data. Cube-based design is scalable to support representation and operations on multi-dimensional attributes and this paper here takes a three-dimensional attribute set $(Time, Position, Direction)$ as example. More specifically, the contributions of this paper can be summarized as follows.

(1) Precomputation cube design: Providing aggregate queries for summarized traffic information is more attractive than simply listing fact tables in VANETs. This paper proposes a novel cube-based scheme, called *Road Cube*, which maintains results of precomputation to facilitate real-time aggregate queries as shown in Section II. The main advantages of Road Cube are to require small space overhead and provide quick query response with high accuracy in VANETs that have unreliable connection and short duration.

(2) Semantic-aware partial materialization:A cube construction essentially requires too much storage space to maintain precomputed measures. Road cube uses information retrieval tool to identify semantically correlated data with query requests to avoid prohibitively expensive costs for full materialization that is often accompanied by the well-known challenge, *curse of dimensionality* [12], in cube construction. Thus, Road Cube only needs to maintain partial but correlated data in its local memory, while ensuring these data are queried by future requests and shared by other vehicles with high probability as shown in Section III.

(3) Practical implementations We implement the road cube by mapping cube cuboids to simplified R-tree [13] nodes to fit VANET context by exploiting the properties of R-tree partitioning multi-dimensional space. We also implemented all mentioned algorithms and tested them by using real-world datasets.

The rest of the paper is organized as follows. Section II presents the road cube design. Section III describes the partial materialization scheme. We evaluate the proposed methods in Section IV. Section V shows related work, and Section VI concludes our paper.

II. ROAD CUBE DESIGN

In this section, we present the design of Road Cube that improves conventional data cube by taking into account a ranking measure to support ad-hoc aggregate queries. We describe ranking aggregates within each cuboid to efficiently answer aggregate queries.

A. Data Cube for Near Real-time Queries

A data cube [5] shows multi-dimensional aggregates and leverages off-line precomputation to obtain near real-time query performance. The data cube can present the relationship of a fact table, which contains a measure attribute and a set of dimensional attributes connecting with dimension tables.

The near real-time cube consists of a lattice of *cuboids*. In practice, each cuboid is associated with an aggregate of measure attribute according to a *group-by* operation on a subset of the dimensional attributes. Data cube can efficiently support real-time query performance via precomputation on the aggregation of all possible combination of dimensions.

A data cube is defined as a relationship $R(A,S)$, which contains p-dimensional attributes, such as, $A_1, A_2, \cdots, A_p$ and non-negative scores, S, to represent the attributes A^S ordered by S. Data cube D_R is hence allowed to have the set of aggregates to become the precomputed results via the group-by operations on R. In the road cube design, the cardinality of $p$attributes are represented as $L_1, L_2, \cdots, L_p$, and in the meantime, the cardinality of the corresponding data cube is represented as $\prod(L_i + 1)$.

A data cube can also support aggregate queries that are correlated with multiple dimensions in an *ad-hoc* manner by verifying a subset of all precomputed combinations of multi-dimensional attributes, as shown in Figure 1. Query requests receive answers by checking partial cuboids. For example, "*How many cars are there running through Street_B*?" can be answered by checking a 1-D cuboid, and "*How many cars are there running south in the morning*?" can be answered by checking a 2-D cuboid. On the other hand, these examples potentially indicate another critical problem of ranking aggregates, since each cuboid possibly contains multiple aggregates and the nearest results should be the answers.

Data cube answers top-k aggregate queries by ranking the group-by results from precomputed aggregate values and further obtaining the top-k groups.

2-D cuboid

(*time, direction,**)	(*time*,*, *position*)	(**,direction, position*)
S (morning, north,*): 620	**S** (morning, *, *Street*.B): 578	**S** (*, north, *Street*.B): 496
S (afternoon, south,*): 593	**S** (afternoon, *, *Street*.C): 350	**S** (*, south, *Street*.B): 424
S (morning, south,*): 471	**S** (afternoon, *, *Street*.B): 303	**S** (*, north, *Street*.C): 338
S (afternoon, north,*): 371	**S** (morning, *, *Street*.C): 292	**S** (*, south, *Street*.C): 321
S (evening, south,*): 46	**S** (afternoon, *, *Street*.D): 219	**S** (*, south, *Street*.D): 242
S (evening, north,*): 42	**S** (morning, *, *Street*.D): 137	**S** (*, north, *Street*.D): 131
	S (afternoon, *, *Street*.A): 92	**S** (*, south, *Street*.A): 123
	S (morning, *, *Street*.A): 84	**S** (*, north, *Street*.A): 68
	S (evening, *, *Street*.B): 39	
	S (evening, *, *Street*.C): 17	
	S (evening, *, *Street*.D): 17	
	S (evening, *, *Street*.A): 15	

Fig. 1. An example of ranking aggregates in a 2-D cuboid for the dimensions *position*, *direction* and *time*.

The ranking aggregates utilize full materialization to first precompute all possible combinations of multi-dimensional attributes and then rank them in the descending order. Storing these multi-dimensional precomputed results often require too much storage space, particularly with the increments of the number of dimensions and size of associated concept of hierarchy, thus resulting in *curse of dimensionality* [12], [14], [15]. Specifically, the concept of hierarchy means a sequence of concept mapping among different levels. For example, considering the dimension *Position*, the *Street* values can be mapped to *City* or *State* which it belongs to. On the other hand, since connection duration in VANETs generally is much limited among the vehicles, there is no enough time to support on-line computation that potentially produces severe information staleness and incompleteness. Therefore, we need to consider partial materialization to obtain significant space savings and provide fast query response within the opportunistic and unreliable VANETs.

B. Real-time Requirements from VANETs

Providing efficient real-time query services in VANETs becomes very important and necessary by alleviating or even avoiding long-delay on-line computation on received messages. A transmitted message typically consists of multi-dimensional attributes, such as its home vehicle ID, current position, speed and direction, and created time. Since the size of the message database maintained in a vehicle is usually much larger than the message size, given limited network bandwidth, this is a big challenge for improving the scalability of such a system and supporting real-time query services [10], [16]. Therefore, ranking-based aggregate query has been a popular research topic recently, where ranking is built upon multi-dimensional aggregates, instead of the measures of base facts.

We make use of the cube-based design to support real-time aggregate queries. The cube structure can provide fast responses to query requests due to its precomputed answers. A data cube can be materialized by precomputing and storing all possible multi-dimensional aggregates to facilitate on-line analytical processing. However, full materialization needs to precompute all combinations of multiple attributes, i.e., cuboids, which requires the huge amounts of storage space to keep precomputed measures. The proposed Road Cube can significantly decrease required storage space by exploiting the correlation of received data with requests' interests. We hence need to simplify the stored data and only keep partial precomputed results, i.e., semantically correlated subset. Correlated data often have higher probability to satisfy the query requests from adjacent vehicles since they potentially keep approximate spatial and temporal locality.

III. Semantic-aware Partial Materialization

This section describes the backgrounds of partial materialization and then presents the semantic correlation analysis with the aid of information retrieval tool.

A. Partial Materialization

The partial materialization in Road Cube utilizes semantic-based technique that allows to reduce over-dimensioning and inherent heterogeneity from overwhelming data and support approximate queries. Typical models include non-negative multi-way array factorization [17], estimation of the probability density [18], dynamic sample selection [19] and loglinear models [20], [21]. Unfortunately, existing methods are either highly theoretical or too complex to be used in high-speed and unreliable VANETs.

In practice, the road cube needs to carefully select precomputed aggregates that consist of some subsets of entire dataset. These selected subsets are represented as some cuboids to satisfy user-specified requests. Since user requests usually produce the checking on some correlated subsets of entire cube structure, it is naturally unnecessary to precompute all possible cuboids. For example, a traffic jam in *Street.A* possibly introduces queries for alternative routes by considering the numbers of cars in history for further prediction and decision making. In Road Cube, we first locate the cuboid that is correlated with the event and then check the precomputed subsets in the cuboid to obtain query answers.

Since a vehicle often receives too many messages, it needs to fast identify correlated data to significantly reduce the space overhead and fit the size of local memory. However, performing the identification of correlated data is non-trivial due to overwhelming arriving data and unreliable connection in VANETs. We hence propose to makes use of an information retrieval tool, i.e., Latent Semantic Indexing (LSI) [22], to help find correlated data and then rank them, some of which will be finally stored in local memory to serve for aggregate queries in Road cube.

B. Message Generation and Transmission

A running vehicle can generate messages with the aid of embedded devices to describe observed events, such as traffic jam, accidents and status of a gas station and parking place. Vehicles need to utilize cooperative approach to carry out information dissemination and support queries based on generated messages. For example, by offering direct communication between vehicles and to and from roadside units (RSUs), VANETs send and receive emergency message in the context of current traffic situation with minimal latency.

Message generation generally depends on two ways, i.e., event-driven and periodic (regular). Event-driven messages are generated when an emergency event is detected and they often contain the description of associated events, such as the position and time of an traffic accident. On the other hand, periodic messages proactively inform neighboring vehicles about status-related information, such as weather and traffic status or the position of a given vehicle. We also need to enhance network scalability to avoid a broadcast storm [23]. This is a key issue for information dissemination and scalable aggregation mechanisms can ensure that vehicles exchange their status and information in an efficient way.

Message transmission involves in the cooperation of multiple vehicles in VANETs, which often addresses geographical areas where packets may be forwarded. Existing work presents broadcast and cluster-based routing schemes. Each node using a broadcast approach, which is easy to implement, needs to broadcast messages to all of its neighbors. Although performing relatively well for a small number of nodes, its performance drops quickly when the network scale increases, leading to exponential increments of required bandwidth and possibly introducing potential broadcast storms. On the other hand, cluster-based routing method is proposed to facilitate data dissemination and information propagation in VANETs, by creating a virtual network infrastructure through the clustering of nodes. The cluster head thus may be used to perform intra- and inter-cluster communication. However, due to the uncertainty of driver behavior and fast-changing of VANET systems, the created clusters require too much overhead involved in forming and maintaining themselves and thus become too short-lived to keep information efficiently and support system scalability through using limited communication overhead.

In practice, we use both temporal and spatial constraints to decide upon message transmission. Specifically, when a vehicle generates a message, it does not broadcast immediately but has to check whether any of temporal and spatial constraints is satisfied. The temporal constraint means the waiting time for transmission, which depends on the distance of current node to the receiver. The waiting time is longer for a closer receiver to avoid broadcast storm. On the other hand, in terms of spatial constraint, we consider the amounts of aggregating messages. When the waiting time expires or the aggregated amounts exceeds, we can geocast the messages to vehicles. Finally, when a vehicle receives the messages, it needs to further aggregate and store them into local space, while possibly being forwarded to other vehicles.

IV. Performance Evaluation

In this section, we evaluate the performance of proposed Road Cube in terms of its effectiveness and efficiency, including initialization costs (time & space), query delay, query accuracy and update time,

A. Experimental Setup

Our experiments utilized the simulator based on NS-2 [24] to evaluate the proposed schemes. We have implemented Road Cube in Java and tested it on a Windows platform with Intel Core2 CPU (2.81GHz) and 2GB available memory. We derive and normalize the street layout from the map snapshots to satisfy the database specifications. All street information can be translated into text-based format that is further transformed into data format recognized by NS-2 by using techniques in [25]. The NS-2 based simulations then utilizes the extracted vehicle mobility traces. The road networks come from Network-based Generator of Moving Objects (http://www.fh-oow.de/institute/iapg/personen/brinkhoff/generator/) including two typical road networks. The San Joaquin and City of Oldenburg road networks respectively contain 38221 and 26607 traversed nodes with 630 moving data objects, which are generated on the networks uniformly with different density from 1% to 10%. Average route lengths are 157913 and 5572 respectively in San Joaquin and City of Oldenburg road networks. In addition, due to space limitation, we mainly present the experimental results from San Joaquin Road Network.

Experiments allow each vehicle to choose the shortest path to the destination, possibly resulting in uneven traffic density due to different speed limits. All experiment parameters are shown in Table I. Each vehicle initially contains its interested content. Road cube is embedded into each vehicle to store the precomputed results of partial materialization to further facilitate ranking-based (top-k) aggregate queries by setting specified I/O interfaces. When a vehicle receives a query request, the request is then issued to cube interfaces to allow aggregate query operations in Road Cube. We assume that each vehicle is able to sense the traffic information within 200m. On the other hand, we set up to 2000 query requests for at most top-20 results. The MAC layer protocol following 802.11 enables distributed coordination function. The query request pattern follows the Zipfian distribution with the parameter value 0.7. For each measurement, 20 simulation runs are used. Note that NS-2 simulator runs on the platform with 2GB available memory and however we also need to evaluate query latency with the increments of memory sizes by emulating the real environments where all vehicles totally contain over 2GB capacity. We hence utilize a snapshot-based method to make parallel emulation possible. Specifically, we first keep track of all query requests and I/O access behaviors for each vehicle, which consist of a visit-based snapshot set within total 3000s running time. The recorded sets are then used to measure the query latency for each vehicle.

TABLE I. Experimental setup.

Parameter	Values
Simulation time	3000s
Number of vehicles	600
Vehicle velocity	35-90 miles/hour
MAC layer module	802.11
Bandwidth	2Mbps
Transmission range	200m
Data packet size	3KB

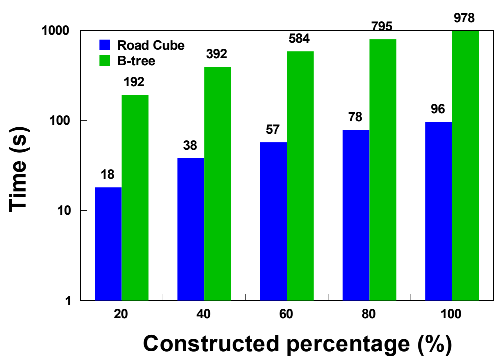
Road Cube is compared with state-of-the-art schemes, which use *Table* for no materialization and *B-tree* for full materialization. Specifically, the *Table* makes use of multiple tables to store traffic messages without any materialization. The *B-tree* [26] precomputes all combinations of multi-dimensional attributes and obtains full materialization. We further compare Road Cube with state-of-the-art work, *Roadcast* [8]. Roadcast leverages popularity-aware Vector Space Model (VSM) that extends conventional VSM by considering the impact of data popularity to identify relevant and popular data towards user's query. Note that both *Table* and *Roadcast* execute on-line computation, thus incomparable to Road Cube in terms of query accuracy. On the other hand, *B-tree* carries out full materialization by precomputing all possible results to support exact-matching services.

B. Experimental Results

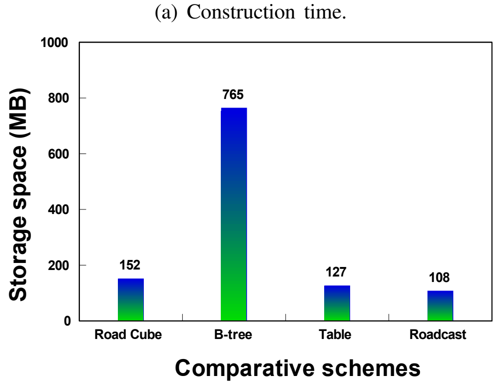
In this section, we evaluate the performance of Road Cube in terms of construction costs, query delays, query accuracy and incremental update time, compared with baseline methods of using *Table* and *B-tree* and the state-of-the-art *Roadcast*.

1) Construction Time and Initial Space: Cube-based designs need to initially carry out the precomputation on the combinations of multi-dimensional attributes, indicated by construction time and required storage space as shown in Figure 2.

We first examine the initial construction time as a metric to evaluate Road Cube and B-tree schemes. The cube construction needs to first precompute the selected cuboids and then carry out the mapping operations from precomputed cube results to physical storage space. Recall that road cube utilizes partial materialization, rather than full materialization like B-tree, to decrease precomputation time and required storage space. Figure 2(a) shows that Road Cube spends much less time to construct initial cube structure than standard B-tree approach due to using partial materialization and only precompute interested measures.



(a) Construction time.



(b) Storage space.

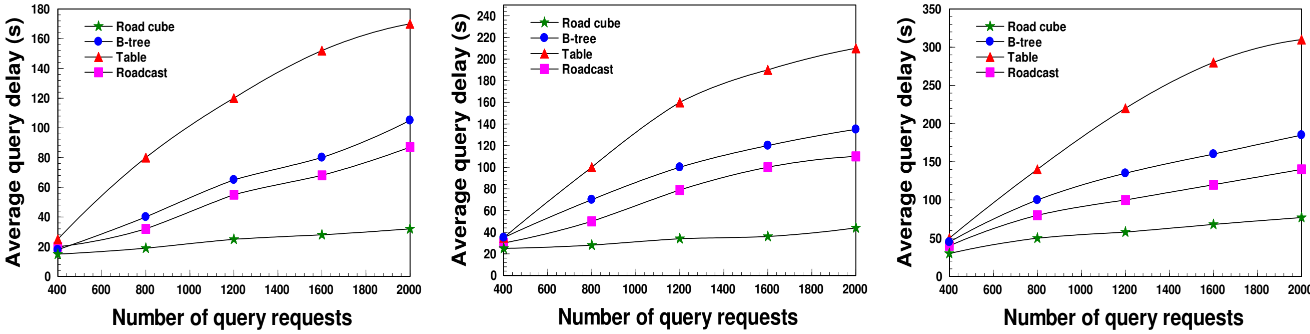
Fig. 2. Structure initialization for San Joaquin Road Network.

We compare the required storage space of *Road Cube*, *B-tree*, *Table* and *Roadcast* schemes. Figure 2(b) shows the space overheads for storing initial interested data. *Table*-based scheme has no materialization and only needs to maintain original data using table lists. In contrast, B-tree uses full materialization to pre-compute all possible combinations to support on-line queries. It hence requires too much storage space and has to transfer certain parts of the precomputed results from high-speed main memory to low-speed disks. On the other hand, we observe that Roadcast with pre-defined 75%-matching threshold requires the smallest storage capacity and the main reason is that Roadcast does not need to maintain precomputed results while using VSM-based analysis to contain interested data. Road Cube obtains as much as 85% smaller space than *B-tree* due to executing partial, not full, materialization for the constructed cube. With the increments of tested data, the significant space savings in Road Cube can also decrease the query latency from system view. Even compared with non-precomputation *Table*, no more than 19.7% space is required to maintain precomputed results. Therefore, Road Cube in essence requires limited extra storage space to support real-time query response. The limited storage space is not a critical problem nowadays, especially when obtaining the benefits from fast queries in dynamic and unreliable VANETs, which will be further verified by the results of query delays as shown in the following Section IV-B2.

2) Query Delays: Query delay refers to the time interval from initiating the query request to receiving the results. Figure 3 shows the average query delays to respectively answer top-5, 10 and 20 queries in San Joaquin Road Network while taking into account three typical attributes under 200MB available memory size used to actively satisfying on-line and off-line query services. We compare Road Cube with *B-tree, Table* and *Roadcast* schemes. *Table* and *Roadcast* in fact use on-line approaches to satisfy query requests, while *Road cube* and *B-tree* use off-line precomputation approaches. At the same time, *Road Cube* and *Roadcast* utilize correlation-based analysis to decrease required space overheads and only keep interested data in local memory.

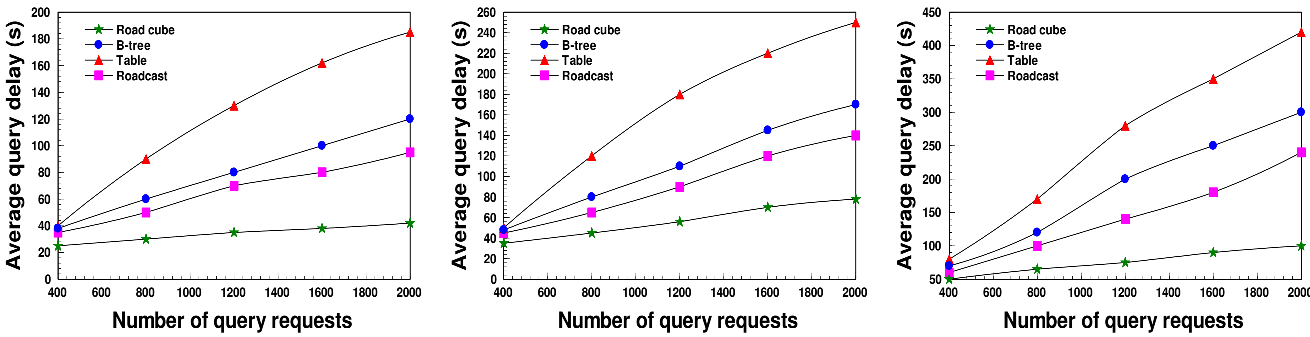
We observe that under the same conditions, off-line approaches, i.e., *Road Cube* and *B-tree*, with the aid of precomputed results can provide faster query response than online approaches, i.e., *Roadcast* and *Table*. The main reason is that *Road cube* and *B-tree* respectively carry out the partial and full materialization to precompute indexed results and thus can significantly decrease query delays. On the other hand, *Roadcast* can save up to 53.6% query time compared with baseline *Table* scheme since the former uses correlation-based analysis to guide efficient queries and the latter needs to execute on-line computation based on entire dataset. Furthermore, it is more interested that although *B-tree* precomputes all measures, it unfortunately produces longer query delays than both *Roadcast* and *Road cube*. We conjecture that *B-tree* can not buffer all computed results into local limited memory and has to transfer partial results to low-speed disks. Frequent visits on disks require extra delays and weaken the benefits of precomputation. In addition, although *Road Cube* occupies a little larger storage space than *Roadcast* as shown in Figure 2(b), it can significantly decrease query delays while the limited space overhead is acceptable. Therefore, Road Cube becomes a good tradeoff between required storage space and query delays. We also use City of Oldenburg Road Network with four typical attributes to examine the scalability of Road cube in terms of network topology changes and attribute increments as shown in Figure 4. We can obtain the same conclusion as San Joaquin Road Network.

Query delays in VANETs are tightly associated with available memory size in each vehicle. We examine the query delays with the increments of memory sizes for satisfying



(a) Top-5 query. (b) Top-10 query. (c) Top-20 query.

Fig. 3. Average query delays for answering top-k queries in San Joaquin Road Network with three attributes.



(a) Top-5 query. (b) Top-10 query. (c) Top-20 query.

Fig. 4. Average query delays for answering top-k queries in City of Oldenburg Road Network with four attributes.

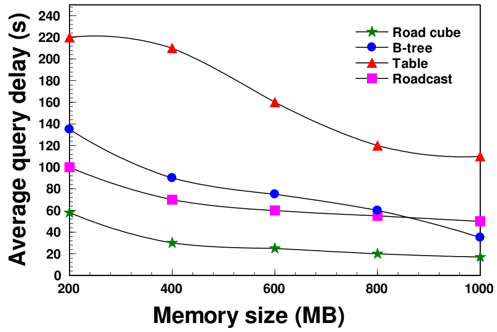
1200 top-20 query requests as shown in Figure 5. We can observe that when the memory sizes increase, query delays of all schemes decrease since each vehicle with larger space can contain more on-line or off-line computed results to quickly answer query requests. Figure 5 shows that *B-tree* is more sensitive to changes of memory sizes than others and the main reason is that with the increments of available memory, *B-tree* can place more precomputed results into high-speed memory, rather than disks, to accelerate query operations. For example, when memory size is up to 1GB, the average query delay of *B-tree* is even smaller than *Roadcast*. The results are verified by two typical network topologies respectively shown in Figure 5(a) and 5(b).

3) Accuracy of Partial Materialization: Since Road Cube utilizes partial materialization approach to decrease construction time and space overheads, it may potentially introduce some query inaccuracy due to data incompleteness and staleness. We evaluate the query accuracy by examining different top-k queries with various memory sizes as shown in Figure 6(a). Note that we obtain the exact-matching results by using brute-force approach. Figure 6(a) and 6(b) respectively exhibit query accuracy of San Joaquin Road Network with three attributes and City of Oldenburg Road Network with four attributes. We observe that more than 92.6% queries even considering 2000 requests for top-20 requests can obtain accurate results from 800MB memory, meaning that there are one or two inaccurate items from 20 found ones. On the other hand, even decreasing the memory size to 400MB, most of query requests can obtain more than 85.6% accuracy, which can satisfy real-time query requirements in VANETs. While taking into account the significant space savings and decrements of construction time, the query accuracy in the Road Cube can be accepted by most applications. In addition, both *Table* and *Roadcast* execute on-line computation, thus incomparable to Road cube in terms of query accuracy. On the other hand, *B-tree* carries out full materialization by precomputing all possible results to support exact-matching services.

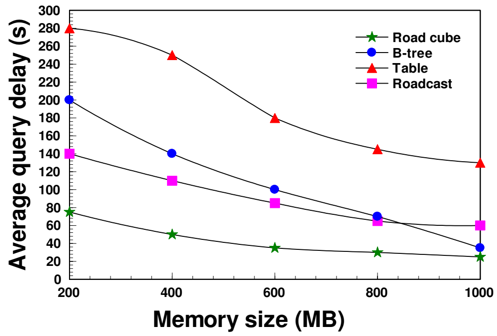
V. Related Work

Data cube, as as a relational aggregation operator generalizing group-by, crosstabs, and subtotals, was proposed in [5] to effectively support queries for decision support. It is a multidimensional extension of conventional 2-D table and presents a new operator, *cube by*, which evolves from *group by* that first partitions a relation into groups according to specified attributes values and then applies aggregation functions to each group.

Although data cube has been successfully applied in database fields, it is difficult to directly use a cube in VANETs due to limited storage space and unreliable connection. VANETs usually produce frequently changed topology and highly dynamic scale and network density, thus resulting in small effective network scope. Each vehicle serves as receiver, sender and also intermediate router [27]. Therefore, VANETs often make uses of relevance analysis to satisfy real-world requirements for comfort and safety [1], [28]. Safety-related applications potentially require direct communication due to their delay-critical nature and their messages are often labeled as high priority, meaning that the messages must be transmitted

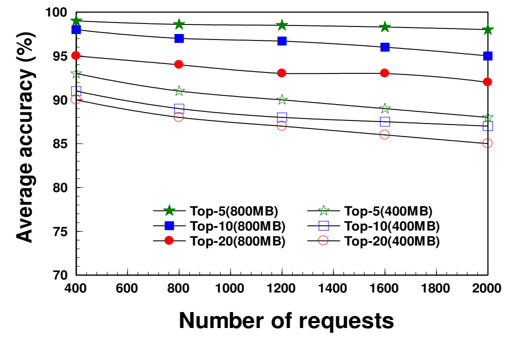


(a) San Joaquin Road Network.

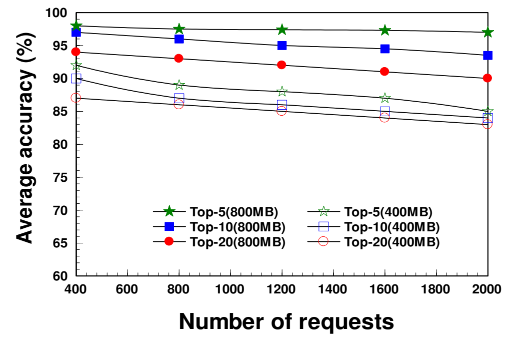


(b) City of Oldenburg Road Network.

Fig. 5. Query delays when providing different memory sizes for 1200 top-20 query requests.



(a) San Joaquin Road Network.



(b) City of Oldenburg Road Network.

Fig. 6. Query accuracy providing different memory sizes.

quickly and reliably using `push` model. Comfort-related application, on the other hand, aims to improve passenger comfort and traffic efficiency, by using `pull` model, which hopes to send the messages to vehicles that possibly are interested in.

Relevance-based approach [1] utilizes current context and the content of the messages to improve network scalability and obtain efficiency of operations both in dense and sparse networks. Opportunistic approach to resource discovery [3] can satisfy the requirements for real-time location-specific information, by using a spatiotemporal relevance function to sort resources and save only the most relevant ones in each vehicle.

VI. Conclusion

In this paper, we propose a novel precomputation scheme, called Road Cube, to support real-time aggregate aggregate queries in VANETs. To the best of our knowledge, Road Cube is the first work using data cube based scheme to provide real-time query services in VANETs. We present the ranking-based cube design in each vehicle to organize and manage traffic messages. In order to avoid the curse of dimensionality, we describe LSI-based semantic correlation analysis to facilitate partial materialization and obtain significant space savings. Extensive experimental results show the effectiveness and efficiency of our proposed schemes.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (NSFC) 61173043; National Basic Research 973 Program of China 2011CB302301; NSFC 61025008, 61232004; the Seed Project of Wuhan National Laboratory for Optoelectronics (WNLO).

References

[1] T. Kosch, C. Adler, S. Eichler, C. Schroth, and M. Strassberger, "The scalability problem of vehicular ad hoc networks and how to solve it," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 22–28, 2006.

[2] Y. Hua, L. Rao, X. Liu, and D. Feng, "Cooperative and efficient real-time scheduling for automotive communications," *Proc. the 34th International Conference on Distributed Computing Systems (ICDCS)*, pp. 134–143, 2014.

[3] B. Xu, A. Ouksel, and O. Wolfson, "Opportunistic resource exchange in inter-vehicle ad-hoc networks," *Proc. MDM*, pp. 4–12, 2004.

[4] Y. Hua, X. Liu, and W. He, "HOSA: Holistic scheduling and analysis for scalable fault-tolerant FlexRay design," *Proc. INFOCOM*, pp. 1233–1241, 2012.

[5] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 29–53, 1997.

[6] T. Wu, D. Xin, and J. Han, "ARCube: supporting ranking aggregate queries in partially materialized data cubes," *Proc. ACM SIGMOD*, pp. 79–92, 2008.

[7] D. Xin and J. Han, "P-cube: Answering preference queries in multi-dimensional space," *Proc. ICDE*, pp. 1092–1100, 2008.

[8] Y. Zhang, J. Zhao, and G. Cao, "Roadcast: A Popularity Aware Content Sharing Scheme in VANETs," *ICDCS*, 2009.

[9] S. Dashtinezhad, T. Nadeem, C. Liao, and L. Iftode., "TrafficView: A Scalable Traffic Monitoring System," *Proc. MDM*, 2004.

[10] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and Harvesting of Urban Data Using Vehicular Sensing Platforms," *IEEE Transaction on Vehicular Technology*, no. 2, February 2009.

[11] D. Pfoser, N. Tryfona, and A. Voisard, "Dynamic Travel Time Maps-Enabling Efficient Navigation," *Proc. SSDBM*, pp. 369–378, 2006.

[12] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," *Proc. ACM symposium on Theory of computing*, pp. 604–613, 1998.

[13] A. Guttman, "R-trees: A dynamic index structure for spatial searching," *Proc. ACM SIGMOD*, pp. 47–57, 1984.

[14] S. Berchtold, C. Böhm, and H. Kriegal, "The pyramid-technique: towards breaking the curse of dimensionality," *ACM SIGMOD Record*, vol. 27, no. 2, pp. 142–153, 1998.

[15] Y. Hua, X. Liu, and H. Jiang, "ANTELOPE: A Semantic-aware Data Cube Scheme for Cloud Data Center Networks," *IEEE Transactions on Computers*, vol. 63, no. 9, pp. 2146–2159, 2014.

[16] Y. Hua, X. Liu, W. He, and D. Feng, "Design and implementation of holistic scheduling and efficient storage for flexray," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2529–2539, 2014.

[17] R. Missaoui, C. Goutte, A. Choupo, and A. Boujenoui, "A probabilistic model for data cube compression and query approximation," *Proc. ACM Data warehousing and OLAP*, pp. 33–40, 2007.

[18] J. Shanmugasundaram, U. Fayyad, and P. Bradley, "Compressed data cubes for OLAP aggregate query approximation on continuous dimensions," *Proc. ACM SIGKDD*, pp. 223–232, 1999.

[19] B. Babcock, S. Chaudhuri, and G. Das, "Dynamic sample selection for approximate query processing," *Proc. ACM SIGMOD*, pp. 539–550, 2003.

[20] D. Barbara and X. Wu, "Loglinear-based quasi cubes," *Journal of Intelligent Information Systems*, vol. 16, no. 3, pp. 255–276, 2001.

[21] D. Barbara and X. Wu, "Using loglinear models to compress datacube," *Proc. WAIM*, pp. 311–322, 2000.

[22] S. Deerwester, S. Dumas, G. Furnas, T. Landauer, and R. Harsman, "Indexing by latent semantic analysis," *J. American Society for Information Science*, pp. 391–407, 1990.

[23] M. Caliskan, D. Graupner, and M. Mauve, "Decentralized discovery of free parking places," *Proc. VANET*, pp. 30–39, 2006.

[24] "The Network Simulator. [Online]," *Available: http://www.isi.edu/nsnam/ns/*.

[25] A. K. Saha and D. B. Johnson, "Modeling mobility for vehicular ad hoc networks," *Proc. VANET*, 2004.

[26] D. Comer, "The ubiquitous B-tree," *ACM Comput. Surv.*, vol. 11, no. 2, pp. 121–137, 1979.

[27] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: a mobility-centric data dissemination algorithm for vehicular networks," *Proc. VANET*, pp. 47–56, 2004.

[28] J. Jakubiak and Y. Koucheryavy, "State of the Art and Research Challenges for VANETs," *IEEE Consumer Communications and Networking Conference*, pp. 912–916, 2008.