2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS): SmartCity16: The 2nd IEEE INFOCOM Workshop on Smart Cities and Urban Computing

Navigating the Last Mile with Crowdsourced Driving Information

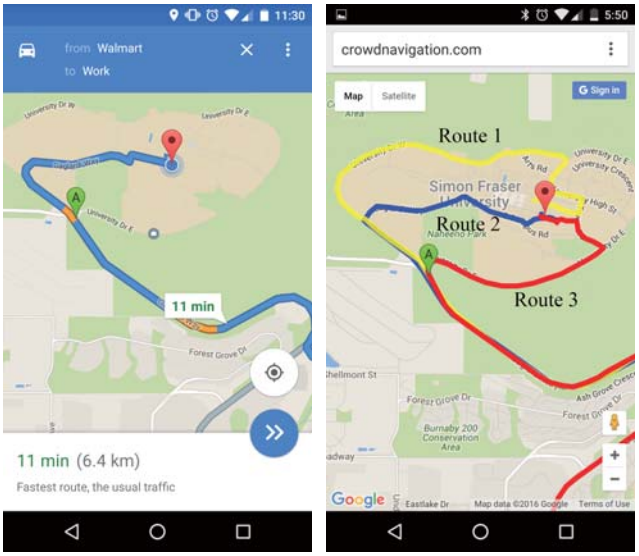
Xiaoyi Fan*, Jiangchuan Liu*, Zhi Wang†, Yong Jiang†,
*School of Computing Science, Simon Fraser University, Canada
†Graduate School at Shenzhen, Tsinghua University, China
Email: {xiaoyif, jcliu}@cs.sfu.ca, {wangzhi, jiangy}@sz.tsinghua.edu.cn

***Abstract*—With digital maps of the transport network and realtime traffic updates, today's navigation services provide good quality routes in the major route level. Once entering the *last mile* near the destination, they unfortunately can be ineffective and, instead, local drivers often have a better understanding of the routes there. Given the deep penetration of 3G/4G mobile networks, drivers are now well connected anytime and anywhere; they are readily to access information from the Internet and share information to the community. These motivate our design of *CrowdNavi*, a complementary service to existing navigation systems, seeking to combat the last mile puzzle. CrowdNavi collects the crowdsourced driving information to identify the local driving patterns, and recommend the best local routes to reach the destinations. In this paper, we present the architectural design of CrowdNavi and the algorithms for different modules, including identifying the last segment from the drivers trajectories, scoring the landmark and locating best routes with user preferences. We have implemented the CrowdNavi app on Android OS, and have examined its performance under various circumstances. The experimental results demonstrate its superiority in navigating drivers in the last segment.**

I. Introduction

Today, such online digital map services as Google Maps are accessed by billions of users on a daily basis[1]. Together with the advances of outdoor positioning services (GPS in particular), the navigation for drivers or pedestrians has increasingly been an essential application on smartphones or car consoles, which seeks to recommend the shortest or fastest routes using the digital maps and GPS. Such real-time driving information as live traffic or construction locations have been incorporated as well. On the macroscale of a transportation network, the quality of the recommended routes are generally acceptable with state-of-the-art navigation services. Yet it is known that [1] the routes from the map-based services often fail to be agreed by local drivers, who have detailed knowledge and experience with local driving conditions.

To illustrate this, consider navigation from a supermarket to an office building in our campus. We first search the Google Maps, which provides two candidate routes as shown in Fig. 1(a). We then set up an experiment and collect the routes of 20 volunteers from the supermarket to our office. Those volunteers are familiar with the roads and drive with their own preferences. As shown in Fig. 1(b), their routes match well with those recommended by Google Maps in the city's



(a) Google Maps (b) Trajectory dataset

Fig. 1: From Marker A to the destination, 5% users drive on Route 1, 20% users drive on Route 2 following Google Maps and 75% users choose Route 3.

major road level. When we are near the starting point or close to the destination, the routes chosen by the volunteers however become quite diverse and deviate significantly from the Google recommended routes. Yet our field check suggests that these routes are highly practical and reasonable. These local users are much more familiar with the exact road conditions in the campus, including the intersections, backstreets and roadside parking slots, and their choices are therefore often better than the recommendation from Google Maps.

In fact, quite often a driver is puzzled by the *last mile* near the destination, e.g., a building in a campus, where Google Maps or other similar navigation services provide little detailed or ineffective guidelines. Fig. 2 shows a common example, where the destination is a drive-through coffee bar near a highway. Google Maps provides a straightforward route, as indicated by the solid line in Fig. 2. Although the destination appears in the line-sight for a driver in the last mile, s/he is effectively stranded to access the coffee bar given the road divider. Local drivers, however, will choose the two dashed lines, which are not straightforward but are accessible. In realworld driving scenarios, drivers will be busy in differentiating the landmarks, unclear shortcut roads, and

[1]Google I/O 2013 session (Google Maps: Into the future). https://www.youtube.com/watch?v=sBAd89C4Q8Q.

978-1-4673-9955-5/16/$31.00 ©2016 IEEE

available parking lots in a strange environment. These make searching the final destination even more difficult.

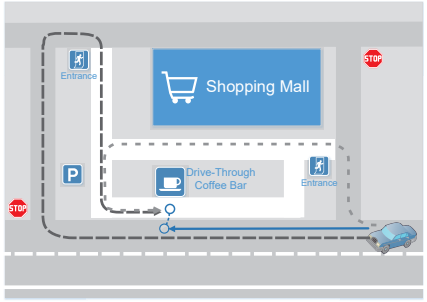


Fig. 2: Sketch of last segment from real world instance.

In short, with detailed map of the transport network and even realtime traffic, existing navigation services provide fastest routes in the major route level; once entering the *last mile*, they unfortunately can be ineffective and, instead, local drivers often have a better understanding of the routes there. These motivate our design of *CrowdNavi*, a complementary service to existing navigation systems, seeking to combat the last mile puzzle. CrowdNavi collects the crowdsourced driving information from users to identify their local driving patterns, and recommend the best local routes for users to reach their destinations. In this paper, we present the architectural design of CrowdNavi and identifies the unique challenges therein, particularly on identifying the last segment in a route from the crowsourced driving information and navigator drivers through the last segment. The challenge however is on (1) how can we identify the last segment? and (2) how can we identify the best route toward the destination in the last segment? Such decisions are to be inferred from massive trajectory information from the crowd. We offer a complete set of algorithms to identify the last segment from the drivers' trajectories and scoring the landmark. We then present an effective navigation algorithm to locate the best route along the landmarks for the last segment. We have implemented the CrowdNavi app on Android mobile OS, and have examined its performance under various circumstances. The experimental results demonstrate its superiority in navigating drivers in the last segment toward the destination.

The rest of this paper is organized as follows. In Section II we present an overview of CrowdNavi as well as the key challenges in its design. In Section III, we discuss the design principles and algorithms in different modules. In Section IV, we compare CrowdNavi with Google Maps based on our real-world case study. Section V provides a literature review, Section VI concludes the paper and discusses the future work.

II. CrowdNavi: Architecture and Challenges

Fig. 3 shows the overall architecture and workflow of CrowdNavi. Its user app of CrowdNavi will be installed in drivers' mobile devices, e.g., a smartphone or 3G/4G-enabled car consoles. If enabled, the app will run in the background, monitoring the moving trajectory of a car using GPS, and periodically reporting the location information to a backend server. The server accordingly maintains a database on the trajectory information of the app users. When a user needs to find the route to a destination[2], the request will be forwarded to the server, which will first identify the *last segment* closed to the destination. The route before the last segment will be recommended by an external map service (e.g., Google Maps), and that for the last segment will then be calculated by the server using the database of the driving pattern from the crowd. We emphasize that the partitioning of the route is necessary given that the local drivers have the best (and sometimes the very only) knowledge for the last segment, as we have shown. In the next section, we will formulate this last segment navigation problem, and present a solution framework with trajectory data modeling, landmark scoring, and best route recommendation, which consists of the core modules of CrowdNavi.

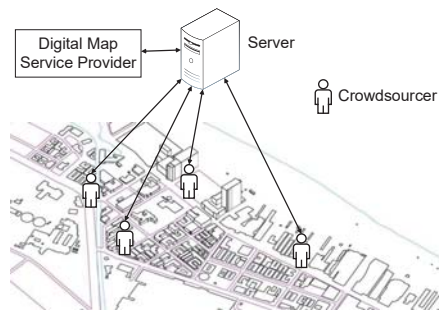


Fig. 3: Architecture of CrowdNavi

Before we proceed with the detailed solutions for the individual modules of CrowdNavi, we first summarize the key notations in Table I. As mentioned, the server collects the trajectories of moving users from the app. For destination d_j, all trajectories toward this destination reported a trajectory set $\mathcal{T}_j = \{T_1^j, T_2^j, ..\}$, in which each trajectory T_i^j is a a sequence of GPS points, ended with d_j. That is, $T_i^j = (n_1, n_2, ...)$, where n_k is a tuple of the longitude, latitude and timestamp of the corresponding GPS sample. Since we focus on last segment navigation toward the destination, we hereby omit the source index for ease of exposition. Note that different trajectories may intersect or overlap with each other, and a user may have multiple trajectories to destination d_j in the crowdsourced reporting context, e.g., the user visits his/her office (as the destination) every day and reports the trajectory for each visit.

To avoid dealing with the massive individual points on the trajectories, we rely on *landmarks* V for calculating routes, where each landmark v_i stands for a geographic region that is of particular significance, e.g., the entrance to a campus or an intersection where many trajectories meet. Let $\Omega(v_i)$ denote the set of users whose historical trajectories appear on landmark v_i, and $\Theta(v_i)$ denote the set of the historical

[2]Note that, to use the navigation service, the user does not have to enable the app all the time to report the moving trajectory, although in this case the user will not help with populating the database.

TABLE I: Summary of Notations

d_j	destination j
t	time
$\mathcal{T}_j$	trajectory set to destination d_j
T_i^j	trajectory i to destination d_j in $\mathcal{T}_j$
n_i	trajectory point n_i
v_i	landmark
V	sets of landmark v_i
x_i	entrance landmark
X	entrance landmark set
L	last segments in trajectory set $\mathcal{T}_j$
e_{ij}	landmark edge between v_i and v_j.
E	sets of landmark edge e_ij
$G = (V, E)$	routable landmark graph
u_k	user k
s_k	user sense of u_k
p_i	landmark popularity of landmark v_i
q_i	user preference on landmark v_i

trajectories passing through landmark v_i. To formulate the last segment navigation problem, we first introduce the concept of *entrance landmark* set X, which is a subset of V, such that $\bigcup_{x_i \in X} \Theta(x_i) = \mathcal{T}_j$ and its cost $\sum_{x_i \in X} C(x_i, d_j)$ is minimized. The cost $C(x_i, d_j)$ is the driving distance from x_i to d_j, i.e., the length of the corresponding trajectory segment. Intuitively, all the trajectories toward the destination will pass through at least one landmark in this set, and these entrance landmarks are the closest to the destination. If two landmarks v_i and v_k are directly connected by a certain trajectory, i.e., the trajectory sequentially pass through v_i and v_k without detouring, we say there is a *landmark edge* (or *edge* in short) e_{ik} connecting them. We then have a *routable landmark network* $G = (V, E)$, where V is the set of all landmarks and E is the set of all edges. Given network G, a *route* from landmark v_i to destination d_j is formed by a sequence of connected landmarks, assuming that d_j is a landmark by default.

Definition 1 (Last Segment):

The last segment L is a sub-graph of $G = (V, E)$ that contains all possible routes from *entrance landmark* $x_i \in X$ to destination d_j, i.e., $L = \bigcup_{x_i \in X} \{r(x_i, d_j)\}$.

As mentioned, the trajectories from the crowd can be highly diverse, particularly around the destination. The last segment navigation therefore is to identify the best entrance landmark and the route toward the destination, as follows:

Problem Definition Given a user query to destination d_j, identify the *entrance landmark set X*, and find the favourite route $r^*(x_i, d_j)$ for any $x_i \in X$ from the last segment L.

III. Identifying and Navigating the Last Segment

The navigation problem may be solved by recruiting experienced users to manually mark the last segments, which however can be labor-intensive and time-consuming, and is hardly scalable with massive user bases. The preference of the manual marking is also questionable. Instead, our CrowdNavi explores the rich information inside the trajectory data from its users, trying to identify the last segment as well as the best route toward the destination from the sheer amount of data.

A. Identifying the Last Segment

We start from the preprocess step to construct a routable landmark graph for identifying all landmarks from trajectory dataset. We discover geographical areas and divide a geographical range into disjoint clusters with a spatial clustering algorithm. DBSCAN [2] is a notably representative, which, given a set of points, groups together points that are closely packed together. Based on the routable graph $G = (V, E)$, we extract out all the historical trajectories $\mathcal{T}_j$ to destination d_j and search the *last segment L* with the spatial and social-community properties in the destination area. The recognition for an entrance landmark depends on two landmark parameters $\Theta(v_i)$ of landmark v_i and driving distance $C(v_i, d_j)$ from v_i to destination d_j. These landmarks can be regarded as *entrance landmark*, when they satisfy the following two conditions. One is that $\Theta(v_i)$ of landmark v_i is exceeding most other landmarks, which means this landmark v_i is located on essential routes to destination d_j. The other condition is that the sum of driving distance $C(v_i, d_j)$ in entrance landmark set should be as small as possible. In most cases, the entrance landmark occurs when people drive or walk away from some certain roads and search for a specific entrance, or are confused by the surrounding environment. Our objective is to find last segment L for destination d_j based on trajectory dataset $\mathcal{T}_j$. With definition 1 for last segment, we formulate this problem of searching entrance landmark of last segment as a minimum weighted set cover problem. Each landmark $v_i \in V$ has a variable $\beta_i \in \{0, 1\}$ which defines whether v_i is in the set of entrance landmark X.

$$\begin{aligned} \text{Minimize} \quad & \sum_{v_i \in V} C(v_i, d_j) * \beta_i \\ \text{s.t.} \quad & \bigcup_{T_k \in \Theta(v_i)} T_k \geq \mathcal{T}_j, \forall v_i \in V \\ & \beta_i \in \{0, 1\}, \forall v_i \in V. \end{aligned} \tag{1}$$

The problem of finding a minimum weighted set cover is a classical optimization problem with approximation algorithms [3]. With the entrance landmark set $X = \{v_i | \beta_i = 1, v_i \in V\}$, we can obtain the last segment set $L = \bigcup_{x_i \in X} \{r(x_i, d_j)\}$ with all possible routes to the destination d_j based on Definition 1.

B. Scoring Landmark

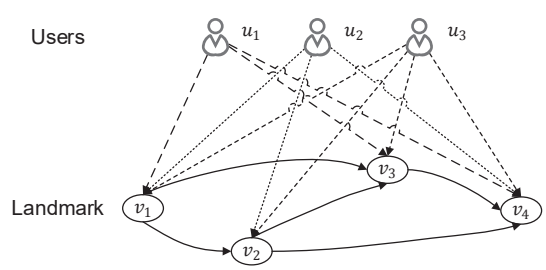


Fig. 4: Landmark scoring model.

We propose a landmark scoring model to quantify the landmark preference q_i , as Fig. 4 shows. Fig. 4 illustrates the

main idea of landmark scoring model, which is inspired by [4]. There are two kinds of nodes, i.e. users and landmarks, where a user has passed through many landmarks and a landmark has been passed through by many users. For example, user u_1 passed through landmarks $v_1 \rightarrow v_3 \rightarrow v_4$, where u_1 points to landmark v_1, v_2 and v_3. Thus a good user can point to many good landmarks, and a good landmark is pointed by many good users. We propose the *landmark preference* and the *user sense* for each landmark and user, respectively. Yet, the user sense and the landmark preference has a more complicated mutual reinforcement relationship than [4], which will be clarified later. Using a power iteration method, the user sense and the landmark preference can be calculated.

Definition 2 (Landmark Popularity): We define the popularity p_i for landmark v_i as the portion of the users whose trajectories appear on landmark v_i. That is,

$$p_i = \frac{|\Theta(v_i)|}{|\mathcal{T}_j|}, v_i \in V \tag{2}$$

where $\Theta(v_i)$ is a set of trajectories passed through landmark v_i and $\mathcal{T}_j$ is the set of trajectories toward destination d_j. The popularity p_i is the fraction of users who like the landmark.

Our main idea to calculate the landmark preference is based on the facts that a high preference landmark will increase its popularity much more rapidly than others, when a large fraction of highly experienced (user sense) drivers redirects to the landmark. Therefore, by observing the increase of landmark popularity with user sense, we now estimate and derive the landmark preference evolution function over time under our model. When we know the current landmark popularity p_i, we can estimate how many new users will pass through landmark v_i. Therefore, we define the preference q_i of a landmark v_i as the following equation:

Definition 3 (User Preference): User preference q_i denotes that all users in the community like the landmark v_i or not. Specifically, user preference q_i on the landmark v_i will increase, if many local users with high *user sense* have considerable trajectories on the landmark v_i.

$$q_i = \sum_{u_k \in \Omega(v_i)} \frac{\mathrm{d}f(u_k, v_i, t)}{\mathrm{d}t} \cdot s_k + p_i \tag{3}$$

where $f(u_k, v_i)$ is frequency of user u_k that passed through landmark v_i. $\Omega(v_i)$ is the set of users who passed through landmark v_i. For ease of computation, the *user preference* q_i has been normalized by the corresponding sum of total user preference. s_k is user sense of user u_k who is aware of this landmark v_i as following.

Definition 4 (User Sense): The sense of user s_k is sum of *landmark preference* q_i , which represents the knowledge of user u_k in last segment.

$$s_k = \sum_{v_i \in u_k} q_i \tag{4}$$

The range of s_k value [0,1], where the initial value is 0. When the user knows each landmark, the value is 1.

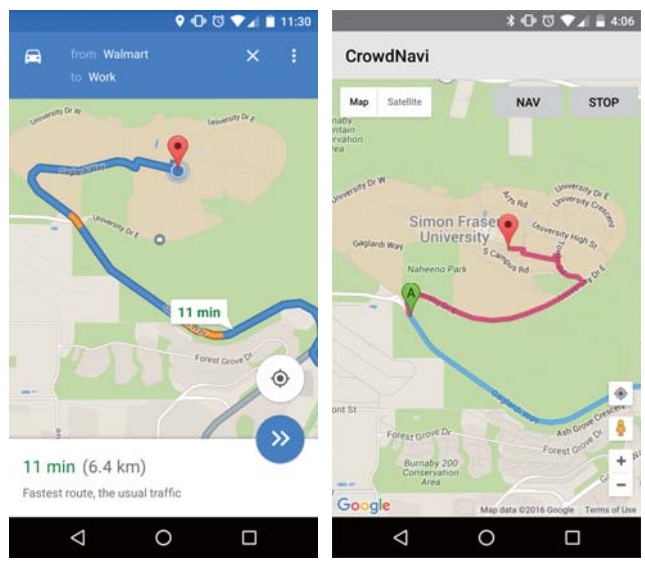
C. Last Segment Routing

With the last segment L and the set of entrance landmarks X, we can build a vertex weighted last segment graph, which is $G' = (V', E', w), V' = \{v_i | v_i \in L\}, E' = \{e_{ij} | e_{ij} \in L\}$ and $w = \{q_i | v_i \in V'\}$. We now proceed to find the optimal entrance landmark $x_i \in X$ with an favourite route $r^*(x_i, d_j) \in X$, so as to maximize the weight of the minimum-weight vertex with Multiple-sources single-sink graph. This problem can be converted to a single-source single-sink problem by introducing a dummy source vertex v_0 into V' that is connected to the original source vertices $x_i \in X$. We solve it as finding a shortest path of maximum bottleneck in weighted vertex graph $G' = (V', E', w)$.

IV. Evaluation

This section presents the evaluation of our CrowdNavi system. CrowdNavi provides the best route for each destination in last mile area, which is evaluated against the top-1 result provided by the Google Directions API[3].

A. Case Studies



(a) Google Maps (b) CrowdNavi

Fig. 5: Case studies of Google Maps and CrowdNavi app.

Before conducting the quantitative performance evaluation, we provide one instance as case studies to demonstrate the effective of CrowdNavi system. The destination of the example in Fig. 5 is one building of our workplace on the campus, which is the same in Fig. 1. We collect hundreds of routes for this destination, where there are many alternative routes. Fig. 5(a) shows that the fastest route provided by Google Maps from user current location. Yet this route is not a good choice with various weakness, such as narrow roads with many crossroads, many people walk on the campus road, coinciding with the bus lines and many reserved street parking for campus service vehicles. As shown in Fig. 5(b), CrowdNavi

[3]https://developers.google.com/maps/documentation/directions/

recommends a high way on the edge of the campus, which is the most convenient road to the work place.

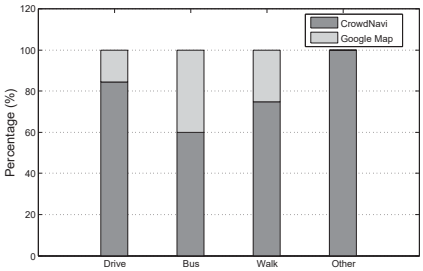


Fig. 6: Web survey comparison of CrowdNavi against Google Maps.

We have conducted a user questionnaire survey to further explore the results. Most of the existing studies on navigation services collect the feedback from volunteers to derive the user experience. Trying to directly understand the QoS of CrowdNavi and real preferences of users on different routes, we have invited local people to fill in our web survey. 90 people have participated in the survey, where 96.67% work or study in our campus and 73.33% know the destination in Fig. 5. The survey contains a series of single-choice questions plus several questions on insensitive personal information. The result is gratified that only 30% of users will not choose our route in Fig. 5(b). Of the users who do not select our route in Fig. 5(b), 78% participants take buses daily to campus and are not aware of the route in Fig. 5(b).

B. Large Scale Experiments

It is very difficult to directly evaluate whether a customized route provided by CrowdNavi for a real user is the actually best one due to the following two reasons. First, a user can only drive on one route at a given time. The user would never know whether other routes are better than the driven one. It is not reasonable that a different user to travel another route simultaneously with their different drive behaviors and knowledge to the evaluation. Second, it is not reasonable to request a single user to drive two different routes separately, since the user can learn from their past driving experiences. The route driven later will benefit from the first test.

To address the above challenges, we closely collaborate with Mojio[4] to analyze their user logs, which can be used as ground truths in the experiments. Mojio connects the vehicles into Internet with a small device that is installed on the vehicle on-board diagnostics port, so it can work even without the smartphones. When a user starts the engine, the vehicle data will be sent to the Mojio cloud platform. The trace data records such information as the time, vehicle model, GPS position and driver operations, which enable us to extract rich information. Starting from 13 Aug 2015 to 27 Aug 2015, Mojio collected the detailed information from 922 vehicles with 6154134 records. We can evaluation the performance of CrowdNavi based on the user selections from Mojio data.

[4]https://www.moj.io/

TABLE II: Mojio Data

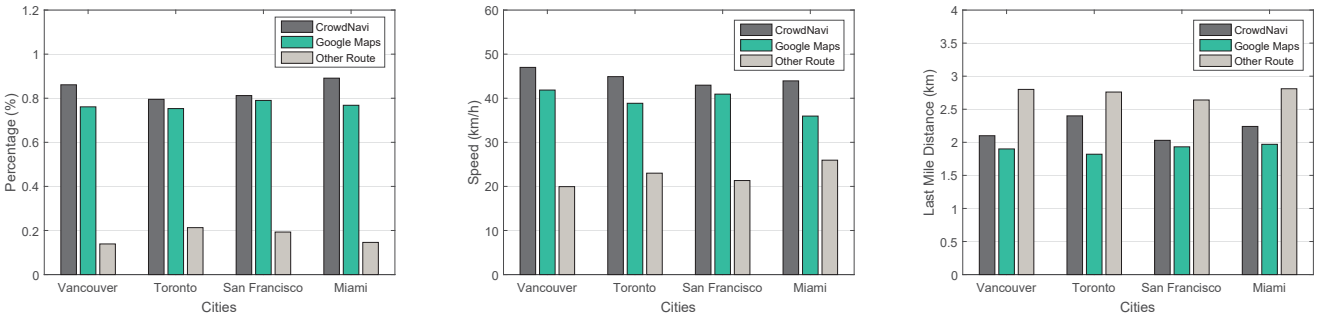
	Records	trajectories	Vehicle	Destinations
Vancouver	459922	4228	191	118
Toronto	194789	1626	133	38
San Francisco	148116	1329	106	45
Miami	720247	5806	256	187

To examine the performance of CrowdNavi, we have conducted experiments in four cities in North American, involving Vancouver, Toronto, San Francisco and Miami. As Table II shows, we select 686 vehicles logs with 12,989 trajectories. 75% trajectories can be regraded as the training dataset and the remaining data is used as the validation dataset. Fig. 7 show CrowdNavi perform well in the large-scale experiments. In particular, Fig. 7(a) illustrate the percentage of the candidate route chosen by the drivers. The results demonstrate that the route of CrowdNavi is more preferred than Google Maps and other routes. We also observe that there are many overlaps between CrowdNavi and Google Maps, which means that CrowdNavi and Google Maps recommend the same route in many cases. We can further explore the difference between CrowdNavi and Google Maps in Fig. 7(b) and 7(c). Fig. 7(b) plots the average speed the users traverse in last mile area, and Fig. 7(c) shows the average distance the users traverse in last mile area. The routes provided by CrowdNavi have higher speed than Google Maps, while the routes of Google Maps have less distance. The above figures together illustrate the preference of most drivers in last mile that drivers would prefer the routes with higher speed rather than the shortest path. The higher speed routes usually contain less STOP signs, less turns and less light traffic, although those routes may have longer than the shortest one. Although the difference of the spent time is trivial in the last mile driving, CrowdNavi usually provides an easier accessible route than Google Maps, e.g., CrowdNavi can recommend a correct road side to avoid crossing the streets. Moreover, we analysis the unpopular routes in the logs, e.g., local people occasionally may drive in the detours, or the inexperienced drivers lost directions in the last mile.

V. Related Work

In this section, we review the related works and clarify the differences as compared to our CrowdNavi.

Route planning plays an important role in our work. Finding the shortest path [5] has been extensively studied for over fifty years. Chen et al. [6] focus on finding the top-k best connected trajectories that are geographically optimal to connect user-given locations. T-drive [7] and the followup work [8] finds the fastest routes to a destination through gradually learning users' driving behaviour from historical GPS logs reported by a large collection of Taxis. In [9], the problem of popular route planning without road network information is investigated. Wei et al. [10] search for popular and attractive travel routes, aiming at finding the top-k routes from uncertain trajectories. The authors in [11] also raised the last mile problem, yet they only discuss the indoor navigating with a magnetic approach, whose context is different with CrowdNavi. Our CrowdNavi focus the last segment navigation with crowdsourced driving



(a) The percentage of the candidate route chosen by the drivers

(b) The average speed of the candidate route

(c) The average distance of the candidate route

Fig. 7: The large-scale experiments to compare CrowdNavi and Google Maps

information, serving as a complement to existing online map and routing planning systems.

The idea of *crowdsourcing* has been popular in recent years, particularly with the advancement of modern smart mobile devices and the deep penetration of 3G/4G mobile networks. Google, TomTom, OpenStreetMap [12], and Waze [13] have leveraged crowdsourcing for online map update and maintenance, where user-submitted changes can be integrated into their map products after manual review. CrowdAtlas [14] infers certain missing lanes and street corners from the raw trajectory data using a clustering algorithm. CrowdPlanner [15] further queries human workers to evaluate recommended routes from different sources and determine the best route based on the feedbacks. Trip routes recommendation using different attributes crowdsourced extracted from users, such as user preference,traffic information and endorsing information, have also been explored [4] [16] [17] [18].

Our design of CrowdNavi is motivated by these crowd-sourced map services and related trip recommendation services. Yet we focus on the last segment navigation, not the overall trip, and closely examine its unique challenges. We also seek to fully automate data collection and analysis with minimum human interference.

VI. Conclusion

In this paper, we have presented the architectural design of CrowdNavi as a supplementary to the current digital map services. The unique challenges therein have been illustrated, particularly on identifying the last segment in a route from the crowsourced driving information and guiding drivers through the last segment. We have offered a complete set of algorithms to cluster the landmarks from the drivers' trajectories, identify the entrance landmarks toward the last segment and evaluate the landmarks preferences. We then presented the navigation algorithms to locate the best route along the landmarks toward the destination. We are currently performing larger-scale field test, and also trying to identify more information inside the crowdsourced driving information toward better navigation.

References

[1] V. Ceikute and C. S. Jensen, "Routing service quality–local driver behavior versus routing services," in *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*. IEEE, 2013.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[3] M. Karpinski and A. Zelikovsky, "Approximating dense cases of covering problems," in *the DIMACS Workshop on Network Design: Connectivity and Facilites Location*, vol. 40, 1998, pp. 169–178.

[4] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *the 18th international conference on World wide web*. ACM, 2009.

[5] B. Ding, J. X. Yu, and L. Qin, "Finding time-dependent shortest paths over large graphs," in *the 11th international conference on Extending database technology: Advances in database technology*. ACM, 2008.

[6] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie, "Searching trajectories by locations: an efficiency study," in *the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010.

[7] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: driving directions based on taxi trajectories," in *the 18th SIGSPATIAL International conference on advances in geographic information systems*. ACM, 2010.

[8] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.

[9] Z. Chen, H. T. Shen, and X. Zhou, "Discovering popular routes from trajectories," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE, 2011.

[10] L.-Y. Wei, Y. Zheng, and W.-C. Peng, "Constructing popular routes from uncertain trajectories," in *the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

[11] Y. Shu, K. G. Shin, T. He, and J. Chen, "Last-mile navigation using smartphones," in *Proceedings of the 21st ACM MobiCom'15*. ACM, 2015, pp. 512–524.

[12] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.

[13] "Waze," https://www.waze.com/.

[14] Y. Wang, X. Liu, H. Wei, G. Forman, C. Chen, and Y. Zhu, "Crowdatlas: self-updating maps for cloud and personal use," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013.

[15] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, and X. Zhou, "Crowdplanner: A crowd-based route recommendation system," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014.

[16] R. Carisi, E. Giordano, G. Pau, and M. Gerla, "Enhancing in vehicle digital maps via gps crowdsourcing," in *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*. IEEE, 2011.

[17] H.-P. Hsieh, C.-T. Li, and S.-D. Lin, "Exploiting large-scale check-in data to recommend time-sensitive routes," in *the ACM SIGKDD International Workshop on Urban Computing*. ACM, 2012, pp. 55–62.

[18] L. Liu, J. Xu, S. S. Liao, and H. Chen, "A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3409–3417, 2014.