

A Visual Reasoning Approach for Data-driven Transport Assessment on Urban Roads

Fei Wang, Wei Chen, Feiran Wu, Ye Zhao, Han Hong, Tianyu Gu, Long Wang, Ronghua Liang and Hujun Bao

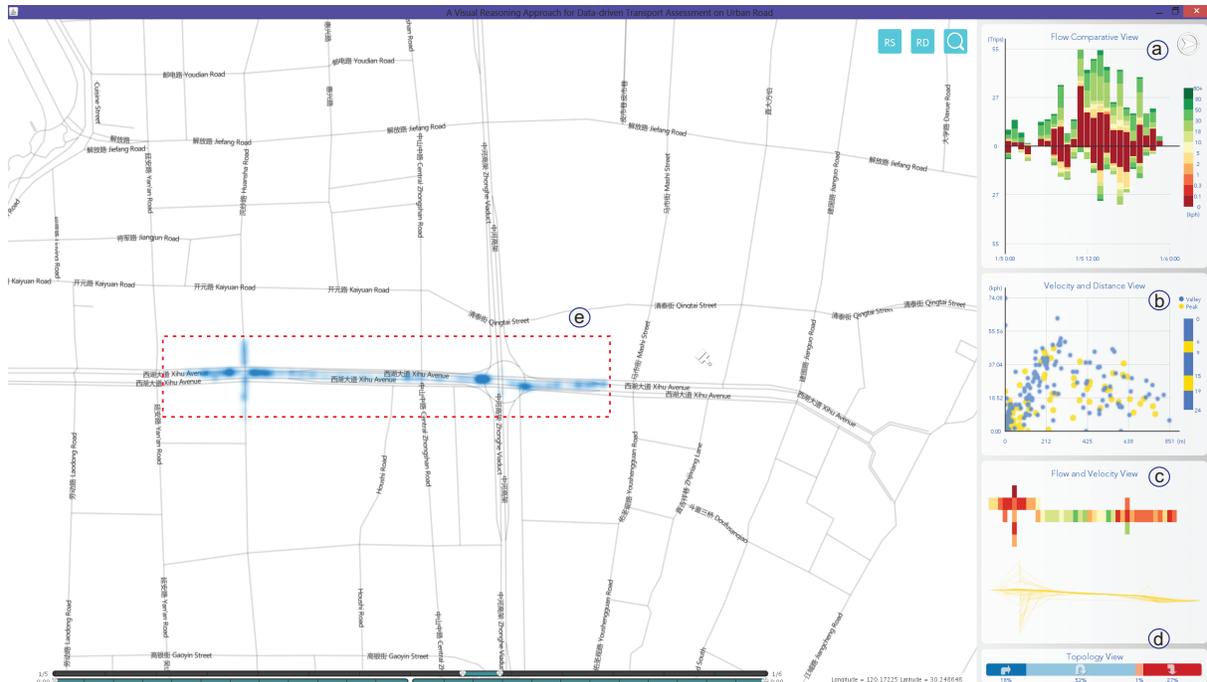


Fig. 1. Our system consists of two parts: a sketch-based query and multiple coordinated views. a) The Flow Comparative View shows the variation of traffic flow of two directions over time. b) The Velocity-and-Distance view shows the relationship between a trip's average speed and distance. c) The Flow-and-Velocity View shows the status of transportation distributed on a road. d) The Topology View shows the ratio of different flow and is also used as a topology filter. e) The Flow Density View shows the density of traffic flow distributed on a road.

Abstract— Transport assessment plays a vital role in urban planning and traffic control, which are influenced by multi-faceted traffic factors involving road infrastructure and traffic flow. Conventional solutions can hardly meet the requirements and expectations of domain experts. In this paper we present a data-driven solution by leveraging a visual analysis system to evaluate the real traffic situations based on taxi trajectory data. A sketch-based visual interface is designed to support dynamic query and visual reasoning of traffic situations within multiple coordinated views. In particular, we propose a novel road-based query model for analysts to interactively conduct evaluation tasks. This model is supported by a bi-directional hash structure, *TripHash*, which enables real-time responses to the data queries over a huge amount of trajectory data. Case studies with a real taxi GPS trajectory dataset (> 30GB) show that our system performs well for on-demand transport assessment and reasoning.

Index Terms—Road-based Query, Taxi Trajectory, Hash Index, Visual Analysis

1 INTRODUCTION

Traffic jams, unbalanced transportation capacities, and frequently occurring accidents are major problems for the road networks in modern cities. Many of these issues can be attributed to the improper planning, maintenance and traffic control. For example: 1) the density of secondary roads is too low to match that of trunk roads, which is caused by insufficient road network planning; 2) traffic flow are produced by vehicles with a large variety of moving distances and speeds; 3) roads and crossroads have mismatched transportation capacities; 4) the functional division of roads is too ambiguous [33].

Conventionally, qualitative and quantitative approaches [15] are employed to assess traffic situations on roads. In qualitative assessment, analysts examine the main factors that influence traffic to judge whether a road meets the design requirements and is adequate for its current traffic flow. On the other hand, some traffic simulation products (e.g., TransCAD [4], Cube [8] and EMME [23]) utilize quantitative

- Fei Wang, Wei Chen, Feiran Wu, Han Hong, Tianyu Gu, Long Wang and Hujun Bao are with the State Key Lab of CAD&CG, Zhejiang University.
- Wei Chen is the corresponding author. He is also with the Cyber Innovation Joint Research Center, Zhejiang University. E-mail: chenwei@cad.zju.edu.cn.
- Fei Wang is also with Hunan Normal University. E-mail: wolffyecn@gmail.com
- Ye Zhao is with Kent State University. E-mail:zhao@cs.kent.edu
- Ronghua Liang is with Zhejiang University of Technology. E-mail:rhliang@zjut.edu.cn

IEEE Symposium on Visual Analytics Science and Technology 2014
November 9-14, Paris, France
978-1-4799-6227-3/14/\$31.00 ©2014 IEEE

empirical models for road network analysis and evaluation. To study traffic situations in a large-scale region, analysts have to manually adjust many parameters in a trial-and-error process.

Data-driven methods have attracted much attention due to the rapid development of sensor and data transmission techniques. One popular approach is to use GPS-trajectory data collected by cars, which sample road conditions in a very short time interval [16]. The trajectory data represents real traffic situations, from which the statistics of traffic flow can be extracted and city-wide travel patterns can be discovered [11]. Researchers can discover city regions with different functions based on human mobility patterns and points of interest (POIs) [42], detect regions with salient traffic problems to evaluate the effectiveness of road layout by abstracting features from taxi traces [44], and predict future movements of objects using historical trajectories [46]. However, few existing work are designed for the analysts to complete their typical tasks which required interactive transport assessment.

The challenges of interactive assessment are two-fold. First, querying a huge amount of trajectories on arbitrarily chosen roads in a city can be quite time-consuming. Thus, a highly efficient scheme for trajectory data organization and query is needed. Second, discovering and analyzing potential problems on roads should be dynamic, analyst-driven, and situation-aware. Visual feedback and interaction are required in this process. To overcome the challenges, in this paper, we present a visual analysis system for analysts to evaluate real traffic situations based on taxi trajectory data. We propose a road-based query model constructed on *TripHash*, a new bi-directional linked hash-based trajectory data structure. *TripHash* is specifically suitable for fast queries and computations in local areas. Moreover, we design a sketch-based dynamic query tool that facilitates interactive and visual study of road situations and vehicle trips. Our system employs multiple coordinated views to support on-demand assessment. In summary, the contributions of this paper are as follows:

- A **dynamic road-based trajectory query model** that facilitates on-demand spatio-temporal queries of trajectories without using any textual query language. Furthermore, it supports complex topological queries [12], namely *enter*, *leave*, *cross* and *cover* of roads.
- A **bi-directional linked hash index scheme** over taxi trajectories that enables real-time query and response. One unique feature of this scheme is that free-style queries in narrow and irregular roads are made easy.
- A **visual analysis system** that incorporates domain expertise for transport analysis and traffic assessment in a situation-aware way.

The rest of this paper is structured as follows. Section 2 introduces basic concepts of transport assessment and summarizes the related work. We describe the problems and analyze requirements of transport assessment in Section 3. In Section 4, we give an overview of data preprocessing and some assumptions. Then a road-based query is described in Section 5 and Section 6. Section 7 presents several case studies using a real dataset. Finally, we conclude our work in Section 8.

2 PRELIMINARIES AND RELATED WORK

2.1 Related Concepts on Transportation Planning

According to the national standard for transportation planning and design [27], urban roads are classified into four categories.

- **Expressways** are designed for long-distance transportation and permit a high speed limit. They connect urban districts with sub-urban areas with widths of 40-60 meters.
- **Trunk roads** connect main districts in urban regions. Their widths are typically 30-60 meters.
- **Secondary roads** function as trunk roads in local regions, which are supplementary yet essential parts of the road network. Their widths are typically 20-30 meters.

- **Branch roads** connect residential communities and villages to secondary roads. Their widths are typically 12-16 meters.

For the sake of clarity, we use the term *road* to describe a street of a road network. A subdivision of a road is called a *road segment*. For megacities with a population of more than 2 million, the designated speed of the four kinds of road are 80, 60, 40 and 30 kilometers per hour respectively [27].

The place where two or more roads intersect is called an *intersection*. The capacity of an intersection denotes the maximum traffic volume in the crossing traffic flow. Capacity is influenced by several factors: the shape and area of the intersection, the number of lanes, the widths of roads, as well as the traffic control measures. These factors complicate the calculation of the traffic capacity of an intersection. We can make assumptions about certain conditions and do the calculations by means of theoretical reasoning methods [26].

2.2 Visual Analysis of Trajectory Data

A variety of techniques and methods adapted from cartography and scientific visualization are studied and applied in trajectory data visualization. Andrienko et al. [1] summarize characteristics of movement data and develop visualization methods for moving objects, spatial events and spatial and temporal distributions. Another example [29] takes time as an independent axis and displays trajectory changes with location and time in 3D space. Trajectories can be convolved to generate Kernel Density Estimations (KDEs) to give an overview at various levels of detail. Willems et al. [40] visualize vessel traffic at two levels of detail to reveal both global patterns, like traffic lanes, and local patterns, such as anchoring zones. Scheepens et al. [32] use density maps to show an aggregated overview of data with multiple attributes. Existing tools provide pleasing visualizations, but few of them filter and highlight relevant location traces on the roads. For instance, TraceViz [6] aims to filter trajectories with high similarity scores by computing the proportion of brushed points and traced points. In this work, we design a brushing tool that considers not only the proportion, but also the orders of brush points and trace points.

Visual analysis of trajectory data generally falls into two categories: macro-pattern analysis and micro-pattern analysis. In macro-pattern analysis, global phenomena, like human behavior, are presented in a large area, such as a city or a district. As for micro-pattern, trajectory data in a local area is studied to discover detailed patterns on a road or an intersection. Song et al. [36] analyze the Japanese victims' movement behaviors while seeking shelter after the leak of the Fukushima nuclear plant. Liu et al. [19] discover diversity patterns of traffic after investigating trajectory data in circular areas. Wang et al. [43] extract complex traffic jam information from global trajectory data and analyze the propagation of traffic jams at specific road segments. Guo et al. [17] study micro-patterns and abnormal behaviors of traffic flow at intersections by analyzing a dataset at one intersection. Chu et al. [7] develop a methodology to discover and analyze the hidden knowledge of massive taxi trajectory data by transforming the geographic coordinates to street names and applying topic modeling technique. T-Watcher [31] presents traffic status from overview to detail in three views: a region view, a road view, and a vehicle view. In this paper, we adopt a micro-pattern analysis of traffic flow in order to assess the rationality of transportation planning.

2.3 Query of Trajectory Data

Previous methods for trajectory data query can be classified into three categories: point query (P-Query), region query (R-Query) and trajectory query (T-Query) [24]. P-Query locates the trajectories that pass through a specific location. R-Query aims at finding trajectories that pass through certain space or time intervals. T-Query detects trajectories that share similarities with a given set of trajectories. Ferreira et al. [13] provide a visualized query tool that enables trip record queries with space, time and other attributes chosen by users. However, this approach is designed for queries over origin-destination (OD) data in large areas, not for trajectories on roads. Although nanocubes [25] can be used to explore spatio-temporal datasets in real-time, it is mainly

designed for large region queries. Our approach allows analysts to freely select roads as constraints to query similar trajectories. It is basically an R-Query based method and differs from existing solutions in that it can handle spatial regions consisting of consecutive and irregular areas.

Native-space indexing is the mainstream method for indexing spatio-temporal data, using approximations such as minimum bounding rectangles (MBRs), octagons and regular grid cells [30]. To query trajectories over a spatial division, a high resolution spatial grid can provide better matches at the expense of more computational resources.

There are two major indexing methods in accordance with trajectory data structures. Multidimensional index methods, for example, 3D R-Tree [39], STR-Trees [24], and HR-Tree [28], extend from R-Tree [18] and construct MBR based on trajectory clustering. Other approaches divide space into grids and build a time index for each grid point, e.g. SETI [5] and MTSB-Tree [47].

For large trajectories, systems like SETI and TrajStore [10] leverage segmentation to reduce the size of bounding boxes. Segments that are spatially close will be saved in the same disk partition so as to accelerate the query processing. However, both of them do not provide topology queries and are expensive for searching trajectories on a road segment. The implementation of R2-D2 [45] gathers points close to each other into the same cell and constructs a hash table within the cell to achieve a fast query of trajectory points in its neighborhood. We extend the indexing method of R2-D2 by adding the topological information of the trajectories and roads into the hash table to enable trajectory queries along connected road intervals.

3 PROBLEM STATEMENT AND TASKS

3.1 Data Description

Trajectory data used in our system was recorded by 8,386 taxis equipped with GPS devices in a time span of one month in Feb. 2012. The taxis travel in Hangzhou, a modern city in China with a population of 7 million. The number of GPS records is about 30 million and the total amount of data takes up to 80 GB. The attributes of a vehicle, including velocity, geographical position, availability and time, were sent to the data server by each GPS sensor via a GPRS link every 20 seconds. We form the trajectory of one taxi by tracking its consecutive GPS records. With no passengers onboard, drivers usually drive slowly searching for them, therefore, we exclude these records because they do not exhibit the real traffic status.

We extract the road network information from OpenStreetMap [22], a free geospatial data service. It contains roads and intersections data. All roads in the data set are defined in vectors. We reconstruct road segments in the road network data, such that two road segments only intersect at their end points.

We also use the city map and divide the urban area of Hangzhou into a rectangular grid. Each cell of the grid is represented by $cell(x, y)$, where x and y are relative coordinates from the top left.

3.2 Traffic Context

The basic element of our approach is a trip, which denotes the entire route of a taxi travelling with passengers from the source to the destination. In other words, a trip starts when a passenger gets in the taxi and stops when the passenger gets off. We define a *trip* and related terms as follows.

Definition 1 (Taxi Trip): A trip $\Gamma = (tid, O^i, p_1, p_2, \dots, p_k)$, where tid is the identification of the trip, and O^i is the i th moving taxi. $p_j (1 \leq j \leq k)$ is the j th GPS point of Γ , which consists of a geospatial coordinate set, a timestamp, and a state of occupation, i.e. $p_j = (x, y, t, s)$. In this work, the state of occupation indicates that passengers are onboard in valid trips.

Definition 2 (Flow): Flow is the number of vehicles passing a reference point per unit of time, e.g., the number of vehicles per hour.

Definition 3 (Density): Density is defined as the number of vehicles per unit length of a roadway at a timestamp.

3.3 Problem Statement

This work is carried out in collaboration with a transportation assessment expert. He describes four main tasks.

Q1: Analysis of road hierarchy

Even though transportation planning has considered potential traffic growth, traffic flow in a megacity is typically overloaded. An imperfect design of the road hierarchy may lead to an unbalanced ratio of trunk roads to secondary roads, which may cause traffic problems. The national standard [27] specifies the criteria of density and quantity of lower-tier roads and nodes, but there exist excessive numbers of vehicles in trunk roads in daily life. At locations where heavy traffic jams happen, there may be inadequate roadways that connect different functional regions in a city.

Q2: Analysis of overlapping traffic flow

There are traffic flow in both rapid speed and normal speed over long-distance or short-distance trips. If both traffic flow overlap heavily, traffic jams would happen. Essentially, our approach leverages the location, speed and availability information of taxis recorded by GPS to determine and study road-based attributes of the traffic situations.

Q3: Analysis of road and intersection traffic capacity

The transportation capacity refers to the maximum traffic flow obtained on a given roadway, if all available lanes are used. It is usually measured in *flow*. In study of traffic capacity, the intersections play a major role for two reasons. First, some intersections may divide a road into several parts with quite different traffic capacity because many cars turn into or out of this road at the intersections. Second, if the capacity of an intersection is lower than that of major roads, drivers will suffer from the mismatch of capacity between roads and intersections.

Q4: Analysis of opposite directions

Most roads have two directions with strong inner connections, and each direction sometimes has dramatically different traffic flow. During the morning rush hour, traffic lanes, whose directions are from suburbs to downtown, will have a high transportation flow, while the other half is highly unimpeded. An opposite scenario can be found during the evening peak. Once the imbalance is located, traffic control methods, such as tidal lanes, can be engaged to ease the situation.

3.4 System Design Constraints

To tackle these problems, an efficient data analysis system with a dynamic data query scheme is needed. Specific capabilities and their constraints are as follows:

T1: Integrate heterogeneous datasets. A set of heterogeneous data is involved in the assessment process including trajectory coordinates, the velocity of a taxi, availability, road networks and spatial grids. A well-designed data model is required to incorporate all these factors to handle the tasks.

T2: Build an effective data index. There are a large amount of roads in a big city, as well as a huge amount of trajectories. The evaluation should be performed in real-time and with immediate visual feedback. Therefore, an appropriate data index for the trajectory database is needed. In addition, a visual interface is preferred to reduce analysts' workload in locating, exploring and analyzing the data.

T3: Dynamic query on road sections. Trajectories on road sections are the foundation of transportation assessment. With consideration of real life situations, road sections of irregular shapes should be used as query condition. Analysts should be allowed to set the coordinates, shape and width of road sections in an intuitive fashion.

T4: Distinguish different traffic flow. On one hand, as in Q4, traffic flow in two directions may be totally different. On the other hand, taxis entering and leaving a road hold different types of traffic flow. We need to identify all of these traffic flow to provide analysts with more detailed information.

3.5 Work Flow

Our work consists of three steps: data preprocessing, road-based queries and visual analysis and reasoning. Preprocessing is conducted off-line. We extract trips from raw GPS data and store them in a 2D

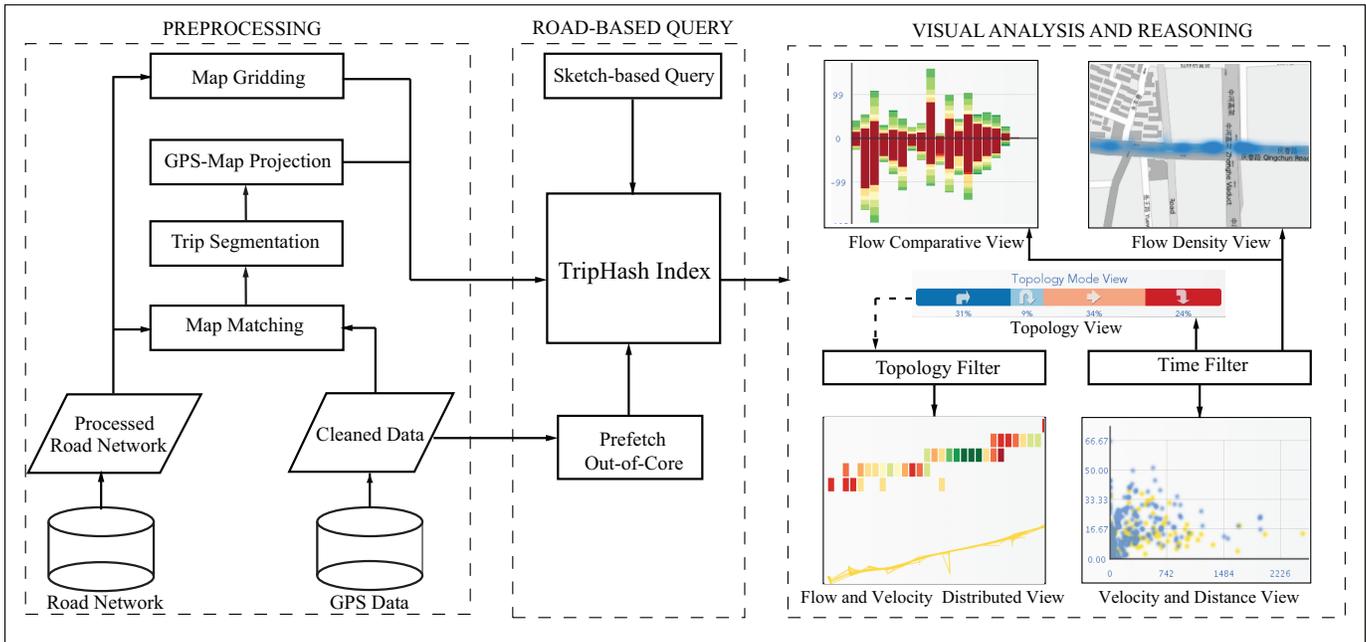


Fig. 2. The conceptual framework of our system.

hash structure. In the second step, analysts use a sketch-based visual query and a road-based query model to get results in real-time. In the third step, analysts can further make dynamic reasoning of traffic situations. Fig.2 illustrates the pipeline of our system.

We explain data preprocessing in Section 4, road-based queries in Section 5, and visual analysis and reasoning in Section 6.

4 DATA PREPROCESSING

The raw GPS recorded data is noisy, incomplete and erroneous.

4.1 Data Cleaning

The raw data is cleaned through the following processes:

- Delete irrelevant data. Some collected points actually mark locations outside the region of interest (e.g. the city).
- Delete abnormal points. If the distance between two successively time-stamped points is larger than a given threshold, e.g., the implied speed is larger than the highest speed limit (80 km/h), they are regarded as abnormal.
- Delete points with time intervals longer than 10 minutes. The reason is that 10 minutes is too long to interpolate.
- Delete GPS points with a state of *empty*. We only consider those trips containing passengers.

After cleaning, the raw data is reduced from 80 GB to 38.2 GB. We use a simple out-of-core algorithm to prefetch data from disks and trips recently queried will be loaded into memory.

4.2 Trip Segmentation

Trips are created by linking cleaned GPS points. It is typical that the time intervals between two consecutive GPS points fall in a range of 15 to 25 seconds. If there is a larger time interval, a linear interpolation is performed along the matched road to add samples. In practice, the ratio of added samples is less than 2%.

4.3 Map Matching

Matching the trajectory data to the road network is essential for query and analysis. There are a number of studies on matching GPS samples on a digital map. These approaches can be generally classified into three classes: local/incremental methods, global methods, and statistical methods. The local/incremental methods try to find local match of geometries. When matching a new position, it only considers a small portion of the trajectory that is close to the position. In this work, we adopt Greenfold's [14] approach to match samples on most parts of the map except intersections, since it works well on a high sampling rate (e.g. 20 seconds/point). When a taxi turns at an intersection, two consecutive sampling points may belong to two roads. Directly connecting them would create a mismatch of the trajectory over the intersecting roads. To address this challenge, we calculate the locations of intersections in the city from the road network data provided by OpenStreetMap. After obtaining the coordinates of intersections, these coordinates are used to interpolate points when a taxi enters an intersection. Such an approach helps create a closely matched trajectory by minimizing the deviation of trip points from roads.

5 ROAD-BASED QUERY

To fulfill the analysis tasks, we design a query model and an efficient index scheme for dynamic querying of trips on road networks.

5.1 Road-based Query Model

A typical search on sets of taxi trips is based on a given road section and a time period, e.g. *finding all trips passing a given road between 6:30 a.m. and 8:30 a.m. in a week*. Our new query model is based on road maps and differs from conventional searches conducted by query languages over spatial and temporal databases. Definitions within the query model, namely *road-based query model*, are given below.

Definition 4 (Region of Query): A region of query, R , is a collection of grid cells on the map, which approximates a road segment.

Definition 5 (Topology): A topology mode of trips, ω , describes the relationships between moving objects and roads, such as *enter*, *leave*, *cover* and *cross*, as shown in Fig.3.

The topology mode is an important indicator of the status of road sections. For example, if there are more cars leaving than entering a road in a time period, the traffic pressure of this road is relatively low.

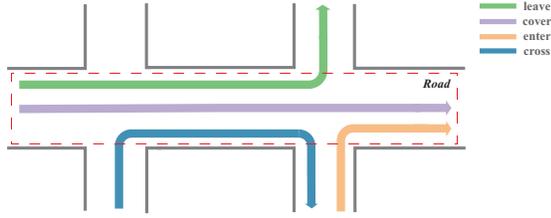


Fig. 3. The trips passing a road exhibit four topology modes.

Definition 6 (Road-based Query): A road-based query $Q(\Gamma, R, \omega, t)$ performs a search on a given spatial region R (road segment) and a temporal interval t , and returns a corpus Γ consisting of trips, Γ_j , that exhibits a topology mode ω during t . Here, $\Gamma_j \subset \Gamma, R \times \Gamma_j \in \omega$.

The road-based query is different from the conventional R-query for two reasons. First, a region of the road-based query is typically an irregular-shaped road segment that consists of consecutive and small areas. However, the R-query involves a large and regular area, e.g. one city block or more. Second, the road-based query includes four topology modes. In contrast, topology modes are generally neglected in the conventional R-query.

5.2 Index Scheme

The general idea of indexing trips is to constrain the trips in the road network. We further design the Trip Grid (TG) as a container which stores recent trips. We use this grid to divide the area of interest into a set of rectangular regions, namely cells, with fixed width. GPS points of trips are projected to the grid. The size of a cell is determined by both the width of roads and the average speed of taxis, and will be explained in Section 6.1.1. We use $cell(x, y)$ to denote a cell at the position (x, y) in the grid.

When a trip passes a $cell(x, y)$, GPS points located in this cell will be inserted into $TG(x_i, y_i)$. We employ a hash table, called *TripHash* (see Fig.4) to facilitate an efficient search of trips over TG . Data stored in the *TripHash* of each cell is indexed by a key, composed of a moving object ID O^i and a timestamp t . It means O^i passes this cell at time t . The value of the hash index in the *TripHash* includes three parts: the GPS point ID pid , the coordinate of this point's previous point in the trip, and the coordinate of its next point. In this way, we store all the moving objects' trips in the *TripHash* table, and all points of a trip are bi-directionally linked. Knowing a trip ID O^i in $cell(x, y)$ at time t also enables us to retrieve the taxi's consecutive segment or previous segment of a trip. This is convenient for querying a fragment of a trip or the holistic trip. To achieve higher performance, we use a B+ tree to index timestamps for temporal query.

Example 1. In Fig.4, O^i starts from cell(3,2) at time t , and then moves to cell(5,4) at time $t+1$. We insert a record into the *TripHash* of cell(3, 2) with a key (O^i, t) and a value $[pid, NIL, (5, 4)]$. We set previous pointer as NIL because cell(3,2) is the first cell of this trip. Similarly, we insert the records into cell(5,4), cell(5,7), and cell(7,7).

5.3 Trip Query

Here we describe how to retrieve all vehicles in *enter*, *leave*, *cover* and *cross* topology modes on a road. If O^i has a sub-trip Γ_j that matches the given road segment R in a time window $\tau = [t_{begin}, t_{end}]$ and meets a topology mode ω , then O^i is contributed to evaluating R . We collect all valuable trips in different topology modes to analyze the traffic flow on a road.

In addition, there are two schemes for retrieving trips which pass a road segment. The first one returns the GPS points that are in the underlying road, as indicated in Algorithm 1. The second one returns points of the entire trips even though they may be beyond the road, as explained in Algorithm 2. The latter is useful when analysts make comparisons between long-distance trips and short-distance trips (Q2).

Find Trips In a Road Given a road segment R , a time window τ and a topology ω , Algorithm 1 returns a collection of trips which meet the spatial, temporal and topological conditions. First we need

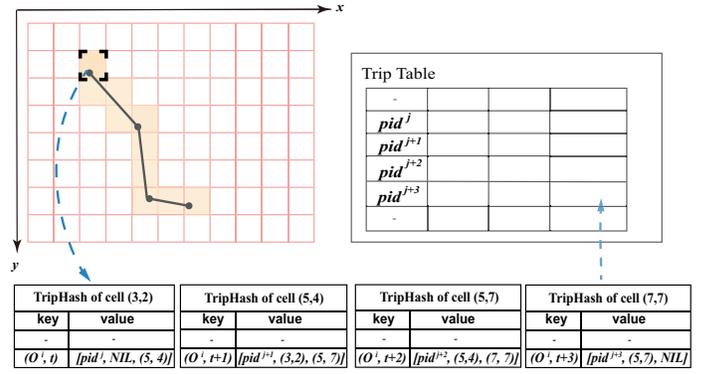


Fig. 4. The TripHash structure connects trips in bi-directional links.

to scan the TripHash at $cell(x_0, y_0)$ to query all trips within τ , where $cell(x_0, y_0)$ is the first cell of R . We conduct both forward and backward traversals to each trip Γ_j , until Γ_j no longer exists in R . If Γ_j meets ω , we add it in the result collection Γ . Then we continue to fetch other trips at $cell(x_k, y_k)$ in R .

Algorithm 1: FindTripsInRoad

Input: A road segment $R = \{(x_i, y_i), 0 \leq i \leq n\}$, a time window $[t_{begin}, t_{end}]$ and a topology ω

Output: Γ

$\Gamma = \Phi;$

for $i = 0; i \leq n; i++$ **do**

$table = TG[R.x_i, R.y_i];$

find $entryset$ from $table$ during $[t_{begin}, t_{end}];$

for *each entry* in $entryset$ **do**

$cell = entry.value;$

$link = \Phi;$

/* traverse a trip to the tail */

$link = link \cup traverse(cell.next);$

/* traverse a trip to the head */

$link = link \cup traverse(cell.prev);$

$trip.nodes = link;$

if $trip = \omega$ **then**

$\Gamma = \Gamma \cup trip;$

return $\Gamma;$

Example 2. In Fig.5, selected cells $[cell(j, k), cell(j+3, k)]$ of a road are in grey from left to right. O^i is a trip in cover mode, with orange arrowed lines linking each point at consecutive timestamps. We use O_t^i to denote the position of a moving object O^i at time t . O_t^{i+1} travels the road in the opposite direction in leave mode during $[t, t+3]$. If we use Algorithm 1 and set the time interval as $[t, t+4]$ with the topology mode as leave, we get a segment of trip as $\{O_t^{i+1}, O_{t+1}^{i+1}, O_{t+2}^{i+1}\}$.

The first cell (j, k) is searched in $[t, t+4]$ and only O_t^i is returned. Then, we search the next point of O^i at $cell(j+1, k)$, O_{t+1}^i , on the selected road segment and mark it. We continue to traverse the trip of O^i until the end of time or O^i surpasses the road segment. Although the first position of O^i is $cell(j, k)$ and the last position is $cell(j+3, k)$, this trip does not meet the topological leave mode, so we have to release this trip. Next, we go to the second cell $(j+1, k)$ and find O_{t+2}^{i+1} , because O_{t+1}^i has been marked. We traverse O^{i+1} in both directions and it returns $\{O_t^{i+1}, O_{t+1}^{i+1}, O_{t+2}^{i+1}\}$. Because O^{i+1} starts from $cell(j+3, k)$, one end of the selected segment, and leaves at $cell(j+1, k)$, this trip is regarded as a valid one. If Algorithm 2 is invoked, the entire trip of O^{i+1} , including O_{t+3}^{i+1} , will be returned.

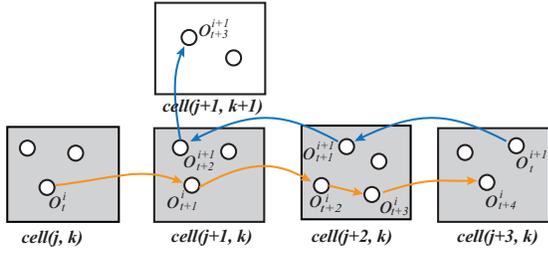


Fig. 5. An example of the look up procedure in Algorithm 1. The selected road consists of grey cells, and two trips are in different topology modes as well as different directions.

Find Entire Trips This algorithm is an extension of Algorithm 1 for calculating the entire distance of a trip exceeding a road segment. We have to find all points of a trip passing R . First, we need to execute Algorithm 1 to find all segments of trips Γ on R . For each trip in Γ , we get its first point and last point. Then, we traverse the entire trip from the tail to the head.

Algorithm 2: FindEntireTrips

Input: A road segment $R = \{[x_i, y_i], 0 \leq i \leq n\}$, a time window $[t_{begin}, t_{end}]$ and a topology ω

Output: Γ

$\Gamma = \text{FindInsideTrip}(R, [t_{begin}, t_{end}], \omega)$;

for each trip in Γ do

$fc = \text{trip.first}$;
 $lc = \text{trip.last}$;
/ traverse a trip to the tail */*
 $link = link \cup \text{traverse}(\text{cell.next})$;
/ traverse a trip to the head */*
 $link = link \cup \text{traverse}(\text{cell.prev})$;
 $\text{trip.nodes} = \text{trip.nodes} \cup link$;

return Γ ;

TripHash is different from the segmentation-based approach, such as SETI. SETI subdivides trajectories into segments and groups them into a collection of spatial partitions, then runs queries over just the spatial partitions that are most relevant to a given query. However, the spatial partition yields a coarse approximation to the trajectories, hence a significant runtime is incurred in eliminating the false hits (see Section 7.1). TripHash is point-based and approximates trajectories more precisely than SETI. Queries of TripHash are in a local area and only trajectories on roads will be returned during analysis.

6 VISUAL QUERY OF ROAD-BASED TRAFFIC

We design a visual interface that supports road-based visual query, trajectory data analysis and data-driven transport assessment. We display traffic flow on each road to accomplish four tasks proposed by the collaborating expert in Section 3.3. As shown in Fig.1, the main interface shows the map and the sketch-based query on the left and the statistical charts on the right. We first describe the visual encoding schemes for the interface (Section 6.1). Then we explain how our system supports dynamic queries and detailed road-based traffic analysis (Section 6.2).

6.1 Sketch-based Dynamic Query

We provide a brushing tool to allow analysts to filter trajectories on a road. We split the map into a grid of the same size as the trajectory segmentation. If a road is brushed, the coordinates of grid cells will be recorded as spatial constraints of a road-based trip query.

6.1.1 Cell Size

It is necessary to optimize the cell's size ξ in designing the brushing tool because we want to balance performances of time and space. That is to say, we make cells as small as possible to reduce unnecessary query regions; on the contrary, we want to make cells big adequately to cover the majority of a road in width. The rule is to multiply the taxis' average velocity by half of the sampling time interval of the trips. In our GPS dataset, the average speed of a taxi is about 11m/s, and the sampling time interval is 20 seconds, thus $\max(\xi) = (11 \times 20) \div 2 = 110m$.

On the other side, the maximum width of roads in a city is 60 m. We adopt $\xi = 110 \div 2 = 55m$, for the purpose of covering a road segment with the brushing tool.

6.1.2 Query Processing

Brushing on Roads We design a brushing tool that allows analysts to sketch on roads. When a cell of a road is on the stroke, the brush will be constrained to the road. We set the width of the brush to two times that of cell size. Three possible positions between the roads and the grids are shown below. For the cell labeled No.1 in Fig.6, the road covers most of the cell, while only a small part of cell No.2 is covered by the road. Cells will be recorded if the distance between the center of the road (see the dashed blue line in Fig.6) and a cell's center is less than $55 \times \sqrt{2} \div 2 = 38.9$ meters. For cell No.3, whether or not it will be stored depends on the orientation of brushing. If the brush turns to the opposite direction of cell No.3, its position is not counted. On the other hand, if the brush stays on the original direction, cell No.3 will be treated as cell No.2.

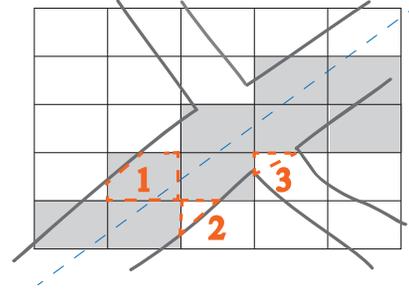


Fig. 6. Three relationships between cells and a road while brushing. The blue line denotes the medial road axis.

Spatial and Temporal Filter When using the spatial filter, a road segment will be selected and traffic flow on this segment will be highlighted. When a temporal filter is used, traffic flow on the road during a time window will be updated. By using both filters, analysts can locate the spatial and temporal range.

6.2 Visual Design

To analyze transportation situations on roads, it is desirable to show trip attributes, geographical and temporal information. For a selected road, we would like to show the traffic flow in time periods, including vehicle speed and the amount of vehicles with different speeds. It gives an overview of whether a road is in a good situation. Then we need to display the relationship between traveling distance and vehicle speed. In addition, we want to present the density of traffic flow on a road. The density measures the total number of vehicles at different parts of a road. Most roads have two directions with different traffic flows. We would like to compare the flow on both sides in one view.

Flow Comparative View To illustrate the variation of traffic flow on a road in a given time period and compare the transportation of forward and reverse lanes of the same road segment in different time periods, we implement a flow comparative view by adopting stacked bar chart (see Fig.1(a)). First, we count the number of vehicles in different speed ranges within a time slot and encode each slot as a colored vertical bin. Vehicles in the same time slot are stacked in one bin. The

speed ranges are color-encoded from red to green. We further split the bar chart into two parts corresponding to the two directions of traffic flow, which are separated by a horizontal line in the middle. The upper part represents the flow of forward lanes and the lower part represents the flow of the reverse lanes. The size of the time slots defines the temporal granularity which is determined automatically according to the time window (t) being studied. The rules of setting temporal granularity are as follows:

- the granularity is a week, if $t \geq 14$ days;
- the granularity is a day, if $t \geq 28$ hours and $t < 14$ days;
- the granularity is an hour, if $t \geq 4$ hours and $t < 28$ hours;
- else, the granularity is half an hour.

For task Q4, this view can compare traffic situations on roads with two directions.

Velocity-and-Distance View We design this view to visualize three important trip attributes, i.e., the distance of trips, the average speed of trips, and the trips in peak hours and off-peak hours. A scatterplot is employed in this view. The horizontal axis represents the distance of trips, and the vertical axis is the average speed of trips (see Fig.1(b)). We further encode trips in-peak hours and off-peak hours with different colors on the scatterplot. To reduce visual clutter, we employ a kernel density estimation (KDE) method. This view is designed to help experts identify different situations of roads. For example, if there are many trip points located at the lower right, the road is in a bad situation as many trips with a long distance suffer from slow speed. It is an essential view demanded by task Q1 and Q2.

Flow-and-Velocity View In this view, we mainly consider the traffic flow variations in the spatial dimension. We encode the amount of trips, speed and the taxis' positions of a brushed road in a block-based map (see Fig.1(c)). The block-based map spans from left to right corresponding to the direction of the brush over the filtered road. If the brushed road is not in an east-west direction, it will be rotated to an appropriate angle. The angle is set depending on the first point and the last point of a stroke. The width of a block indicates the number of cars (set as 20) passing a location within a time interval. If there are 2 blocks at one position, it means that about 40 taxis passing by. The color represents the average speed of taxis at the place. In this view, we further encode trips of different topology modes with different colors in a line graph below the block map. If one topology mode is selected, the related lines are highlighted in the line graph. This view provides information of traffic flow status at each position of a road, which is essential for the analysis of road and intersection traffic capacity proposed in task Q3.

Topology View To recognize the characteristics of a road, we represent the ratios of the trips with different topology modes by four different colors and labels (see Fig.1(d)). Each part of this view denotes a mode from left to right, i.e., leave, cover, enter, cross. Its width represents the ratio of trips with this mode among all trips. This view is designed for the study of traffic capacity for task Q3, since the proportions of traffic flow in different topology modes can reflect how an intersection affects connecting roads.

Flow Density View This view shows the spatial distribution of traffic flow on a city map. Analysts can find locations where jams tend to happen easily based on this view. KDE is used to show the density of traffic flow. The density factor \hat{h} of a location can be calculated using the formula:

$$\hat{h} = \frac{1}{T} \sum_T \phi(n) \quad (1)$$

where T is the given time interval and n is the number of trips passing the location.

There are two major advantages of our visual system compared with existing solutions. First, our system is a micro-pattern method and gives detailed views of traffic flow. By using the brushing tool, our

system can provide a local view of one road or one part of a road. Statistics are only related to trajectories on the selected road with the improved resolution. Second, our system can be used to analyze a large region in iterative exploration. It supports arbitrary selection of the majority of roads in a city. Interested road segments in any shape can be brushed as well.

7 EXPERIMENTS

We have conducted our experiments over a large volume of data. In this section, we first present a performance evaluation by comparing our query model with SETI and 3D R-tree. Then we perform three case studies to demonstrate how our method can effectively assist domain experts in performing analysis and seeking traffic problems in Hangzhou, China. The domain expert was trained to use the system before the analysis is performed in the case studies.

7.1 Performance Evaluation

The experimental platform is an Intel Xeon ES430 2.66 GHz desktop that is configured with 16 GB of main memory and a 3 TB 7200 RPM Seagate disk, running Windows 8. We use the real trip data of 500 taxis in one day with about 0.5 million GPS points. In the experiments, we use two types of road-based queries: a query of long distances and a query of short distances in the same time interval.

We use two performance metrics. The first is the overall run time. Visual analysis systems often require quick responses to experts' strokes. The second is the memory consumption of an indexing data structure. Since the volume of the trajectory data is large, we cannot load all data into memory. The size of an indexing data is a key factor in prefetching data. We compare TripHash with SETI and 3D R-tree. We implement SETI and 3D R-tree using the open source XXL [2] library for better performance. The size of SETI's cell in the query of long distances is set much larger than that in the query of short distances. Fig.7 shows the runtime and the size of indexes for distance of 2 KM and 20 KM. TripHash performs the best of the short distance, both in terms of the size of the index and the run time. It is exhaustive for SETI and 3D R-tree to search in such a small area. Though TripHash calls for more space to cache the index in the long distance query, it provides the quickest query response.

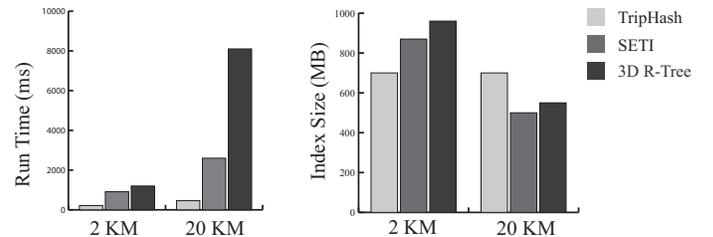


Fig. 7. Performance comparisons of TripHash, SETI and 3D R-tree

7.2 Case 1: Traffic Jams on Expressway Influenced by Other Roads

Desheng Road is an east-west elevated road in Hangzhou with high normal speed in off-peak hours. The speed of most vehicles in off-peak hours is much higher than in peak hours and the travelling distance in off-peak hours is longer as well (see Fig.8(c)). However, when analyzing this expressway, we can find from Fig.8(d) that the speed of vehicles drops significantly at the intersection with S3. In the peak hours, there is a high density area at each direction of S3 through the Flow Density View (Fig.8(a)). At the same time, we see that at 9:00 A.M. and from 5:00 P.M. to 8:00 P.M., traffic jams are more serious (Fig.8(b)). In the Topology View (Fig.8(d)), the cross trips have a higher percentage (71%) than the other modes on this road section. It shows that most vehicles want to leave or enter the expressway and the congestion on the expressway is caused by traffic jams on the intersecting road.



Fig. 8. Traffic jams on Desheng Road start at the intersection with S3, where the overflow traffic cause congestion on the expressway.

7.3 Case 2: Watershed and Mismatching Analysis of an Intersection

Qingchun Road, one of the most important east-west roads in Hangzhou, runs across the city center and conveys very high traffic flow. One major intersection of Qingchun Road consists of several upward and downward ramps from Zhonghe Elevated Road.

We first select Qingchun Road in a west-east direction (see Fig.9(a)), by setting the endpoint of the intersection with Zhonghe Elevated Road. We can see from Fig.9(c) that the cross and leave cars take up a total percentage of 33%, while the enter and cover cars take up 65%. Then we select the road again by setting the endpoint of the selection after intersection with Zhonghe Elevated Road (see Fig.9(d)). The share for cross and leave cars jumps to 70% (see Fig.9(f)). This pattern indicates a large shift in the traffic flow before and after the intersection. A large number of cars turn into the elevated road, leaving the east part of Qingchun Road after the intersection much more unimpeded than the west part before the intersection. We can also find that more taxis run from east to west into the city center than those from west to east in the morning peak hours (see Fig.9(b, e)). In fact, such intersections, or so-called *watershed-like* intersections, exist in most long roads running across both the downtown and the suburban. The *watershed* pattern may lead to the improvement of traffic lights with better turning signals and cycle times.

We also find that there exist two points with a very high density of flow at both the west and east parts of the intersection (see Fig.9(d)). It indicates long waiting time of all drivers suffered from the low capacity of the intersection.

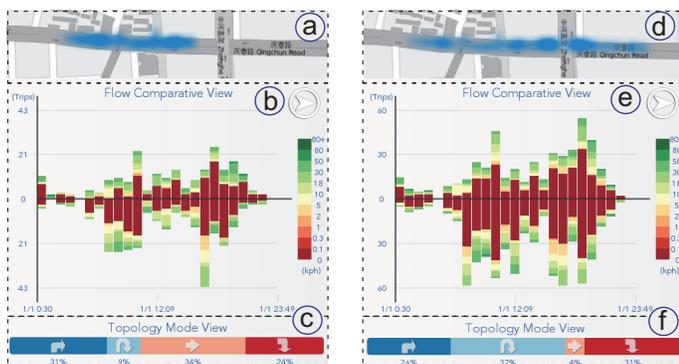


Fig. 9. A comparison between the results of two queries shows a low-capacity *watershed-like* intersection of a trunk road. The left column is the first query and its results, while the right column is the second query and its results.

7.4 Case Study 3: Using Opposite Directions to Analyze Reversible Lanes

Tianmushan Road connects a series of important office buildings, malls, hospitals to the residential areas of the Xihu District. There is a very high daily traffic flow of east bound in the morning, and a high west bound traffic flow in the evening rush hour. Meanwhile, the lanes of the opposite direction always have a lower occupation in the corresponding hours. Such roads are considered having a *tide-like* pattern, which is reflected by an unbalanced distribution above and below the middle horizontal line in Flow Comparative View (Fig.10(b)). In contrast, a parallel road next to Tianmushan Road, Shuguang Road (Fig.10(d)), has the similar *tide-like* patterns, which however shows a much more balanced view in Fig.10(e). The different transportation capacities of the two roads can be attributed to the implementation of a reversible lane. In Nov.11th, 2008, the city government assigned one of the five lanes of Shuguang Road as the reversible lane used for different directions in corresponding rush hours. Through our visual analytics views of Fig.10(b) and (e), it is confirmed that the use of the reversible lane is successful. This is further justified through the comparative views of Fig.10(c) and (f), which indicate that drivers can run faster and longer on Shuguang Road than on Tianmushan Road.

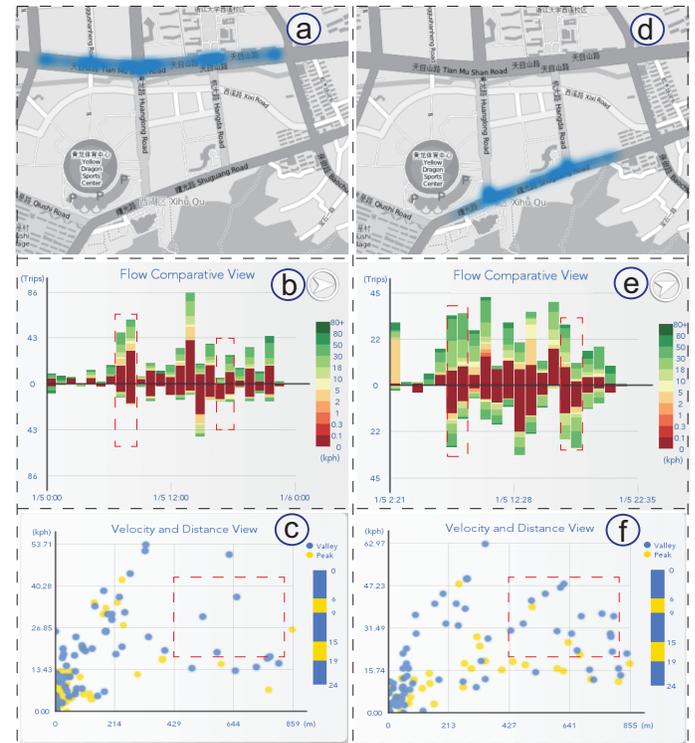


Fig. 10. Traffic flow patterns on two adjacent parallel roads. The left column is for Tianmushan Road, while the right column is for Shuguang Road.

8 CONCLUSION AND FUTURE WORK

We have proposed a visual analysis system designed for interactive transport assessment based on massive taxi trajectory data. To support dynamic querying, we develop a novel road-based query model for users to interactively conduct visual transport assessment tasks on roads. This model is built upon a bi-directional hash structure, namely TripHash, which enables real-time responses to queries over a huge amount of trajectory data. Some well-established visualization techniques are integrated into our system as coordinated views, including Flow Comparative View, Velocity-and-Distance View, Flow-and-Velocity, Topology View, and Flow Density View. We have demonstrated the usefulness of our system with case studies on real data sets.

The system has been tested in a transport assessment company owned by our collaborators. Comments from experts in this company read: 1) This system provides a qualified, graphically-interactive road assessment method; 2) The assessment of whether a road meets its current requirements calls for fewer analysts and devices; 3) Some interesting phenomena that are unseen before can be detected with the assistance of the system.

In future work, we will improve the road based query model aimed to overcome its limitations. (1) We plan to experiment with more flexible widgets for time query specifications, since a flexible time query is very important to transportation analysis. (2) We will design better interaction tools, such as brushing, to provide enhanced user-friendly interface. (3) We will integrate more data sources such as POI data and the public transportation data into the assessment system.

ACKNOWLEDGMENTS

We would like to thank Anthony K. H. Tung at National University of Singapore who introduced R2-D2 to us and inspired us to work on the project. We also express our gratitude to the anonymous reviewers and paper chairs for their constructive and valuable comments. This work was supported by the Major Program of National Natural Science Foundation of China (61232012), the National High Technology Research and Development Program of China (2012AA12090), the Zhejiang Provincial Natural Science Foundation of China (LR13F020001), the Doctoral Fund of Ministry of Education of China (20120101110134), the Fundamental Research Funds for the Central Universities, and the NUS-ZJU SeSama center. Ye Zhao is supported by NSF IIS-1352927. Ronghua Liang is supported by New Century Excellent Talents in University of China under Grant No. NCET-12-1087, Zhejiang Provincial Natural Science Foundation under Grant No. LR14F020002, and Zhejiang Provincial Qianjiang Talents under Grant No. 2013R10054.

REFERENCES

- [1] G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel. *Visual analytics of movement*. Springer Publishing Company, 2013.
- [2] J. v. Bercken, B. Blohsfeld, and e. a. Dittrich. Xxl-a library approach to supporting efficient implementations of advanced database queries. In *Proceedings of the Conference on Very Large Databases (VLDB)*, pages 39–48, 2001.
- [3] A. Biswas, S. Dutta, H.-W. Shen, and J. Woodring. An information-aware framework for exploring multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2683–2692, 2013.
- [4] Caliper. TransCAD Transportation Planning Software. Website, 2014. <http://www.caliper.com/tcovu.htm>.
- [5] V. P. Chakka, A. Everspaugh, and J. M. Patel. Indexing large trajectory data sets with seti. In *CIDR*, 2003.
- [6] Y.-J. Chang, P.-Y. Hung, and M. Newman. Traceviz: brushing for location based services. In *the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, pages 345–348. ACM, 2012.
- [7] D. Chu, D. A. Sheets, Y. Zhao, Y. Wu, J. Yang, M. Zheng, and G. Chen. Visualizing hidden themes of taxi movement with semantic transformation. In *Pacific Visualization Symposium (PacificVis), 2014 IEEE*, pages 137–144. IEEE, 2014.
- [8] Citilab. Cube. Website, 2014. <http://www.citilabs.com/>.
- [9] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
- [10] P. Cudre-Mauroux, E. Wu, and S. Madden. Trajstore: An adaptive storage system for very large trajectory data sets. In *IEEE International Conference on Data Engineering (ICDE)*, pages 109–120. IEEE, 2010.
- [11] N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. *IEEE/ACM Transactions on Networking*, 9(3):280–292, 2001.
- [12] M. Erwig and M. Schneider. Developments in spatio-temporal query languages. In *Tenth International Workshop on Database and Expert Systems Applications*, pages 441–449. IEEE, 1999.
- [13] N. Ferreira and J. e. a. Poco. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158, 2013.
- [14] J. S. Greenfeld. Matching gps observations to locations on a digital map. In *Transportation Research Board 81st Annual Meeting*, 2002.
- [15] W. Guan. Modeling traffic flow characteristics with qualitative and quantitative knowledge. In *Proceedings of IEEE Intelligent Transportation Systems*, volume 1, pages 151–155. IEEE, 2003.
- [16] A. Günemmann, R.-P. Schäfer, K.-U. Thiessenhusen, and P. Wagner. Monitoring traffic and emissions by floating car data. *Institute of Transport Studies Working Paper*, (ITS-WP-04-07), 2004.
- [17] H. Guo, Z. Wang, and B. e. a. Yu. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *IEEE Pacific Visualization Symposium*, pages 163–170. IEEE, 2011.
- [18] A. Guttman. *R-trees: A dynamic index structure for spatial searching*, volume 14. ACM, 1984.
- [19] L. L. S. L. H. Q. L. M. N. H. Liu, Y. Gao. Visual analysis of route diversity. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 171–180. IEEE, 2011.
- [20] M. Hadjieleftheriou, G. Kollios, V. J. Tsotras, and D. Gunopulos. Efficient indexing of spatiotemporal objects. In *Advances in Database TechnologyEDB 2002*, pages 251–268. Springer, 2002.
- [21] M. Hadjieleftheriou, G. Kollios, V. J. Tsotras, and D. Gunopulos. Indexing spatiotemporal archives. *The VLDB Journal*, 15(2):143–164, 2006.
- [22] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.
- [23] INRO. EMME. Website, 2014. <http://www.inro.ca/en/products/emme/index.php>.
- [24] K. Z. K. Deng, K. Xie and X. Zhou. Trajectory indexing and retrieval. In *Computing with Spatial Trajectories*, pages 35–60. Springer, 2011.
- [25] L. Lins, J. T. Klosowski, and C. Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2456–2465, 2013.
- [26] T. Lomax, S. Turner, and G. e. a. Shunk. *Quantifying Congestion. Volume 1: Final Report*. Number Project 7-13 FY’92. 1997.
- [27] Ministry of Construction, PRC and State Bureau of Technical Supervision. *Code for Transport Planning on Urban Road*. China Planning Press, 1995.
- [28] M. A. Nascimento and J. R. Silva. Towards historical r-trees. In *Proceedings of the ACM symposium on Applied Computing*, pages 235–240. ACM, 1998.
- [29] Oculus. GeoTime. Website, 2014. <http://www.geotime.com/>.
- [30] K. Porkaew, I. Lazaridis, and S. Mehrotra. Querying mobile objects in spatio-temporal databases. In *Advances in Spatial and Temporal Databases*, pages 59–78. Springer, 2001.
- [31] J. Pu, S. Liu, Y. Ding, H. Qu, and L. Ni. T-watcher: A new visual analytic system for effective traffic surveillance. In *IEEE 14th International Conference on Mobile Data Management*, volume 1, pages 127–136. IEEE, 2013.
- [32] R. Scheepens, N. Willems, H. van de Wetering, and J. J. van Wijk. Interactive visualization of multivariate trajectory data with density maps. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 147–154. IEEE, 2011.
- [33] D. Schrank, B. Eisele, and T. Lomax. Tti’s 2012 urban mobility report. *Texas A&M Transportation Institute. The Texas A&M University System*, 2012.
- [34] T. Schreiber. Measuring Information Transfer. *Physical review letters*, 85(2):461, 2000.
- [35] L. Shi, Q. Liao, Y. He, R. Li, A. Striegel, and Z. Su. Save: Sensor anomaly visualization engine. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 201–210. IEEE, 2011.
- [36] X. Song, Q. Zhang, and Y. e. a. Sekimoto. Modeling and probabilistic reasoning of population evacuation during large-scale disaster. In *Proceedings of the 19th ACM SIGKDD*, pages 1231–1239. ACM, 2013.
- [37] Y. Tao and D. Papadias. Efficient historical r-trees. In *International Conference on Scientific and Statistical Database Management*, pages 223–232. IEEE, 2001.
- [38] Y. Tao and D. Papadias. The mv3r-tree: A spatio-temporal access method for timestamp and interval queries. 2001.
- [39] Y. Theoderidis, M. Vazirgiannis, and T. Sellis. Spatio-temporal indexing for large multimedia applications. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 441–448. IEEE, 1996.
- [40] N. Willems, H. Van De Wetering, and J. J. Van Wijk. Visualization of vessel movements. In *Computer Graphics Forum*, volume 28, pages 959–

966. Wiley Online Library, 2009.

- [41] X. Xu, J. Han, and W. Lu. Rt-tree: an improved r-tree index structure for spatiotemporal databases. In *Proceedings of the 4th International Symposium on Spatial Data Handling*, volume 2, pages 1040–1049. IGU Commission on GIS, 1990.
- [42] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD*, pages 186–194. ACM, 2012.
- [43] X. Y. J. Z. W. H. v. d. Z. Wang, M. Lu. Visual traffic jam analysis based on trajectory data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2159–2168, 2013.
- [44] Y. Zheng, Y. Liu, J. Yuan, and X. Xie. Urban computing with taxicabs. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 89–98. ACM, 2011.
- [45] J. Zhou, A. K. Tung, W. Wu, and W. S. Ng. R2-d2: a system to support probabilistic path prediction in dynamic environments via semi-lazy learning. *Proceedings of the VLDB Endowment*, 6(12):1366–1369, 2013.
- [46] J. Zhou, A. K. Tung, W. Wu, and W. S. Ng. A semi-lazy approach to probabilistic path prediction. In *Proceedings of the 19th ACM SIGKDD*, pages 748–756. ACM, 2013.
- [47] P. Zhou, D. Zhang, B. Salzberg, G. Cooperman, and G. Kollios. Close pair queries in moving object databases. In *ACM International Workshop on Geographic Information Systems*, pages 2–11. ACM, 2005.
- [48] H. Zhu, J. Su, and O. H. Ibarra. Trajectory queries and octagons in moving object databases. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 413–421. ACM, 2002.