Urban Traffic Coulomb's Law: A New Approach for Taxi Route Recommendation

Yongxuan Lai, Zheng Lv, Kuan-Ching Li, and Minghong Liao

***Abstract*— Recently, an increased amount of effort has been focused on optimizing the selection of routes for taxis, as part of the development of smart urban environments, and the increase of the accumulated trajectory data sets. One challenging issue is to match and recommend appropriate cruising routes to taxis, as most taxis cruise on streets aimlessly looking for passengers. Drivers encounter lots of difficulty in optimizing their cruise routes and hence increasing their incomes, and such inability not only decreases their profit but also increases the traffic load in urban cities. In this paper, the concept of *urban traffic Coulomb's law* is coined to model the relationship between taxis and passengers in urban cities, based on which a route recommendation scheme is proposed. Taxis and passengers are viewed as positive and negative charges. It first collects useful information such as the density of passengers and taxis from trajectories, then calculates the traffic forces for cruising taxis, based on which taxis are routed to optimal road segments to pick up desired passengers. Different from existing route recommendation methods, the relationship among taxis and passengers are fully taken into account in the proposed algorithm, e.g., the attractiveness between taxis and passengers, and the competition among taxis. Moreover, real-time dynamics and geodesic distances in road networks are also considered to make more accurate and effective route recommendations. Extensive experiments are conducted on the road network using the trajectories generated by approximately 5,000 taxis to verify the effectiveness, and evaluations demonstrate that the proposed method outperforms existing methods and can increase the drivers' income more than 8%.**

***Index Terms*— Traffic force, urban traffic Coulomb's law, cruising route recommendation, taxi trajectories.**

I. Introduction

TAXIS play important role in the daily life of urban citizens. Different from other public transportations such as buses or subways, taxis do not follow fixed routes periodically. Drivers have to plan their own routes right after passengers are dropped off [1], and their incomes are largely determined by the selection of these routes. Nevertheless, it is not easy for a driver to schedule and select the best route in order to maximize his/her earnings. Fig. 1 depicts the scatter graph between the drivers' daily income and their activeness, which is represented by the number of recorded daily active GPS points. From the figure, we could see that the daily income positively correlates with the drivers' activeness, However, drivers with roughly the same activeness have large difference on the incomes. There are 343 drivers whose recorded number of daily active GPS points are within range (250,260] (indicated by the red rectangle). Their average and maximal income are 1064.5 and 1502.9 RMB respectively, and the standard variance is 149.23. This is mainly due to the route selection. If drivers select a route that has no/fewer passengers, they would waste their time and cruise in vain, which also burdens the traffic load in urban cities.

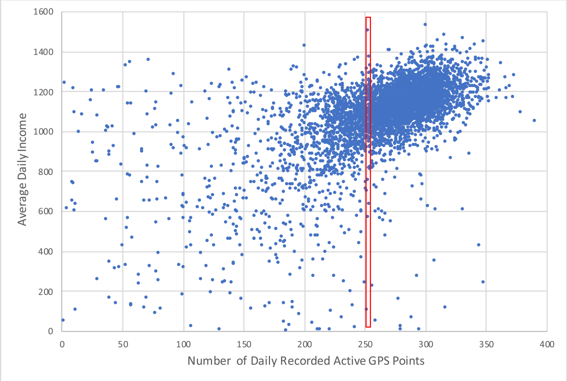


Fig. 1. Scatter graph between drivers' daily incomes and their activeness which is represented by the number of daily recorded active GPS points. There are 343 drivers whose recorded number of daily active GPS points are within range (250, 260] (indicated by the red rectangle). Their average and maximal income are 1064.5 and 1502.9 RMB respectively, and the standard variance is 149.23.

Recent years even some demand responsive taxi service platforms, such as Didi Taxi and Uber, emerge to guide taxis to pick up passengers, the drivers still need to plan their own cruising routes when there are no demands from the platform. Thus, it is a key issue to help taxi drivers to select better and efficient cruising routes to increase their incomes. This would mutually benefit both drivers and passengers, as well as to improve the efficiency of urban transportation systems.

One possible solution is through the learning of massive trajectory and operation datasets. With the advancement of smart devices and networking technologies, location-based services (LBS) is widely used in our daily lives [2]–[4].

Manuscript received May 22, 2017; revised December 26, 2017 and April 7, 2018; accepted September 13, 2018. This work was supported in part by the Natural Science Foundation of China under Grant 61672441 and Grant 61872154, in part by the Shenzhen Basic Research Program under Grant JCYJ20170818141325209, in part by the National Key Technology Support Program under Grant 2015BAH16F01, and in part by the State Scholarship Fund of China Scholarship Council under Grant 201706315020. The Associate Editor for this paper was A. Marconi. *(Corresponding author: Yongxuan Lai.)*

Y. Lai is with the Software School, Xiamen University, Xiamen 361005, China, and also with the Shenzhen Research Institute, Xiamen University, Shenzhen 518057, China (e-mail: laiyx@xmu.edu.cn).

Z. Lv and M. Liao are with the Software School, Xiamen University, Xiamen 361005, China (e-mail: xmdxlz@stu.xmu.edu.cn; liao@xmu.edu.cn).

K.-C. Li is with the Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan (e-mail: kuancli@pu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2018.2870990

Most taxis are equipped with GPS localizer, and geographic positions are reported to the operating company periodically, e.g. 2-3 times every minute [5]. Other information of each taxi, such as the occupancy, is also recorded. Having these datasets, it is possible to create new recommendation strategies to help drivers select their cruising routes and pick up more passengers [1], [6], [7]. In the passenger-pickup scenario, taxis seek for passengers and passengers concurrently wait for taxis, drivers have to compete with each other to pick up more passengers. With this respect, there are similarities between the traffic network and the electrostatic field in physics. Stated in Coulomb's law [8], there consists positive and negative charges in electrostatic field, as like charges repel and unlike charges attract. Similarly, the concept of *Urban Traffic Coulomb's law* is developed, where urban traffic network is viewed as a big electrostatic field and taxis and passengers as the positive and negative charges. A taxi 'repels' another taxi as they are competing to find passengers, and a passenger 'attracts' a taxi as they are looking for each other.

As we can note, there are two aspects in the *Urban Traffic Coulomb's law*: the former is the attractiveness between taxis and passengers, and the latter is the competition among taxis. Most existing research matched to the first aspect, road segments with more passengers and hotspots are recommended as cruising routes [1], [6], [7], [9]–[12]. Yet they failed to consider the second aspect, i.e., the conflict and competition among vacant taxis, which leads to ineffectiveness of the route recommendation strategy. In this study, we propose a taxi cruising route recommendation algorithm based on *Urban Traffic Coulomb's law*, which takes into account both aspects of the Urban Traffic Coulomb's law for the algorithm design. The proposed algorithm collects useful information from historical trajectories, and calculates the traffic attraction and traffic forces for cruising taxis, based on which optimal road segments are recommended to drivers to pickup desired passengers. Extensive simulations conducted based on a real trajectory dataset verified the effectiveness of the proposed method. Results show that the proposed scheme can improve taxis' income by more than 8% when compared to the ground truth and other methods.

The major contributions of this paper are as follows:

- The concept of *Urban Traffic Coulomb's law* is proposed to model the relationship between taxis and passengers in urban cities. Taxis and passengers are viewed as positive and negative charges. Formulas that calculate the *traffic charges* and *traffic forces* are also defined within this concept;
- We proposed a cruising route recommendation method for taxi drivers based on *Urban Traffic Coulomb's Law*. It first collects useful information such as the density of passengers and taxis from trajectories, then calculates the attraction and traffic force for cruising taxis, based on which taxis are routed to optimal road segments to pick up desired passengers. The proposed algorithm considers both the attractiveness between taxis and passengers, and the competitions among both taxis and passengers;
- We optimised the *Urban Traffic Coulomb's Law* on route recommendation by considering the realtime dynamics and geodesic distances in road networks. The attraction force on a taxi is calculated based on distance on road networks, and it combines forces from the extended regions and objects within its current region. Also, the short-sighted route recommendation problem on road network is avoided by redefining the recommended road segment.
- We conduct extensive experiments on large number of real-world historical GPS trajectories and operation dataset to verify the effectiveness of the proposed method. Experimental results show that the proposed scheme can effectively increase the income of taxi drivers compared to other methods.

The remaining of this paper is organized as follows. Section II presents the related works; section III and IV give some preliminaries and introduce the concept of the *Urban Traffic Coulomb's Law*. Next, section V presents the detailed description of the proposed algorithm, followed by experiments studies and analysis delivered in Section VI. Finally, section VII concludes the paper and presents future directions.

II. Related Work

A. Modeling of Vehicle Trajectories

A continuous trajectory is a function which gives the location of a moving object as a continuous function of time [13]. Vehicle trajectories can be utilized in many ways, including urban planning, mobility pattern analysis, traffic condition prediction and recommendation system. Reference [14] presented a trajectory clustering method to discover spatial and temporal travel patterns in a traffic network, where the Longest Common Subsequence (LCS) between two vehicle trajectories are used as the similarity measure, and a density-based clustering algorithm is extended to incorporate the LCS-based distance for the clustering. Reference [13] introduced the symmetrized segment-path distance to measure the spatial-temporal similarities of trajectories. Reference [15] proposed the use of a trajectory flow graph, a dynamic graph of aggregated flows constructed from individual trajectories, to better understand and analyse city-wide mobility patterns. In [6], a location-to-location graph model was adopted to capture the relation between the passenger get-off location and the next passenger get-on location. Similarly, in [1] taxi drivers' picking-up/dropping-off behaviors from the GPS trajectories of taxicabs are learned, and then the knowledge are fed into a probabilistic model which estimates the profit of the candidate locations for a particular driver based on where and when the driver requests the recommendation. In this way, the historical and real-time request data are fused to enhance the recommendation.

B. Route Recommender Systems

There are a number of research on the improvement of cruising and routing planning for the taxi drivers or vehicles in urban scenarios. Some focus on the angle of drivers, e.g. behaviours of drivers and taxies [16], and more others are based on mining the patterns on real-world datasets [17]–[20]. In this research the recommendation system based on GPS trajectories is roughly classified into two categories: the macroscopic and the microscopic recommender systems.

In macroscopic recommender systems, only the driving directions are provided to taxi drivers, rather than the complete driving routes. Generally, passengers pick-up locations are extracted from GPS trajectories, and these locations are clustered into multiple representative small areas, which are the recommended driving directions for taxi drivers. For example, hotspots were extracted from large amount of pick-up points using clustering algorithm [21], so the hotspots are recommended to taxis. Reference [17] took raw data and combined users' location histories and created tree-based hierarchical graph. It applies hypertext induced topic search to infer interesting locations, based on which the travel recommendations are conducted. Reference [22] proposed an improved ARIMA method to forecast the spatial-temporal variation of passengers in a hotspot to help taxi drivers find new passengers. Reference [11] extracted vehicular mobility pattern from the large-scale GPS trace datasets based on clustering, and adopted a strategy for taxis to select pick-up points using a Markov Decision Process (MDP) model. Reference [6] adopted a location-to-location graph model, referred to as OFF-ON model to capture the relation between the passenger get-off location and the next passenger get-on location. It then estimates the expected fare for a trip started at a recommended location based on the waiting time, distance and some other factors. Reference [23] proposed a Spatio-Temporal Profitability (STP) map to guide taxicabs to cruise around most profitable locations. The process assembles the scores into an STP map that suggests potentially profitable locations to the taxicab driver. By following the suggestions, the driver can reduce cruising time and thus increase hits/her income.

Compared with macroscopic recommender systems, the microscopic systems provide taxi drivers with actual driving routes. A recommendation system for both taxi drivers and passengers is proposed in [24]. For taxi drivers, the system recommends a good parking place with shorter waiting time and longer distance for the next trip. However, it is not evaluated by how much they can improve the revenue. Reference [10] proposed a heuristic algorithm called MSCR that scores each road segment and thus obtains a cruising route with the highest score. In [25], a system called pCruise is proposed to reduce the taxi's cruising mile by recommending the shortest cruising route with at least one expected available passenger. Reference [26] proposed a Gaussian Process Dynamic Congestion Model for non-myopic adaptive routing to minimize the collective travel time of entire fleets of vehicles. The method characterizes both the dynamics and the uncertainty of congestion conditions.

The proposed approach in this paper belongs to the microscopic one, which is more flexible and reliable to recommending the next driving road segment. Most of existing research on macroscopic recommender systems mainly focuses on the pick-up and drop-off points or the live trips from loaded taxis. It neglects the competition among vacant taxis, so is tended to recommend places with potential passengers, such as airport and train station, regardless of the impact of other taxis. The concept of Urban Traffic Coulombs Law naturally takes the competition among taxies or passengers into consideration. We take the density of vacant taxis and passengers into consideration, and reasonably recommend a direction with larger "attractive force" rather than with more passengers.

TABLE I

Xiamen Taxi Dataset. (a) Position Trajectory Dataset (T). (b) Operation Dataset (O)

(a)

id	lon	lat	time
8250864460	118.024232	24.475228	2014-07-02 15:33:03
speed	**direction**	**occupied**	
61 (km/h)	270 (^{o})	"vacant" / "occupied"	

(b)

id	sLon	sLat	onTime
8259447816	118.089600	24.486733	2014-07-01 00:00:11
fee	**eLon**	**eLat**	**outTime**
9.6 (RMB)	118.096625	24.492792	2014-07-01 00:02:25

Also the existing research focuses solely on historical trajectories or data set, which makes the recommendation not adapting to the dynamics of road network and trip demand. Instead, the proposed approach combines the historical trajectories and real-time dynamics together for the recommendation. The timeless of trajectories would improve the precision of recommendation results.

III. Preliminaries

A. Road Network and Route Recommendation

Road network can be represented and built in many ways [27]–[29]. In this research, the road network is characterized by a set of intersections and a set of road segments (e.g. extracted from OpenStreetMap [30]). A road network is represented by a graph $G = (I, S)$, where $I = I_1, I_2, \ldots, I_n$ is a finite set of n intersections and $S = S_1, S_2, \ldots, S_m$ is a finite set of m road segments. A road segment is determined by two intersections, $S_i = (I_j, I_k)$, and it contains five properties: the identification $S_i.id$, the direction $S_i.dir$, the length $S_i.length$, the starting intersection $S_i.s$, and the ending intersection $S_i.e$. Similarly, a road intersection I_i is associated with four properties, the identification $I_i.id$, the longitude $I_i.lon$, the latitude $I_i.lat$, and the number of segments that connect to it $I_i.num$. A route W is a directed sequence of L road segments, i.e. $W = S_1, S_2, \ldots, S_L$, where $W.s = S_1.s$, $W.e = S_L.e$ and $S_j.e = S_{j+1}.s$ for $1 \leq j < L$ which means that consecutive road sections contained in a route should share an intersection.

Real world taxi datasets are used in this research. The position trajectories T records the position of taxis and the operation trajectories O records the trips of taxis. Table I(a) and Table I(b) list the format of GPS data with an example. id is the identification of the trajectory or the transaction, lon, lat denotes the longitude and latitude of the position, $time$ denotes the timestamp when the position is recorded. $sLon, sLat, onTime$ stands for the longitude, latitude and time when a passenger gets on a taxi. Similarly, $eLon, eLat, outTime$ stands for the longitude, latitude and time when a passenger gets off a taxi. Column fee stands for the money of a transaction or trip, base on which the ground truth of drivers' earnings are deduced.

In fact, a taxi driver only makes decision when he/she arrives at an intersection. Selecting different road segments leads to different incomes among taxi drivers. Based on the definitions and notations above, we formally define the problem of *optimal taxi route recommendation*:

Definition 1: The optimal taxi route recommendation problem.

Given: A road network $G = (I, S)$, a trajectory T, a operation dataset O, and a set of taxis X.

Objective: Find the optimal route W for all taxis in X to increase earnings. Specifically, providing the best road segment to a taxi drivers as soon as he/she reaches an intersection and thus finding the optimal route W.

It should be noted that the optimal route from a fixed intersection depends on the time context since the distribution of passengers and taxis and traffic condition change throughout the time. Intuitively, there are more passengers waiting for taxis in the morning at some places, e.g. a residential area, and more passengers in the evening at other places, e.g. an entertainment area. Nevertheless, there is a great difference on the total amount of passengers in different time slots, e.g., the number of passengers waiting for taxis in the daytime is much larger than that of those during the midnight. Therefore, a taxi route recommendation should be established on the fully take advantage of the historical regularity and real-time change of traffic. In this study we use the historical trajectories as well as the real-time trajectories to make route recommendations for taxis. Additional detailed discussions are presented in Section V.

IV. Urban Traffic Coulomb's Law

In this section, preliminaries of the proposed approach are introduced, including *Coulomb's law* and the modified version applied to urban traffic, called *Urban Traffic Coulomb's law*, or *UTCL*.

A. Coulomb's Law

Coulomb's law is one of the most important laws in electricity to quantify the force between two electrostatic charges. It states that the magnitude of the electrostatic force between two point charges is directly proportional to the scalar multiplication of the magnitudes of charges and inversely proportional to the square of the distance between them; the direction of the force is along the straight line that joins them. In short, *Coulomb's law* could be simply formulated as Eq. (1):

$$\overrightarrow{F}(q_1, q_2) = k \times \frac{q_1 \times q_2}{r^2} \times \overrightarrow{e}_{q_1,q_2} \tag{1}$$

where $k = 8.99 \times 10^9$ is the Coulomb's constant, q_1 and q_2 are the signed magnitudes of the charges, the scalar r is the distance between the charges, $\overrightarrow{e}_{q_1,q_2}$ is the radius vector between q_1 and q_2. $\overrightarrow{F}$ is the electric force between q_1 and q_2.

B. Urban Traffic Coulomb's Law

Urban traffic network can be considered as a special category of "electric field" by isomorphing passengers and taxis as charges with different signs. For a vacant taxi driver, his next potential passenger may exist in any region. In other words, he/she is pulled by the "electric forces" produced by the electric charges in all regions around him. It is clear that the idea of *Coulomb's Law* could be utilized to calculate the "electric force" from the surrounding regions to the taxi driver, denominated as *Urban Traffic Coulomb's Law* (UTCL), being defined as follows:

Definition 2 (Traffic Charge): A traffic charge c of region r is composed of taxis and passengers in r, corresponding to the sum of all the "electric charges" in r in "Coulomb's Law".

Traffic charge describes the ability on attracting taxis of a region, which is positive in most cases. Fig. 2 depicts the distributions of traffic charges in Xiamen City, China based on Xiamen taxi dataset. Here we give an abstract concept as there may have different formal definitions at different circumstances. The formal definition on traffic charge used in this paper is given in section V-B.

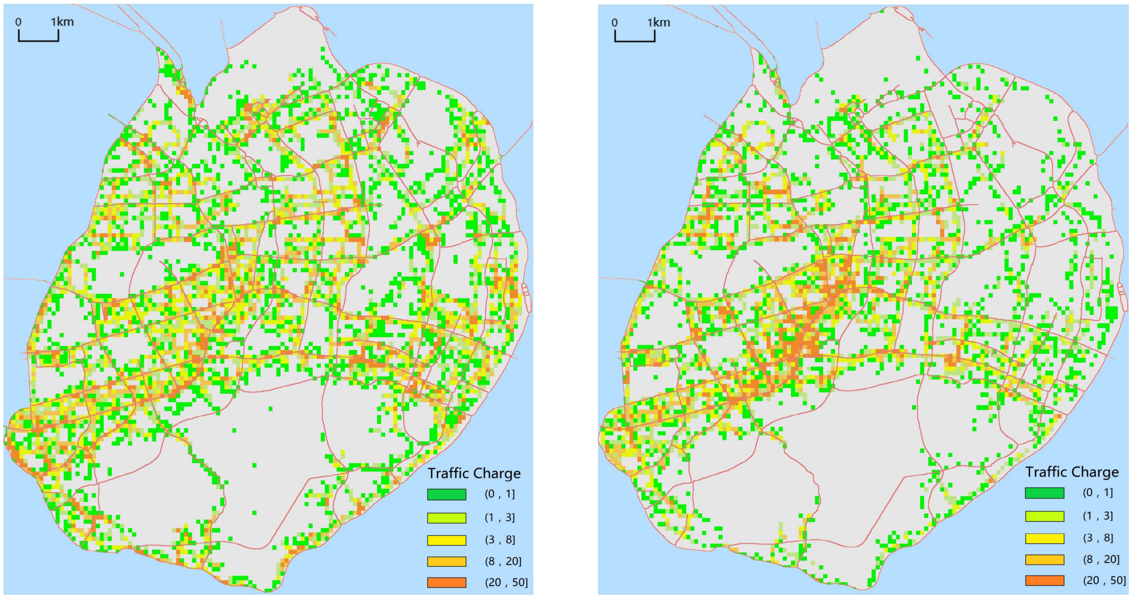
Definition 3 (Traffic Force): Traffic force describes the direction and magnitude of the attractiveness on taxis and passengers, corresponding to the "electric force" in "Coulomb's Law". For a taxi in region r_1, the magnitude of the traffic force from region r_2 is directly proportional to the magnitudes of the traffic charge in r_2 and inversely proportional to the square of the distance between them. The direction is along the straight line joining them pointing to r_1. Specifically, it can be stated as a mathematical expression:

$$\overrightarrow{F}(r_1, r_2, t) = k' \times \frac{C_{t,r_2}}{R^K_{r_1,r_2}} \times \overrightarrow{e}_{r_1,r_2} \tag{2}$$

where t is the current time slot, R_{r_1,r_2} is the distance between r_1 and r_2, $K \geq 0$ is a constant, and the force is inverse proportional to $R^K_{r_1,r_2}$. k' is a constant to normalise the result. In this research $k' = 1$, as the absolute magnitude of k' does not affect the overall calculation of route recommendations. C_{t,r_2} is the traffic charge of r_2 in t, R_{r_1,r_2} is the distance between the r_1 and r_2, $\overrightarrow{e}_{r_1,r_2}$ stands for the direction from r_1 to r_2. Note that either Euclidean distance or road distance could be used to calculate R_{r_1,r_2}. Details of distance calculation are discussed at section VI-C.

V. Routes Recommendation

In this section, we present a formal description of the proposed recommendation approach *UTCL*. Fig. 3 depicts the framework of the proposed approach. First, the processing of both spatial and temporal domains for metadata collection are performed. Next, the traffic charges are calculated based on the historical trajectories and the real-time trajectories. The first two steps belong to the *off-line mining*. During the process of *online route recommendation*, the traffic force will be calculated using *UTCL* as soon as an empty taxi reaches an intersection. Finally, a recommended road segment connected with this intersection will be provided. This process is repeated until the taxi picks up a passenger.



(a) (b)

Fig. 2. Distributions of traffic charges in Xiamen City, China based on Xiamen taxi dataset. The traffic charges at 3:00 are larger and the distribution is more centralised on some places of interests compared with that at 18:00. The traffic charges are calculated based on historical and recent factors including potential passengers, rival taxis, traffic conditions and revenue per trip. (a) At 18:00. (b) At 3:00.

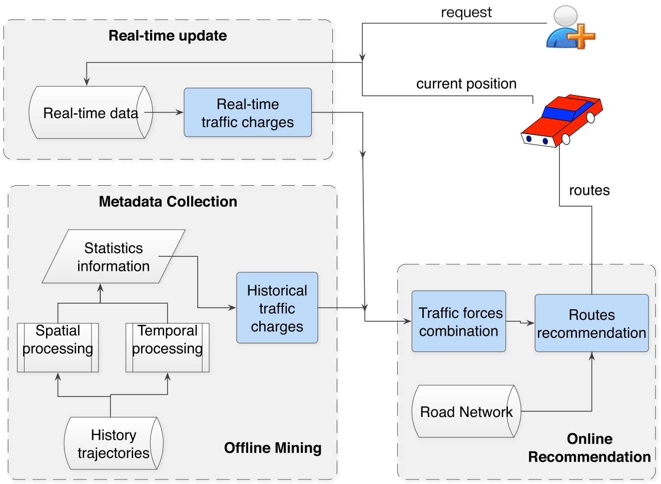


Fig. 3. Overview of the routes recommendation system.

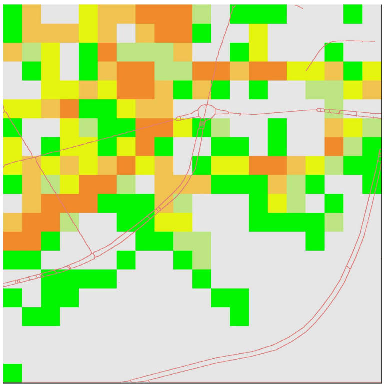


Fig. 4. Distribution of traffic charges at 18:00 at the area of Xiamen Railway Station. Each grid is seen as a region, and has a width of 0.001 degree of longitude and a length of 0.001 degree of latitude.

A. Spatio-Temporal Processing and Metadata Collection

Metadata collection means to collect and extract useful statistics information from trajectories, such as the density of passengers and taxis in a given area. Considering the influence of spatio-temporal context, two types of work need to be done before metadata collection, i.e., spatial processing and temporal processing.

Spatial processing is the first step, which divides a map into a few smaller regions with comparable areas, and then a grid-based method processing is conducted. For example, the map is divided by 0.001 degree of longitude and 0.001 degree of latitude in the experiment, where each unit is seen as a region. Grid granularity is very important in spatial processing, since fine granularity may provide enough meaningful and distinctive historical information, while coarse granularity may be short of historical trajectories or unable to capture the local characteristic of a region. Additional discussions on the influence on grid granularity will be presented in section VI-F1.

Temporal processing is another important step before metadata collection. As discussed in section III, the distribution of passengers and taxis, and traffic condition of each region changes over time. Therefore, metadata collection should base on a pre-defined time division. The influence of the division of time slots to recommendations will be presented in section VI-F2.

Once spatial processing and temporal processing are completed, statistic information on each region and each time slot can be collected from trajectories. The definition of several primary metadata are listed below:

1) The passenger density in region r at time slot t:

$$P_{t,r} = |\Theta_1| \tag{3}$$

where $\Theta_1 = \{l | l \in O \wedge l.onTime \in t, [l.slon, l.slat] \in r\}$. Here, $|\Theta_1|$ means the magnitude of set Θ_1.

2) The average passenger density in all non-zero regions at time slot t:

$$P_t = \frac{\sum_{r \in G} P_{t,r}}{|G|} \tag{4}$$

where G is the set of grid regions in the road network, and $|G|$ stands for the total amount of regions divided during spatial processing.

3) The average fee of one trip in region r at time slot t:

$$M_{t,r} = \frac{\sum_{p \in \Theta_1} p.fee}{P_{t,r}} \tag{5}$$

where p is a trip in Θ_1.

4) The density of vacant taxis in region r at time slot t:

$$V_{t,r} = |\Theta_2| \tag{6}$$

where $\Theta_2 = \{l | l \in T \wedge l.time \in t \wedge [l.lon, l.lat] \in r \wedge l.occupied = \text{“}vacant\text{”}\}$.

5) The density of all taxis in region r at time slot t:

$$A_{t,r} = |\Theta_3| \tag{7}$$

where $\Theta_3 = \{l | l \in T \wedge l.time \in t \wedge [l.lon, l.lat] \in r\}$.

6) The average speed of all taxis in region r at time slot t:

$$S_{t,r} = \frac{\sum_{k \in \Theta_3} k.speed}{A_{t,r}} \tag{8}$$

B. Traffic Charge Storage

The definition of *traffic charge* is given in this section, using the notations defined at formula (3) – (8).

Without loss of generality, passengers are defined as positive charges $\oplus$, vacant taxis as negative traffic charges $\ominus$, and the occupied taxi or on- board passenger as neutral charges $\bigcirc$. And the sign of traffic charge in a region r at time slot t, $C_{t,r}$, is defined as follows:

$$sign(C_{t,r}) = \begin{cases} \oplus, & P_{t,r} > V_{t,r} \\ \ominus, & P_{t,r} < V_{t,r} \\ \bigcirc, & P_{t,r} = V_{t,r} \end{cases} \tag{9}$$

where $P_{t,r}, V_{t,r}$ are the densities of passengers and vacant taxies in region r at time slot t. Here we assume one taxi only serve one passenger at a time, as one positive traffic and one negative charge nullify each other. The magnitudes of traffic charge $C_{t,r}$ is defined as follows:

$$C_{t,r} = \frac{P_{t,r}}{P_t} \times (2 - \frac{V_{t,r}}{A_{t,r}}) \times (1 + \frac{S_{t,r}}{S_{max}}) \times (1 + \frac{M_{t,r}}{M_{max}}) \tag{10}$$

The traffic charge is composed of four parts. The first part is $\frac{P_{t,r}}{P_t}$, which reflects the ratio of passenger density in region r at t compared to the average level at t. In the second part $\frac{V_{tr}}{A_{tr}}$ is about the taxis, reflecting a positive influence on passengers, as well as a negative influence on vacant taxis. $(2 - \frac{V_{t,r}}{A_{t,r}})$ indicates that the larger the ratio of vacant taxis in a region, the less appeal to a vacant taxi outside this region. It is clear that the density of passengers and taxis is of vital importance in the calculation on the magnitude of traffic charge in a region. For the third part, a better traffic condition $(\frac{S_{t,r}}{S_{max}})$ means less time consumption on a trip and more live trips in a fixed time which may generate more income. Thus, a better traffic condition can promote the traffic charge of a region. Last, a region with higher revenue per trip $(\frac{M_{t,r}}{M_{max}})$ is preferred. For the last three parts, we use ‘2 minus’ and other measures to maintain the result between 1 and 2 thereby preventing the undue influence on these three aspects. From Eq. (10), we see a region with higher traffic charge considered to have more potential passengers, less rival taxis, better traffic condition and higher revenue per trip.

Here, the traffic charges extracted from the historical trajectories are called the *historical traffic charges*, represented as C_h, containing the historical experience of urban traffic. Similarly, the traffic charges extracted from the recent trajectories are called the *recent traffic charges*, represented as C_r, containing the real-time changes of urban traffic. As depicted in Figure 3, we calculate the historical traffic charges and recent traffic charges respectively and then combine them into the final traffic charges. The definition of final traffic charges, represented as C_f, is shown as:

$$C_f = w \times C_h(ts) + (1 - w) \times C_r(f) \tag{11}$$

where $w \in [0, 1]$ stands for weight of historical traffic charge, C_h stands for the historical traffic charge, ts stands for the corresponding time slot division, $1 - w$ stands for weight of recent traffic charge, C_r stands for the recent traffic charge and f stands for the corresponding update frequency. When w is equal to 1, the final traffic charge is equal to the historical traffic charge; when w is equal to zero, the final traffic charge is equal to the recent traffic charge. For example, if current time is 8:15 AM, the current time slot is (8:00, 8:30] (if slot time is set to 0.5 hour). The historical traffic charge would be calculated on the (8:00, 8:30] time slot over all data, and the recent traffic charge would be calculated based on the past two time slots, i.e. (7:00, 7:30], (7:31, 8:00]. The optimal weight could be tuned to design an efficient recommender method.

C. Integrating Historical and Real-Time Factors

The above steps belong to the off-line mining and the final traffic charges are stored into the database to calculate the traffic forces which gives real-time route recommendations. In this section we discuss the integration of historical and realtime dynamic factors.

As discussed in section IV-A, a charge would be affected by all charges in the electrostatic field. Similarly, a taxi is affected by *traffic forces* from all regions in the urban traffic network. Although we define the traffic charges by grids/regions, this also leads to large amount of computation if all regions are considered to calculate the traffic force. In reality, some regions far away from a taxi have little impact on this taxi. Therefore, it makes little sense to add up all regions when calculating the *attraction* to the taxi. Therefore we define a extended region around the taxi's current region, e.g. 4 square kilometer, while the regions outside of this extended region would not be considered [23]. The size of the extended region should be large enough to contain enough regions that have significant effect. We denote the *extended region* of a taxi as M, so each region in M would have a force of push or pull the taxi so as to lead it to its optimal routes. Fig. 6(a) depicts the forces between a taxi and regions in M.

As defined in Eq. (10), the traffic charges represent historical and recent factors such as potential passengers, rival taxis, traffic conditions and revenue per trip. To integrate the realtime factors for the recommendation, the current set of vacant taxis and passengers are also considered for the calculation of traffic forces. Suppose taxi i is currently located in region r, denote the set of vacant taxis in r by X_r, and the set of passenger by Y_r. Then every taxi $x \in X_r$ would have a push force to i and every passenger $y \in Y_r$ would have a pull force to i. The recommendation server would maintain the sets of vacant taxis and passengers, which change dynamically overtime. Fig. 6(b) depicts the forces between a taxi and elements in X_r and Y_r.

D. Combination of Traffic Forces

The concept of *traffic force* from one region to another is depicted in Definition 2. Here we use the concept of *Attraction* to represent the vectorial sum of all *traffic forces* that affect a taxi. The definition of *Attraction* is given as follows.

Definition 4 (Attraction): Attraction is the aggregated result of traffic forces produced from other regions and elements within its current region to a taxi, including both directions and forces. Attraction on taxi i in region r, which is with an extended region M, during time slot t could be derived by the formula as below:

$$\overrightarrow{A}(r,t) = \sum\nolimits_{r' \in M} \frac{C_{t,r'}}{{R_{r',r}}^K} \times \overrightarrow{e}_{r',r} \tag{12}$$

$$+ \sum\nolimits_{x \in X} \frac{C_{t,r}}{V_{t,r} \times {R_{i,x}}^K} \times \overrightarrow{e}_{i,x} \tag{13}$$

$$+ \sum\nolimits_{y \in Y} \frac{C_{t,r}}{P_{t,r} \times {R_{i,y}}^K} \times \overrightarrow{e}_{i,y} \tag{14}$$

where r' stands for a region in the extended region M, $V_{t,r}$, $P_{t,r}$ are the densities of vacant taxis and passengers in r at time slot t, and other variables are the same as those in Eq. (2). The attraction is combined from three parts: (12) is the force from regions belonging to its extended region, (13) is the force from vacant taxis, and (14) is the force from passengers.

Fig. 6 illustrates the combination of traffic force corresponding to taxi B in Fig. 5. Taxi B is current located at region r. Region $r1, r3, r5, r7, r8$ are in the extended region M, they would have traffic forces on taxi B. Specially, there is a "push" force from $r3$ and a "pull" force from $r5$, because from the historical and recent analysis of the traffic charges, there are vacant taxis in $r3$ and passengers in $r5$. And within region r at current time, there are two other vacant taxis (A, D) and a passenger (D), all of which would have a force on B. So finally, the combined force, or called the attraction, is illustrated in Fig. 6(c).

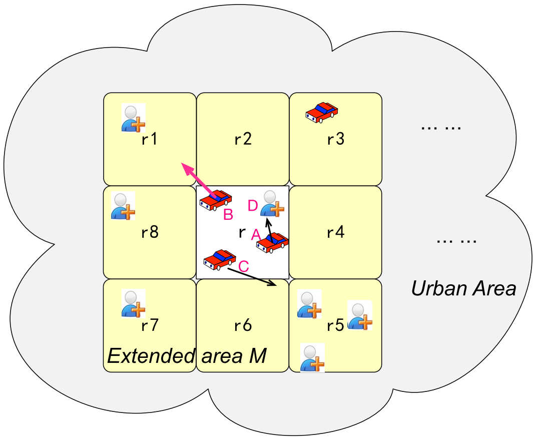


Fig. 5. Illustration of a taxi in its current region r, and its extended region M. Arrows are the combined traffic force.

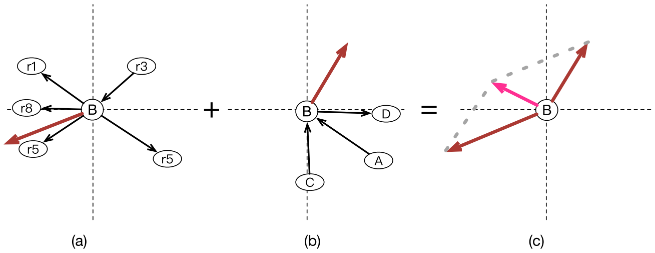


Fig. 6. Illustration of combining traffic forces for a taxi B. The arrows in red colour in (a) and (b) are the combined forces, and the arrow in pink denotes the final combined force. (a) Traffic force in M. (b) Traffic force in r. (c) Combined force.

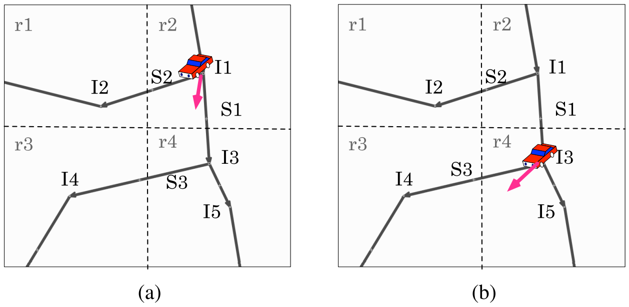


Fig. 7. An illustration of route recommendation. I1, I2, I3, I4 and I5 stand for five intersections. S1, S2, S3 stand for three road segments. r1, r2, r3, r4 stand for four regions. (a) At time t_1. (b) At time t_2.

We will show how to recommend routes using the calculated *attraction*s in the next section according to Eq. (12).

E. Route Recommendation

As a vacant taxi cruises on the street, a taxi driver can make decisions about cruising routes anytime, but can only take actions at intersection on streets [25]. Therefore, for a recommendation system, the recommended result should be given at the same time, exactly the way the proposed method works. Additional details about the recommendation algorithm are presented using examples shown in Fig. 7.

Imagine a vacant taxi driver arrives at I_1 at time slot t_1. At this moment, he can select to drive along S_1 or S_2.

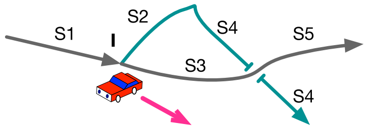


Fig. 8. Illustration of “short-sighted” route recommendation. The combined force, which is denoted by the red arrow, has a minimal angle with $S3$, yet tpath $S2 \rightarrow S4$ is a better route because it directs to the direction of the traffic force.

The proposed algorithm will judge these two choices and recommend him the most suited one. First, the proposed algorithm will acquire the traffic charge from the database and further calculate the *attraction*. After comparing the direction of the *attraction* with the direction of $\overrightarrow{I_1 I_2}$ and $\overrightarrow{I_1 I_3}$, the proposed algorithm will choose the road segment nearest to the direction as the recommended road segment. Hereby, the force of *attraction*, which stands for the attraction degree, would not be used in the proposed method. Suppose S_1 is the next recommended road segment and the driver arrives at I_3 at time slot t_2. The same procedure presented above will be performed again but with different time slot and region. As shown in figure 7, S_3 is the recommended road segment this time. The above steps will be under processing until the driver picks up the next passenger and a new recommendation process will start as soon as the passenger leaves the taxi. The recommended road segment is given in Eq. (15):

$$S_{rec} = \underset{S_j}{\text{argmin}} \, \{\varphi(S_j, \overrightarrow{A}(r,t)) : S_j \in S \wedge S.si = I\} \quad (15)$$

where I stands for the intersection the taxi reaches, S_j is the road segment started with I, $\varphi(S_j, \overrightarrow{A}(r,t))$ stands for the angle between the road segment S_j and the traffic force $\overrightarrow{A}(r,t)$.

The recommendation is based on the direction of the traffic force. However, a road segment that has the similar direction might not necessarily leads to the directed area suggested by the traffic force in real world road networks. As illustrated in Fig. 8, the combined force, which is denoted by the red arrow, has a minimal angle with road segment $S3$. But path $S2 \rightarrow S4$ would be a better route because it directs to the direction of the traffic force. To avoid such a “short-sighted” route recommendation, we add another parameter k to calculate the optimal road segment. Suppose the taxis is at intersection I, and S_j is a road segment that starts at I, then we define the set of k-segment routes as $W(S_j, k)$, where a route $Z \in W(S_j, k)$ is a directed sequence of k road segments, i.e. $Z = S_1, S_2, \ldots, S_k$, where $Z.s = S_1.s, Z.e = S_k.e$ and $S_i.e = S_{i+1}.s$ for $1 \le i < k$. The angle is rewritten as $\varphi(S_j, \overrightarrow{A}(r,t), k)$:

$$\varphi(S_j, \overrightarrow{A}(r,t), k) = min(\frac{1}{k} \sum_{S \in \{Z/Z.S_1\}} \varphi(S, \overrightarrow{A}(r,t))), \quad Z \in W(S_j, k) \quad (16)$$

Here we calculate the average angles between the traffic force and all possible k-step routes. $\{Z/Z.S_1\}$ means we remove the first road segment for the average calculation.

F. Algorithm and Complexity Analysis

The recommendation procedure is personalized, since drivers are located at different locations, as well the attraction forces are different for each other. From a macro perspective, this mechanism can prevent sending the same information to multiple drivers, which may result in localized competition and a non-equilibrium state. The online recommendation procedure is depicted in Algorithm 1.

Algorithm 1 Online Recommendation Using UTCL

```
Input:
    $lon_c$, current longitude;  $lat_c$, current latitude;
    $T_c$, current time;   $M$: the extended region;
    $C$: storage of traffic charges;
    $w$: weight of historical traffic charge;
    $occupied$: taxi's current status;
Output:
    $route$: recommended route;
1: $route \leftarrow \emptyset$;
2: while taxi is working do
3:   update $T_c, lon_c, lat_c$;
4:   while ($occupied == false$) do
5:     if (a passenger is picked up by the taxi) then
6:       $occupied \leftarrow true$;
7:     else
8:       $region \leftarrow getRegion(lon_c, lat_c)$;
9:       $intersection \leftarrow getIntersection(lon_c, lat_c)$;
10:      $objects \leftarrow getObjects(T_c, region)$;
11:      $attraction \leftarrow getForce(T_c, region, C, M, objects, w)$;
12:      $rs \leftarrow getRecommend(attraction, intersection)$;
13:      $route \leftarrow route + \{rs\}$;
14:      send $route$ to the driver;
15:    end if
16:    $route \leftarrow \emptyset$;
17:  end while
18: end while
```

The procedure works when a taxi is working, and the time, location are periodically updated (line 3). When a taxi drops off his most recent passenger, it cruises on the street looking for the next passenger. If a passenger is picked up, the taxi is occupied and no route recommendation is needed (line 5-6); else, the algorithm will first acquire the current region, intersection, and objects within the currrent region (lines 8-10). Then it calculates the final traffic force based on the historical traffic charges, recent charges, and real-time traffic dynamics as defined at Fig. 6 (line 11). Next, attraction will be calculated given the position and time according to Eq. (12) (line 12). Finally, the recommended road segment will be obtained using the method proposed in section V-E and provided to the driver (lines 13-14). This process loops until the taxi picks up the next passenger.

The traffic charges are calculated in the preprocessing phase, the cost of storage for the charge is $n{*}k$, where n is the number of grids in the simulation field and k is the number of time slots within a day. The computation on *attraction* and routes recommendation are real-time. The cost depends on line 11, where forces on the taxi from extended region and objects within the region are combined. There are m forces on the taxi, where $m = |M| + |r|$ is the number of regions in the extended region M plus the number of objects in region r. The algorithm iteratively combine two forces into one, and forms a geometric sequence with scale factor 0.5, so the computational complexity of the force combination process is $O(2m)$.

VI. Experimental Analysis

A. Environment Setup

The proposed method is evaluated using the taxi trajectory data of Xiamen (China) during July 2014, containing about 5,000 taxis and the reporting frequency is 1-3 times per minute. At the mean time, an associated taxi operation dataset is also used. Totally, there are over 220 million GPS position records and 8 million taxi live trips. The dataset is spit up by two, from July 1^{st} to July 21^{st} being the training set and the remaining is considered as test set. The region is limited to $[118.066E,118.197E] \times [24.424N,24.561N]$ partitioned into 131×137 grids with equal intervals. The extended region M is set to 0.02 degree of longitude and 0.02 degree of latitude. During data cleaning processing, the GPS records outside region R and those containing contradictory in time are removed from the dataset. To evaluate the proposed method, we use the maps available provided by OpenStreetMap to build a road network. Totally, 52479 road segments and 49773 intersections are stored in PostgreSQL. Besides, time slot and the update frequency of traffic charges are both set to 1 hour. w_h is set to 0.8 and 0.2 for w_r.

B. Map Matching

After building a road network, it is of vital importance to locate GPS trajectories into corresponding roads, which is also called map matching. The purpose of map matching is to integrate the positioning data with the spatial road network data, to identify the actual way on which the vehicle is traveling and further to determine the vehicle location on that way. For each GPS record, the map matching is processed in three steps: (1) Identifying possible road segments, (2) Identifying candidate road segments, and (3) Weighing candidate road segments.

1) Identifying Possible Road Segments: For each GPS record, it is inefficient to match all possible road segments to search suitable road segments to the record. Rather, we only need to identify a few road segments that covers all possible segments for the GPS record whilst filter others. According to [31] and [32], GPS location errors can be as large as 100 meters in a city with dense tall buildings and viaducts. Actually, 100 meters can be roughly regarded as 0.001 latitude or longitude. Imagine there is a circle of radius 0.001 latitude or longitude centered at the GPS record, the GPS record can only reside on the road segments that intersect or tangent to the circle. From our investigation, 99.27% road segments in our road network are less than 0.005 latitude or longitude long. Such a circumstance is described in Figure 9.

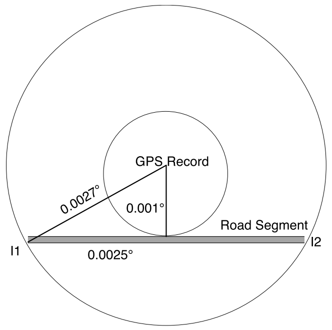


Fig. 9. The maximum distance appears when the road segment is tangent to the circle and the tangent point turns to be the midpoint.

Therefore, we test each road segment on the following criteria: whether there exists a road segment that meets the condition that the distance between the endpoint and the GPS record is less than 0.0027 latitude or longitude long. The GPS record would not reside on a road segment that fails to meet this criteria.

2) Identifying Candidate Road Segments: After obtaining all possible road segments of a vehicle's GPS record, we need to identify the candidate road segments satisfying some basic conditions.

First, we need to check whether the distance between the road segment and the GPS record is less than the distance threshold, which is set to 0.001 latitude or longitude as told before. To calculate the distance between the road segment and the GPS record, we need to calculate the closest point on the road segment, which can be achieved by projecting the location of GPS record onto the road segment. If the projection point lies on the road segment, the distance between the road segment and the GPS record turns to be the distance between the projection point and the GPS record. Otherwise, if the projection point lies outside the segment, we should calculate the distances from the GPS record to the road segment's two terminal points, and choose the shorter one.

Second, we also need to make sure the difference between the headings of the road segments and the driving direction is less than 60°, a heading difference threshold used in Pfoser's work [33]. The direction of road segment can be calculated using its two terminal points (first and second points). There are two directions about one road segment, and the smallest value as the difference between road segment and vehicle heading is selected. If the difference is less than the threshold, the road segment will be identified as a candidate road segment; the road segment is not considered as a candidate road segment even if the GPS record-Road segment distance is very short otherwise.

3) Weighting Candidate Nodes: After identifying a set of candidate road segments for each GPS record, each road segment is given a weight based on the following two factors: (1) proximity between the locations of GPS record and the road segment, (2) similarity between the vehicle heading and the direction of road segment. Given one GPS record g and its k candidate road segments $S = s_1, s_1, \ldots, s_k$, the score of one candidate road segment s_i can be computed as:

$$score(g, s_i) = \theta_1 \frac{dis\,(g, s_i)}{0.001} + \theta_2 \frac{\alpha\,(g, s_i)}{60^\circ} \tag{17}$$

where θ_1 and θ_2 are weights of proximity about location and similarity about direction respectively. dis () is the function of the distance between one GPS record and one road segment, and α () is the angle difference function between them. After the score is calculated for each candidate road segment from the GPS record, the road segment that has the minimum score is considered as that matched the GPS record on the map.

C. Road Distance Between Regions

As in (2), the traffic force is defined based on the distance between any two regions. We use the road distance to have better estimation, i.e. R_{r_1,r_2} is the road distance between two regions. There are many points within a region, yet we define R_{r_1,r_2} as the distance between the *central intersections* in r_1 and r_2. Given a region r, its central intersection is denoted by CI_r:

$$\begin{aligned} CI_r &= \underset{I}{\operatorname{argmin}}\{f(I) : I \in L(r)\}, \\ f(I) &= \frac{c}{|I.num|} + \frac{|I.lat - C_r.lat|}{r.l} + \frac{|I.lon - C_r.lon|}{r.w} \end{aligned} \tag{18}$$

where $L(r)$ denotes the set of intersections in r, c is a constant integer balancing the value ($c = 8$), $I.num$ is the number of road segments that connect to I, $C.r$ is the geometric central point of the rectangle defined by r, and $r.w, r.l$ are the width and length of r. So an intersection that connects with more road segments and is closer to the central point C_r is preferred to become a central intersection of that region. Then the distance between region $r1$ and $r2$ is defined as:

$$R_{r_1,r_2} = dist(CI_{r_1}, CI_{r_2}) \tag{19}$$

where function $dist$ calculates the shortest road distance between CI_{r_1}, CI_{r_2} in the road network. The calculation is only performed once and the result could be stored as the metadata.

D. Simulation

To verify the effectiveness of the proposed method, we selected 50 most active taxis from 8 a.m. to 12 p.m in the test set. We simulated the moving paths of virtual taxi drivers from the starting locations of these 50 drivers.

Algorithm 2 presents the pseudocode of the simulation. Given a taxi and its current position, the positioning information is obtained first (lines 3-4). The positioning process is simplified here, while the specific steps can be found in section VI-B. If a passenger appears at the current road intersection at that time, the revenue of this trip will be obtained and added up as income (lines 6). Next, the travel time of this trip would also be added and the taxi will move to the location where the passenger is dropped off (line 7-8). If there is no passenger in the road intersection, the taxi will move along the recommended route. First, the final traffic charges are calculated using the historical traffic charges and the real-time objects in the current region (line 10), the attraction force is calculated (line 11) and the recommended road segment is received (line 12). The taxi will move along the recommended road segment and move to the next intersection (line 13), and the travel time will be calculated and added (line 14). This simulation ends when the time reaches the final simulation time (line 2).

Algorithm 2 Simulation Algorithm

```
Input:
    $lon_c$: current longitude;   $lat_c$: current latitude;
    $T_c$: current time;   $T_f$: final simulation time;
    $C$: storage of traffic charges;
    $w$: weight of historical traffic charge;
    $M$: extended region of current location of taxi;
Output:
    $income$: the income of a taxi;
 1: $income \leftarrow 0$;
 2: while $T_c < T_f$ do
 3:   $region \leftarrow getRegion(lon_c, lat_c)$;
 4:   $intersection \leftarrow getIntersection(lon_c, lat_c)$;
 5:   if $(findPassenger(T_c, intersection))$ then
 6:     $income \leftarrow income + getTripMoney()$;
 7:     $T_c \leftarrow T_c + getTripTime()$;
 8:     $lon_c, lat_c = getOffLocation()$;
 9:   else
10:     $objects \leftarrow getObjects(T_c, region)$;
11:     $attraction \leftarrow getForce(T_c, region, C, M, objects, w)$;
12:     $rs \leftarrow getRecommend(attraction, intersection)$;
13:     $lon_c, lat_c \leftarrow moveAlong(rs, intersection)$;
14:     $T_c \leftarrow T_c + getTravelTime(rs, T_c)$;
15:   end if
16: end while
```

E. Evaluation

Beside the ground truth and UTCL, we also conducted some recommendation methods for the comparison and analysis:

1) HITS [17]: the modified "HITS" model that recommends taxi drivers to regions that are with the highest profit based on historical dataset.
2) STP [23]: adopts a spatio-temporal profitability map to guide taxicabs to cruise around most profitable locations. The map suggests potentially profitable locations to the taxicab driver, and taxies follow the suggestions of map to reduce cruising time and thus increase the profit.
3) Random: randomly selects a road segment when meeting an intersection, which represents the case when no direction or route guidance are available for drivers.

Fig. 10 shows the probability of earnings between the ground truth and proposed scheme. Taxis that follow the recommended routes by UTCL have higher probability to make higher revenue, about 81 percent of the taxis earns more than 50 RMB per hour; while only 66 percent of taxis earn more than 50 RMB per hour with the ground truth. No taxis

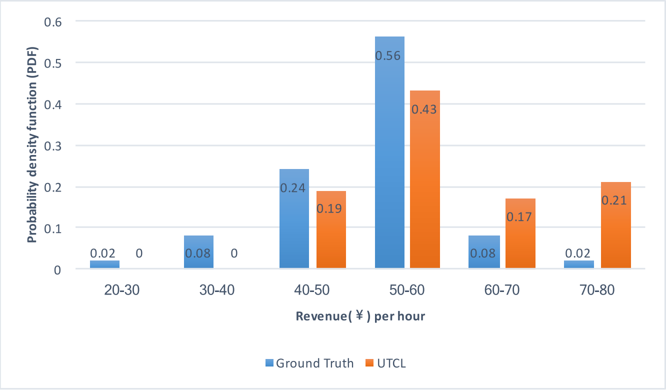


Fig. 10. Comparison of taxis' earning probability between UTCL and the ground truth.

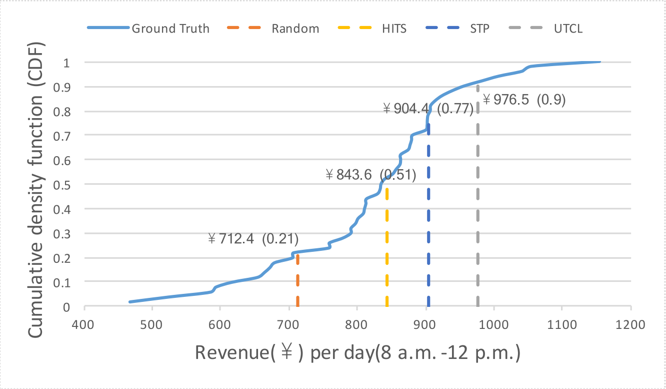


Fig. 11. Comparison of average earning of taxis on UTCL, STP, HITS and randomised method.

in UTCL earn less than 40 RMB per hour, yet in the ground truth, about 10 percent of drivers earn less than 40 RMB per hour. In general taxis following the routes recommended by the proposed method have a better earning probability, which demonstrate the effectiveness of the proposed method.

Fig. 11 shows the cumulative density probability of driver earnings of the ground truth, and the average revenues of the schemes are also indicated on the line. The average average of the randomized method is 712.4, smaller than that with the ground truth (i.e. 822.7), indicating that the drivers' experience is effective on supporting the selection of cruising routes. The effectiveness of using a passenger-finding strategy is shown when compared with the results without a passenger-finding strategy. The average revenue earned by the taxi with UTCL is about 976.5, which is among the top 10% of the taxis, while those in HITS and STP are 843.6 and 904.4 respectively. UTCL has the best performance among the three recommendation methods. The revenue of taxis has been increased by more than 8% when using UTCL, which indicate higher competitiveness of the proposed method.

F. Impact Factor Analysis

1) Partition Granularity: Partition granularities determine the spatial accuracy on recommendation. Fig. 12 shows the impact of different partition granularities, where "PG" stands for partition granularity. "PG=0.001" means each region is partitioned into 0.001 degree of longitude and 0.001 degree of latitude, so does "PG=0.005" and "PG=0.01". Simulation results showed that a smaller granularity like PG=0.001 are more likely to reach a high revenue (i.e. 1000-1300), while larger granularity tends to get a low revenue (i.e. 0-800). About 45 percent of taxis earn less than 800 when PG=0.01, and about 30 percent for PG=0.05, while only 20 percent of taxis earn within this range. This is mainly because larger granularity may lead to fewer regions and rougher calculations, thus resulting in ambiguous recommendation. In simulation experiments performed, PG=0.01 signifies to divide the map of Xiamen into 14 × 14 grids with an area of nearly 1 square kilometer, which makes it hard to make a targeted recommendation in such a big area. Nevertheless, partition granularity should not be too small, as we have presented in section V-A. Too small granularity may lead to few trajectory data on each grid and thus affect the recommendation, it also leads to needless massive consumption on calculations.

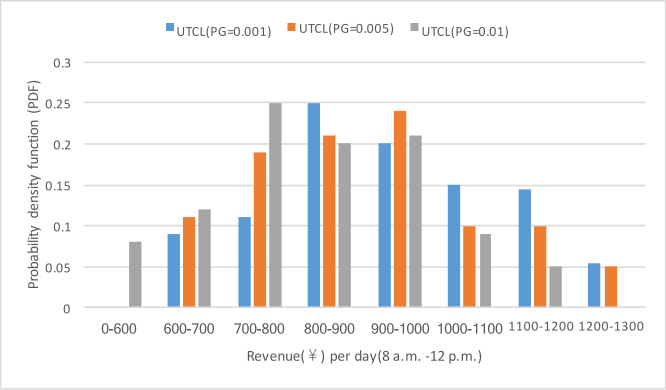


Fig. 12. Simulation results on different partition granularity. "PG" stands for partition granularity. "PG=0.001" means each region is partitioned into 0.001 degree of longitude and 0.001 degree of latitude, so does "PG=0.005" and "PG=0.01".

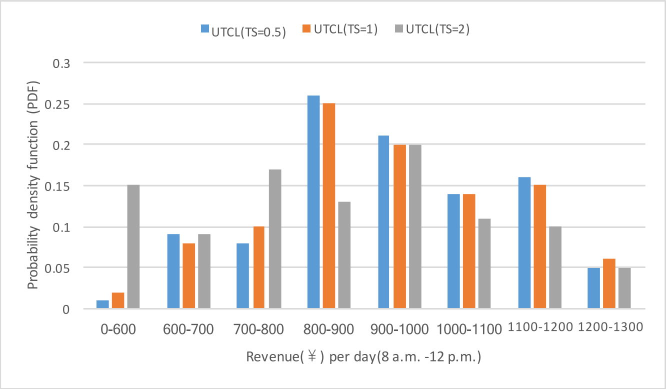


Fig. 13. Simulation results on different time slots. "TS" stands for time slot and "TS=0.5" means time slot is set for 0.5 hour. So does the "TS=1" and "TS=2".

2) Time Slots: Time slot determines the temporal accuracy of recommendation. The differences on time slots are shown in Fig. 13. We see that the ratio of high revenue taxi drivers (i.e. 1000-1300) for TS=2 is about the same as that for

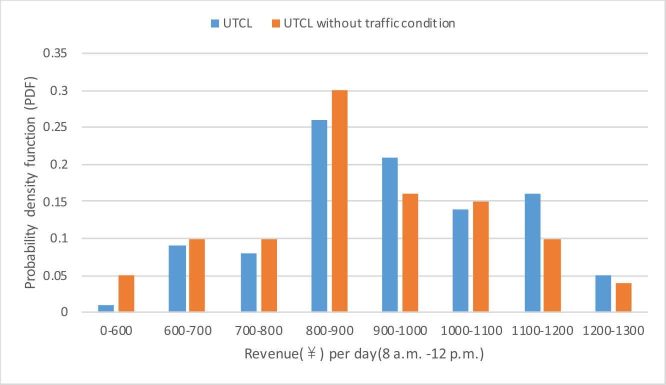


Fig. 14. Simulation results on using or without using traffic condition.

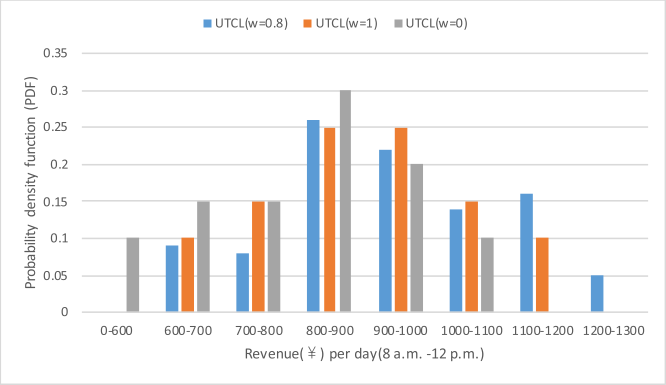


Fig. 15. Comparison on different ratio between historical charges and recent charges. $w = 0.8$ means the final traffic charges are composed of 80% historical traffic charges and 20% recent traffic charges. $w = 1$ means to use the historical charges alone; $w = 0$ means to use the recent charges alone.

TS=0.5 and TS=1, while the ratio of high revenue taxi drivers for TS=2 is lower. About 24 percent of taxis earn less than 700 when the time slot is 2 hours, while only 2 percent of taxis earn less than 600. This is reasonable since smaller time slot captures more accurate traffic change and thus make a better recommendation. When the time slot is too large, e.g. larger than 2 hours, the distribution of traffic conditions change, and may change from peak period to flat period. This makes the recommendation outdated, which harms the overall performance of the recommendation. A time slot of 0.5 hour is a good estimation on the traffic conditions and charges, which is a good time division that improves the recommendation in UTCL.

3) Traffic Condition: As defined in (10), the speed of road network is considered to calculate the traffic charge. Yet to study the impact of the traffic condition, we also conduct the recommendation when the factor of speed is taken away. Fig. 14 shows the differences with or without the factor of traffic condition. Compared to UTCL without traffic condition, the distribution from UTCL is more concentrated in higher revenue. As we can observe, about 35 percent of taxi drivers make more than 1000 from the simulation with traffic condition, whilst only 20 percent of drivers can make that much for simulation without traffic condition. The factor of speed leads to more accurate estimation of the charge calculation and hence the recommended routes. Regions that have large speed means less time consumption on one trip. Thus, given a fixed time slot (e.g. a day in the simulation), a taxi has more chance to pick up more passengers through saving time consumption while heading to regions with good traffic condition. Simulation results demonstrate that it is necessary to take traffic condition into consideration for the calculation of traffic charges.

4) Ratio Between Historical and Recent Charges: The final traffic charges are composed of the historical traffic charges and the recent traffic charges. $w \in [0, 1]$ stands for the weight of historical traffic charge, $1 - w$ stands for the weight of recent traffic charge, which is calculated based on the past two time slots. Fig. 15 shows the impact of the ratio w. In this experiment, given the condition that 0.5 hour for the time slot, the best outcome is achieved when w is set to 0.8. The probability of UTCL when $w = 0.8$ on the high revenue recommendation section, e.g. greater than 1000, is higher than those when only using historical trajectories ($w = 1$) or recent trajectories ($w = 0$) alone. About 35 percent of taxi drivers make more than 1000, while only 16 and 10 percent of taxi drivers make more than that amount of revenue. This indicates the effectiveness of integrating the recent traffic charges for the route recommendation. Recent charges stands for the most temporal closed factors that affects the current recommendations on routes. An appropriate ratio between them can reflect the distribution of traffic more accurately, thus make a better recommendation.

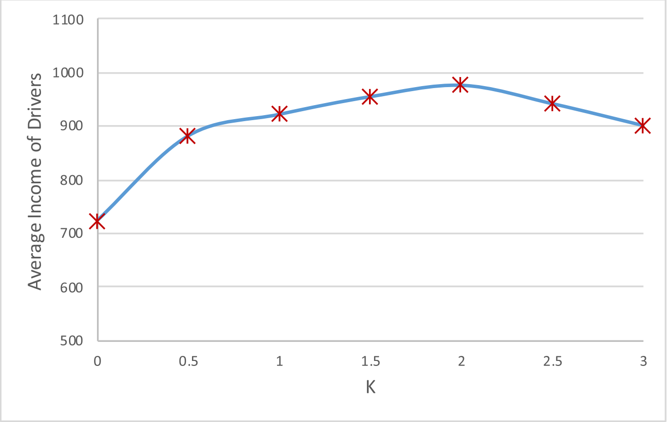


Fig. 16. Impact of K on the average income of drivers in UTCL.

5) Impact of K: In Coulomb's law the force is inverse proportional to the squared distance. Yet in Urban Traffic Coulomb's law, we adopt a more general form. As defined in Eq. (2), the traffic force is inverse proportional to K powering the distance, i.e. R_{r_1,r_2}^K, where R_{r_1,r_2} is the distance between region r_1 and r_2. In the experiment we varied K to see its impact on the performance in Fig. 16. When K is small, e.g. 0, the traffic force has no relation with the distance, so passengers, whether they are near or far away, would have the similar impact on taxis. When K is larger, the traffic force would be dominated by the distance. Both of these cases do not accord with the relations between taxis and passengers. UTCL gains the best performance of the average income of drivers when K is [1.5, 2]. This accords with the Coulomb's law in physics, where K is set 2.

6) Number of Steps Looking Forward: To avoid the "short-sighted" route recommendation problem, UTCL looks k road segments forward to calculate the optimal road segment as described in Eq. (16). In the experiment, we varied k from 1 to 5, and found that $k \geq 3$ works well in the real-world road network. The ratio when UTCL directs taxis to the right road segments is 96.21% when K is 1, and the ratio grows to 1.0 when K is set 3.

VII. Conclusion and Future Works

In this paper we coin the concept of *Urban Traffic Coulomb's Law* to model the relationship between taxis and passengers in urban cities, based on which a new framework of recommending cruising routes is proposed. Taxis and passengers are viewed as different types of charges. Traffic charges and *attraction*s are calculated for each region at different time slots according to *Urban Traffic Coulomb's Law*, then the cruising routes for drivers are computed by comparing the difference between attraction force and the headings of adjacent road segments. Different from other route recommendation methods, the relationship among taxis and passengers are fully taken into account in the proposed algorithm, e.g. the attractiveness between taxis and passengers and the competition among taxis. Besides, the recent trajectories and real-time traffic dynamics are also taken into account for recommendation, while most existing methods are focused on the historical trajectories alone. Experimental results show that the proposed method can effectively provide taxi drivers with better routes. Compared to other methods, drivers with this proposed method have better performances, where drivers' income increases more than 8 percent.

There are some directions as future work for this research. Cruising is not the only way to find passengers, and waiting at temporary places, e.g., taxi stops, may be an efficient option compared to cruising around, which is a situation that may be taken into account. Another direction is, the waiting time and preferences of passengers are also important factors to be included in the recommendation algorithm. Further studies based on the *Urban Traffic Coulomb's Law* is promising, we are optimistic that it would increase the drivers' income as well as improve the overall traffic efficiency and benefit the living conditions in urban cities.

References

[1] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: A recommender system for finding passengers and vacant taxis," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2390–2403, Oct. 2013.

[2] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving K-anonymity in privacy-aware location-based services," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 754–762.

[3] C.-Y. Chow and M. F. Mokbel, "Trajectory privacy in location-based services and data publication," *ACM SIGKDD Explor. Newslett.*, vol. 13, no. 1, pp. 19–29, 2011.

[4] J. Shao, R. Lu, and X. Lin, "FINE: A fine-grained privacy-preserving location-based service framework for mobile devices," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 244–252.

[5] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 316–324.

[6] Y.-L. Hsueh, R.-H. Hwang, and Y.-T. Chen, "An effective taxi recommender system based on a spatiotemporal factor analysis model," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 28–40.

[7] M. Zhang, J. Liu, Y. Liu, Z. Hu, and L. Yi, "Recommending pick-up points for taxi-drivers based on spatio-temporal clustering," in *Proc. Int. Conf. Cloud Green Comput.*, Nov. 2012, pp. 67–72.

[8] D. Hetherington, "Coulomb's law," *Phys. Educ.*, vol. 32, no. 4, p. 277, 1997. [Online]. Available: http://stacks.iop.org/0031-9120/32/i=4/a=025

[9] B. Li *et al.*, "Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2011, pp. 63–68.

[10] H. Dong, X. Zhang, Y. Dong, C. Chen, and F. Rao, "Recommend a profitable cruising route for taxi drivers," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2014, pp. 2003–2008.

[11] S. Qian, Y. Zhu, and M. Li, "Smart recommendation by mining large-scale GPS traces," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 3267–3272.

[12] T. Takayama, K. Matsumoto, A. Kumagai, N. Sato, and Y. Murata, "Waiting/cruising location recommendation based on mining on occupied taxi data," in *Proc. WSEAS Int. Conf. Math. Comput. Methods Sci. Eng.*, 2010, pp. 225–229.

[13] P. C. Besse, B. Guillouet, J.-M. Loubes, and F. Royer, "Review and perspective for distance-based clustering of vehicle trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3306–3317, Nov. 2016.

[14] J. Kim and H. S. Mahmassani, "Spatial and temporal characterization of travel patterns in a traffic network using vehicle trajectories," *Transp. Res. C, Emerg. Technol.*, vol. 59, pp. 375–390, Oct. 2015.

[15] J. Kim, K. Zheng, S. Ahn, M. Papamanolis, and P. Chao, "Graph-based analysis of city-wide traffic dynamics using time-evolving graphs of trajectory data," in *Proc. 38th Australas. Transp. Res. Forum (ATRF)*, 2016, pp. 1–13.

[16] S. Liu, S. Wang, C. Liu, and R. Krishnan, "Understanding taxi drivers' routing choices from spatial and social traces," *Frontiers Comput. Sci.*, vol. 9, no. 2, pp. 200–209, 2015.

[17] Y. Zheng, L. Zhang, X. Xie, and W. Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. Int. Conf. World Wide Web (WWW)*, Madrid, Spain, Apr. 2009, pp. 791–800.

[18] C. Etienne and O. Latifa, "Model-based count series clustering for bike sharing system usage mining: A case study with the Vélib' system of Paris," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, p. 39, 2014.

[19] C. Liu, K. Qin, and C. Kang, "Exploring time-dependent traffic congestion patterns from taxi trajectory data," in *Proc. IEEE Int. Conf. Spatial Data Mining Geograph. Knowl. Services*, Jul. 2015, pp. 39–44.

[20] T. V. Le, S. Liu, and H. C. Lau, "Reinforcement learning framework for modeling spatial sequential decisions under uncertainty: (Extended abstract)," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2016, pp. 1449–1450.

[21] H.-W. Chang, Y.-C. Tai, and Y.-J. Hsu, "Context-aware taxi demand hotspots prediction," *Int. J. Bus. Intell. Data Mining*, vol. 5, no. 1, pp. 3–18, 2010.

[22] X. Li *et al.*, "Prediction of urban human mobility using large-scale taxi traces and its applications," *Frontiers Comput. Sci.*, vol. 6, no. 1, pp. 111–121, 2012.

[23] J. W. Powell, Y. Huang, F. Bastani, and M. Ji, "Towards reducing taxicab cruising time using spatio-temporal profitability maps," in *Advances in Spatial and Temporal Databases*. Berlin, Germany: Springer-Verlag, 2011, pp. 242–260.

[24] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger," in *Proc. Int. Conf. Ubiquitous Comput.*, 2011, pp. 109–118.

[25] D. Zhang and T. He, "*pCruise*: Reducing cruising miles for taxicab networks," in *Proc. IEEE 33rd Real-Time Syst. Symp. (RTSS)*, Dec. 2012, pp. 85–94.

[26] S. Liu, Y. Yue, and R. Krishnan, "Non-myopic adaptive route planning in uncertain congestion environments," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2438–2451, Sep. 2015.

[27] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *Proc. Int. Conf. Very Large Data Bases*, Trondheim, Norway, Aug./Sep. 2005, pp. 853–864.

[28] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 15, no. 5, pp. 312–328, 2007.

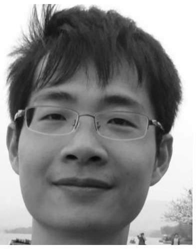
[29] M. A. Quddus, *High Integrity Map Matching Algorithms for Advanced Transport Telematics Applications*. London, U.K.: Imperial College London, 2007.

[30] *OpenStreeMap*. Accessed: Oct. 30, 2016. [Online]. Available: http://www.openstreetmap.org/

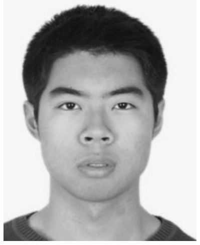
[31] H. Zhu, Y. Zhu, M. Li, and L. M. Ni, "SEER: Metropolitan-scale traffic perception based on lossy sensory data," in *Proc. IEEE Int. Conf. Comput. Commun., Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Rio de Janeiro, Brazil, Apr. 2009, pp. 217–225.

[32] M. A. Quddus, R. B. Noland, and W. Y. Ochieng, "Validation of map matching algorithms using high precision positioning with GPS," *J. Navigat.*, vol. 58, pp. 257–271, May 2005.

[33] D. Pfoser and C. S. Jensen, "Capturing the uncertainty of moving-object representations," in *Proc. Int. Symp. Spatial Databases*, vol. 1651, 2002, pp. 111–131.



Yongxuan Lai received the Ph.D. degree in computer science from the Renmin University of China in 2009. He was an Academic Visiting Scholar with the Data and Knowledge Engineering Group, University of Queensland, Australia, from 2017 to 2018. He is currently an Associate Professor with the Software School, Xiamen University, China. His research interests include network data management, vehicular ad hoc networks, and big data management and analysis.



Zheng Lv is currently pursuing the master's degree with the Department of Software Engineering, Xiamen University, China. His research interests include vehicular networks, recommendation systems, and data mining using trajectories.

Kuan-Ching Li received the Ph.D. and M.S. degrees in electrical engineering and the Licenciate degree in mathematics from the University of São Paulo, Brazil, in 2001, 1996, and 1994, respectively. He is currently a Professor with the Department of Computer Science and Information Engineering, Providence University, Taiwan. His research interests include cluster and grid computing, vehicular networks, parallel software design, and life sciences computing.

Minghong Liao received the Ph.D. degree in computer science from the Harbin Institute of Technology, China, in 1993. He is currently a Professor with the Software School, Xiamen University. His research interests include intelligent networks and pervasive computing.