Crowdsourced Road Navigation: Concept, Design, and Implementation

Xiaoyi Fan, Jiangchuan Liu, Zhi Wang, Yong Jiang, and Xue Liu

The authors provide an overview of past and present road navigation technologies. They discuss recent advances in crowd intelligence, identifying the unique challenges and opportunities therein. They present a case study that utilizes the crowdsourced driving information to combat the last mile puzzle for road navigation.

ABSTRACT

Map services and applications have been studied extensively within the mobile computing community in the past two decades, starting from standalone GPS receivers and then moving toward connected smart terminals with live digital maps of transportation networks and even real-time traffic. More recently, with the deep penetration of modern 3G/4G networking and social networking, the crowd intelligence from the social community has been explored toward crowdsourced navigation. In this article, we first provide an overview of the past and present road navigation technologies. We then discuss very recent advances in crowd intelligence, identifying the unique challenges and opportunities therein. We further present a case study that utilizes the crowdsourced driving information to combat the last mile puzzle for road navigation.

INTRODUCTION

With the advances of outdoor positioning services (GPS in particular), automated road navigation has quickly risen to become a killer application over the past two decades. Earlier generations of navigation services rely on dedicated GPS devices from such major companies as TomTom, Garmin, and Magellan. Given the deep penetration of third/fourth generation (3G/4G) mobile networking and social networking, drivers are now well connected anytime and anywhere; they can readily access information from the Internet and share the information with their community. Online digital map services such as Google Maps are experiencing an explosion in use and serving billions of users on a daily basis.[1] Real-time driving information such as live traffic or construction locations has been incorporated as well. On the macro-scale of a transportation network, the quality of the recommended routes is generally acceptable with state-of-the-art navigation services. However, it is known [1] that the routes from the map-based services often fail to be agreed on by local drivers, who have detailed knowledge of local/dynamic driving conditions. There is great potential in exploring the crowd intelligence toward *crowdsourced navigation*.

In this article, we first provide an overview of past and present road navigation technologies. We then discuss the very recent advances in crowd intelligence, identifying the unique challenges and opportunities therein. Following in chronological order, we summarize the key activities in two stages (Fig. 1):,

1. *Standalone devices*, starting from GPS receivers and then smart devices (smartphones in particular) that seamlessly integrate such positioning technologies as assisted GPS (A-GPS), WiFi positioning, motion sensors, and cellular network positioning
2. *Crowd intelligence*, which has been popular in map building, navigation, and localization, particularly with the advancement of smartphones with social networking

We further present a crowdsourced navigation application as a case study, *CrowdNavi* [2], which incorporates a complete set of algorithms to automatically cluster the landmarks from drivers' trajectories and locate the best route. We highlight the key design and implementation issues therein and demonstrate its superiority with a real-world example for last mile navigation.

The remainder of this article is organized as follows. We present representative works from the early stages of research on standalone devices from GPS to smartphones. We focus on the use of crowdsourced human intelligence in navigation. We then present the case study of CrowdNavi. Finally, we conclude the article.

NAVIGATION WITH STANDALONE DEVICES: FROM GPS TO SMARTPHONES

The cornerstone of any navigation service is a reliable positioning technology, and GPS is no doubt the dominating one. The GPS project, launched by the United States in 1973, provides geolocation and time information to a receiver anywhere on Earth where there is an unobstructed line of sight to four or more GPS satellites. It operates independent of any telephonic or Internet reception. Other similar systems (e.g., Galileo in the European Union and Beidou in China) have been or are being deployed.

For civilian use, GPS can reach a 3.5 m horizontal accuracy. While high-sensitivity GPS chipsets have been adopted in recent years, standalone GPS still does not work well in urban and indoor environments. As a result, complementary positioning systems are employed in smartphones (e.g., cellular network and WiFi positioning techniques).

Assisted GPS: Most smartphones are GPS-enabled and further employ a technology known as A-GPS, where an assistance server provides satel-

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

Digital Object Identifier: 10.1109/MCOM.2017.1600738

Xiaoyi Fan and Jiangchuan Liu are with Simon Fraser University; Zhi Wang and Yong Jiang are with Tsinghua University; Xue Liu is with McGill University.

lite orbit and clock information to counteract GPS signal degradation.

WiFi Positioning: Tens of millions of WiFi access points (APs) in the last decade have been deployed in homes, businesses, retail stores, and public buildings. The density of APs in urban areas is so high that the signals often overlap, creating a natural reference system of geo-locations. WiFi positioning identifies existing WiFi signals and determines the current location of a physical device, typically using triangulation-based solutions.

Motion Sensors: Advanced navigation systems also explore the supplemental information from the rich sensors of smartphones (e.g., accelerometers, gyroscopes, and odometers). Inertial localization methods can be grouped into two different techniques: *dead-reckoning-based* and *pattern-recognition-based*. Dead reckoning localization techniques [3] estimate a user's location through the aggregation of odometry readings from a previously estimated or known position. Pattern-based localization methods [4] use more sensors of the device to find the closest pattern that has been coupled with physical locations.

After obtaining the location, the best route toward the destination can be formulated as a navigating problem, which has been intensively studied for over 50 years in both theory (i.e., fastest/shortest path in graph theory) and the real world [5]. Earlier generations of standalone devices locally calculated the route based on static maps. This was later extended with live traffic or cloud-based calculation; however, the experiences from other drivers are not explored.

New Generation Navigation with Crowd Intelligence

The idea of *crowdsourcing* has received significant attention in the past decade from both academia and industry. Crowdsourcing with an incentive mechanism has been increasingly popular in map building, navigation, and localization, particularly with the advancement of modern smart mobile devices and the deep penetration of 3G/4G mobile networks. Google and TomTom have leveraged crowdsourcing for online map update and maintenance, where user-submitted changes can be integrated into their map products after manual review. Existing commercial map service applications have relied heavily on their users to constantly update, maintain, and improve their services. A representative of such community-driven services, Waze,[2] collects traffic information, incident reports, and complementing map data from users, then provides online navigation. Another example, OpenStreetMap,[3] aims to establish and maintain crowdsourced map infrastructure by its volunteer community, and has attracted millions of contributors and users. The recent activities in this field are summarized in Table 1 with the following categories.

Digital Map Construction: Recent works have suggested that automatic map update can be performed based on user trajectory analysis. CrowdAtlas [13] exploits the crowdsourcing approach by synthesizing raw GPS traces to infer and complement missing lanes. CrowdMap [11] leverages crowdsourced sensors and image data to track user movements, then uses the inferred user motion traces and context of the image to produce an accurate floor plan. CrowdInside [12] utilizes the smartphone sensors that are ubiquitously available with humans and automatically construct accurate motion traces.

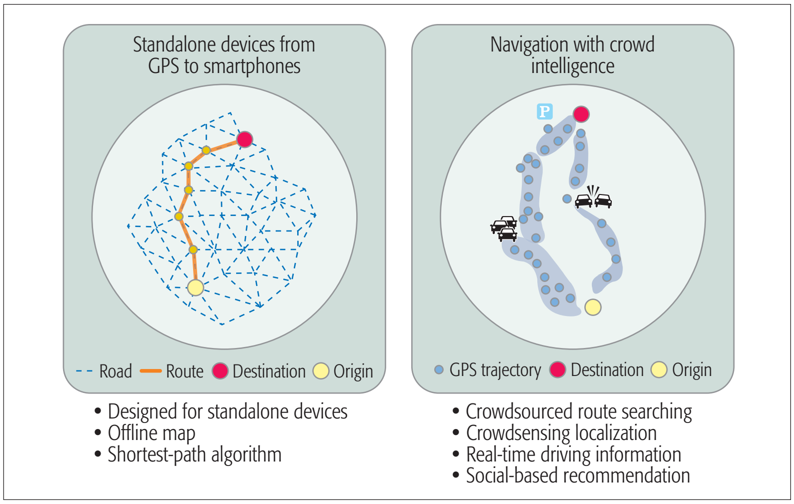

Figure 1. Navigation from standalone devices to the crowd.

Route Planning: The advance of GPS-enabled devices allows people to record their location histories with GPS traces, which contain rich human behaviors and preferences related to travel. In the recent past, a branch of research has been performed based on user location or trajectory history recorded in GPS traces, including searching for the most popular/fastest route [6], mining uncertain trajectories [7], and detecting interesting/preferred routes for users [8]. CrowdPlanner [9] is a crowd-based route recommendation system, which queries human workers to evaluate recommended routes and determine the best route based on feedback. The evaluations completed by humans can be quite different from those fulfilled by computers, since human evaluations contain our knowledge and experiences, which can outmatch the majority of machine algorithms.

Social-Based Recommendation: Social networks and communities have been incorporated into route recommendation systems as well. The emerging social media applications generate huge amounts of spatial data on human activities. Geotagged photos and check-in data can be used to identify how people sequentially visit places in an area. Route recommendation [14] is an interesting topic in terms of the collective knowledge learned from users' historical trajectories. A user needs a sophisticated itinerary conditioned with a sequence of locations and a user's specific travel duration and departure place. The itinerary could include not only standalone locations but also detailed routes connecting these locations and a proper schedule (e.g., the typical time of day that most people reach the location and the appropriate time length that a tourist should stay there).

A key challenge in crowdsourcing systems is to offer enough incentive for a user to contribute her/his experiences. Google Local Guide[4] encourages users to help others with their local experience (e.g., writing reviews or rating restaurants) with benefits in return. It is worth noting that crowdsourced systems identify the best route toward a destination by mining the massive trajec-

[2] https://www.waze.com/

[3] https://www.openstreetmap.org/

[4] https://www.google.com/local/guides/

Authors	Catogries	Collected data	Technique summary
T-drive [6]	Route planning	GPS trajectories	Find fastest routes on historical trajectories
Wei *et al.* [7]	Route planning	GPS trajectories	Popular routes from uncertain trajectories
Zheng *et al.* [8]	Route planning	GPS trajectories	Propose a HITS-based inference model
CrowdNavi [2]	Route planning	GPS trajectories	Navigate in the last mile with crowdsourced driving information
CrowdPlanner [9]	Route planning	Crowdsourced answers	Determine the best route based on human worker answers
Zee [10]	Localization	Motion sensors	Fingerprint crowdsourcing-based indoor localizatoin
CrowdMap [11]	Digital map construction	Image and motion sensors	Producing floor plans based on image and motion sensors
CrowdInside [12]	Digital map construction	Motion sensors	Producing floor plans based on motion traces
CrowdAtlas [13]	Digital map construction	GPS trajectories	Infer and complement missing lanes
Yoon *et al.* [14]	Recommendation	GPS trajectories	Rank itinerary from user-generated GPS trajectories
Google Local Guide	Recommendation	Crowdsourced answers	Encourage users to help others with their local experience
Wang *et al.* [15]	Location authentication	WiFi APs	Use physical devices to complete proximity authentication

Table 1. Crowdsourcing in navigation application.

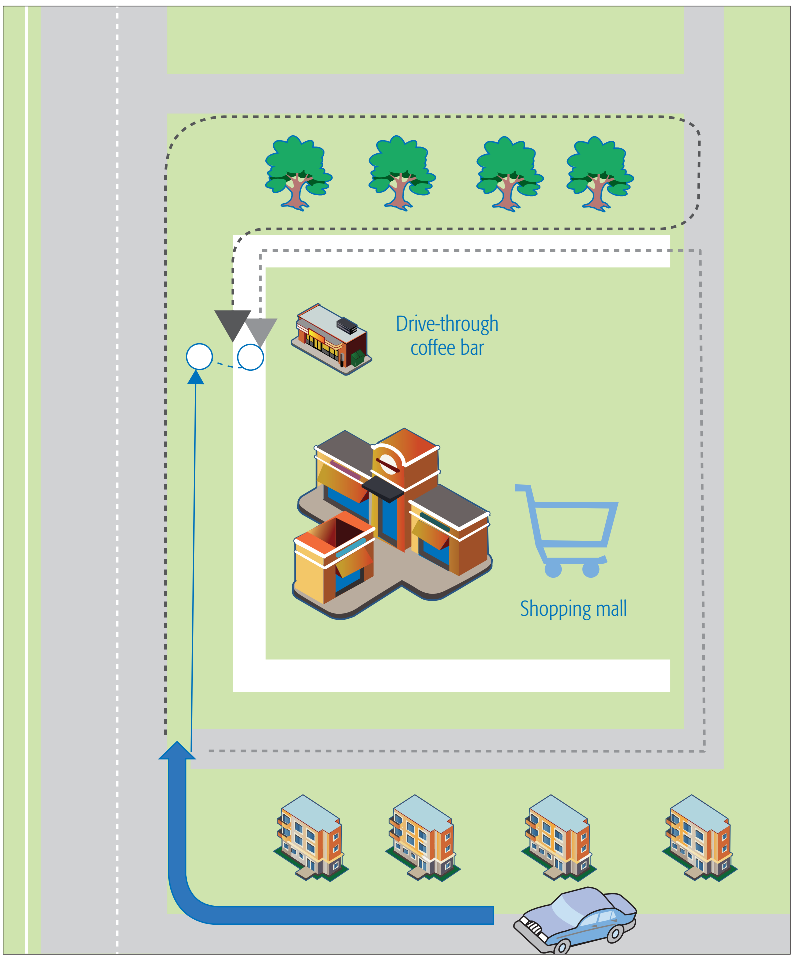

Figure 2. Sketch of the last segment from a real-world instance.

tory information from the crowd, and therefore should be automated with minimum user interference. Crowdsourced systems also rely on real-time data from many users for trip planning and route selection, and thus malicious attacks can influence the decisions. Fortunately, crowdsensing mobile devices are widely equipped with sophisticated embedded sensors, involving accelerometers, digital compasses, and gyroscopes, which provide opportunities to defend against mischievous devices using such techniques as fingerprinting [10]. For example, Wang *et al.* [15] use physical devices to complete proximity authentication, as this is ineffective when attackers have physical access to the phone.

Case Study: CrowdNavi

We now use *CrowdNavi*, a real-world app, as a concrete case to discuss the challenges and solutions in designing and implementing a crowdsourced navigation system. CrowdNavi is a complementary service to existing navigation systems, particularly focusing on the final road segment toward the destination. With a detailed map of the transportation network and even real-time traffic, existing navigation services provide fastest or shortest routes at the major route level. But quite often a driver is puzzled by the *last mile* near the destination (e.g., a building on a campus), where Google Maps and similar navigation services provide less detailed guidelines or simply fail. Figure 2 shows a simple 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 from access to the coffee bar given the road divider. Local drivers, however, will choose the two dashed lines, which are not straightforward but are accessible. In real-world driving scenarios, drivers will be busy differentiating the landmarks, unclear shortcut roads, and available parking lots in a strange environment. These make searching the final destination even more difficult. To combat this last mile puzzle, CrowdNavi collects the crowdsourced driving information from users to identify their local driving patterns, and recommend to users the best local routes to their destinations.

Architecture and Design

CrowdNavi is to be installed on drivers' mobile devices (e.g., smartphones 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 of the trajectory information of the app users. When a user needs to find the route to a destination,[5] the request will be forwarded to the server, which will first identify the *last segment* closest to the destination. The route before the last segment will be recommended by an external map service (e.g., Google Maps). The last segment will then be calculated by the server using the database of the driving

[5] 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 the user will not help with populating the database.

pattern gathered from the crowd.

To this end, CrowdNavi incorporates a landmark scoring model (Fig. 3) inspired by [8]. There are two kinds of nodes, 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 landmarks v_1, v_3, and v_4. Thus, a good user can point to many good landmarks, and a good landmark is pointed out by many good users.

The concepts of *landmark preference* and *user sense* can then be developed for each landmark and user, respectively. The landmark preference is calculated based on the facts that a high preference landmark will increase in popularity much more rapidly than others when a large fraction of highly experienced (user sense) drivers redirect to the landmark. Therefore, by observing the increase of landmark popularity with user sense, a landmark preference evolution function can be estimated over time. With the knowledge of user preference in the last segment area, we can build a vertex-weighted last segment graph. Finding the optimal route then becomes to maximize the weight of the minimum-weight vertex with a multiple-source single-sink graph. This problem can be converted to a single-source single-sink problem by introducing a dummy source vertex and solving it to find the shortest path of the maximum bottleneck in a weighted vertex graph.

Based on this model, the quality of the routes and drivers are graded by other users' experiences, which enables the incentive to motivate experienced drivers to share resources. Different from Google Local Guide,[6] users in CrowdNavi only need to enable the *contribution* option so that CrowdNavi collects the crowdsourced driving information to identify their local driving patterns. Therefore, instead of setting a standardized pricing rule, CrowdNavi can identify highly experienced drivers and give them more benefits.

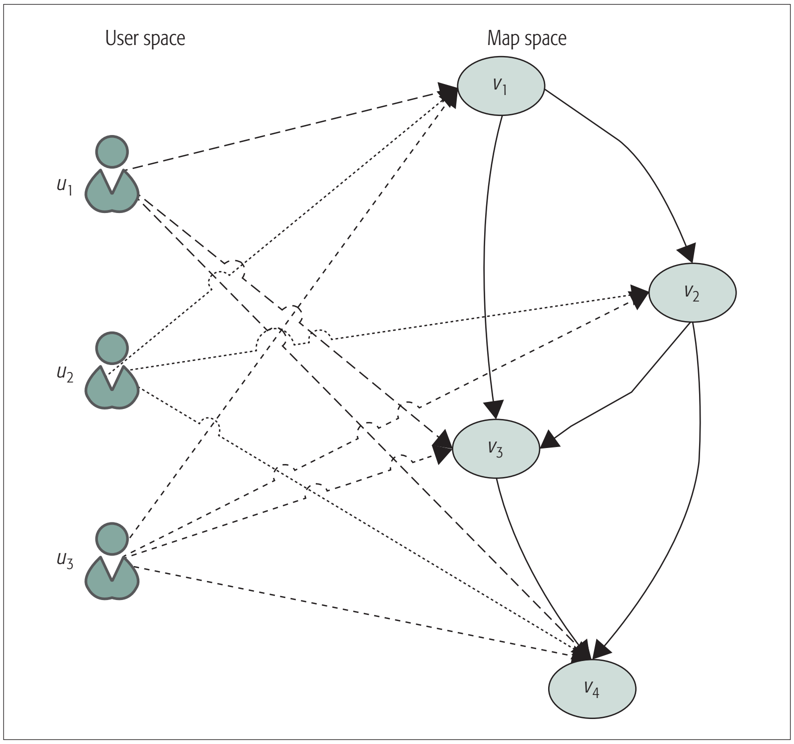

Figure 3. Landmark scoring model.

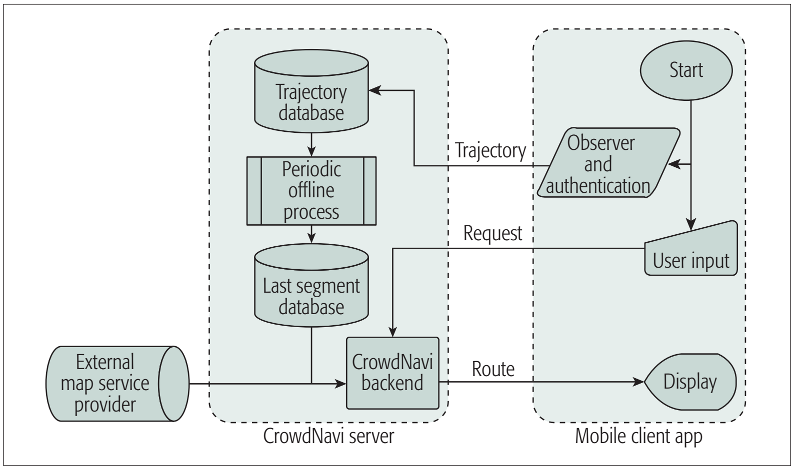

Figure 4. CrowdNavi workflow.

Implementation in Android

We have implemented *CrowdNavi* app in Android OS 5.1.1 working together with the Google Maps Android application programming interface (API).[7] Figure 4 shows the workflow between the mobile client app and the CrowdNavi server. The system follows a simple client-server architecture. The processes run offline and periodically execute our algorithms to find the favorite routes based on the trajectory data. We deploy the database on the servers to store the trajectories and last segment information. The CrowdNavi Backend module on the server allows mobile clients to query the route. The route requests and routes are delivered by JavaScript Object Notation (JSON), which is easy to extend to other platforms. CrowdNavi can be deployed on public cloud platforms and allow efficient big data processing to handle the massive trajectory data (e.g., Apache Hadoop[8] and Apache Hive[9]), which enables higher scalability and stability with lower implementation costs.

The client side is based on Java in the Android-studio programming environment. Figureure 5b shows the interfaces of the mobile client app of CrowdNavi running on a Google Nexus 5 Android smartphone, which is similar to the Google Maps app. CrowdNavi also runs in the background as an observer to sample and buffer the user's real-time GPS streams. When high-quality Internet connectivity is available (e.g., Wi-Fi or a 3G/4G mobile network), CrowdNavi uploads the trajectory data in a batch to the server serving as a crowdsourcer. To minimize the network cost and energy consumption on the mobile device, we have included an optimized implementation of packet caching and compress on the mobile client app. Our implementation of trajectory data caching is able to balance the network cost and the mobile resources with caching and compressing data by a mobile device. If transmission failures occur, CrowdNavi will store data in the local storage and an available Wi-Fi connection can trigger batch data transmission.

CrowdNavi servers are deployed on an m4.2xlarge instance of the Amazon EC2 cloud platform, as Fig. 4 shows. The trajectory data from

[6] https://www.google.com/local/guides/

[7] https://developers.google.com/maps/documentation/android/

[8] http://hadoop.apache.org/

[9] https://hive.apache.org/

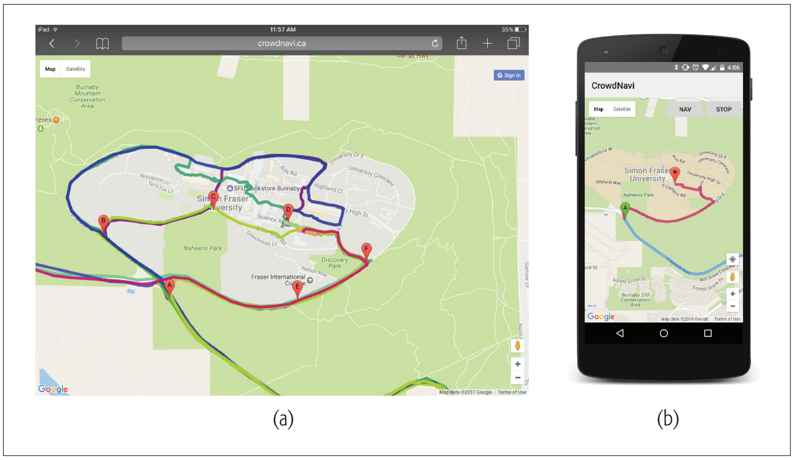

Figure 5. a) From marker A to destination D, 20 percent of users drive on route (A-B-C-D) following Google Maps, and 75 percent of users choose route (A-E-F-D); b) a screen sample of the CrowdNavi App.

users can be inserted into the trajectory database. Last segment and landmark preference calculation are the computation bottlenecks in our approach, since it involves an algorithm with power iterations. This algorithm further invokes the costly bottleneck of path calculation for each last segment. The iterative algorithms are executed periodically offline, which targets a relatively stable landmark preference. In practice, the landmark preference for each last segment can converge in tens of iterations. As discussed, CrowdNavi is a complement for Google Maps in the last segment, which provides the route and relies on the Google Directions API for searching for the rest of the route.

Location Authentication in CrowdNavi

As mentioned earlier, crowdsourced systems are inherently vulnerable to mischievous or malicious users who are seeking to disrupt the system [15] For example, malicious users supported by the competitors of navigation applications can forge their physical locations to mislead our system, or retail business owners can guide potential customers to their stores by contributing false negative routes. This is exacerbated by the fact that there are no widely deployed techniques to authenticate the location of mobile devices. CrowdNavi as a crowdsourced service faces a number of security vulnerabilities and is making efforts on location authentication.

Significant attacks against crowdsourced maps applications can be achieved by using widely available mobile device emulators that run on computers. Most mobile emulators run a full OS (e.g., Android, iOS) and simulate hardware features such as a camera and GPS. Attackers can generate user actions on CrowdNavi, such as clicking buttons and typing text, and feed predesigned GPS sequences to simulate location positioning and device movement. By controlling the timing of the GPS updates, they can simulate any movement speed of the emulated devices. By exploiting this vulnerability, attackers can create fake traffic hotspots and misleading routes at any location on the map.

The main idea is that the messages describing the device moving from inertial sensors should keep consistent with the data from the GPS device, and CrowdNavi uses the cross-validation method among the multiple sensors to verify the physical status of mobile devices. CrowdNavi holds the assumption that the multisensor cross-validation method requires the messages from inertial sensors at the millisecond level, and it is difficult for attackers to generate such accurate and dense data. In particular, GPS locations in short intervals (5–10 s) can be precisely predicted and matched with the estimated trajectory, where a sequence of inertial sensor readings are used to generate the estimated trajectory. Smartphones with inertial navigation systems contain accelerometers, gyroscopes, and magnetometers, which can track orientation and position changes and the users' absolute direction. The messages from those sensors are extremely dense and difficult to emulate, which increases difficulties and limits the ability to amplify the potential damage incurred by any single attacker. Since precise location inference by inertial sensors is possible in 1 or 2 min, CrowdNavi uses the messages from inertial sensors for location authentication, where the estimation errors have not yet aggregated in such a short interval (5 s).

CrowdNavi depends on the known GPS position and velocity at $t - 1$ time as the initial status and then uses a set of messages from inertial sensors during $[t - 1, t]$ time to compute the trajectory in a few seconds. Assuming the sensor readings are correct and sensitive, CrowdNavi applies a physics approach to estimate the coming device location and compare it with the GPS point at t time. In particular, CrowdNavi gets the device's moving orientation by the Android API (`SensorManager.getOrientation()`), as well as the acceleration along this moving direction. When the estimated trajectory is frequently mismatched with the roads on the map or the inertial inferred location is often inconsistent with the coming GPS point, CrowdNavi grades this device as an attacker, and its trajectory as discarded.

A Running Example

We now use an example to demonstrate the effectiveness of the CrowdNavi system. The example, running on our campus, which is known as complex for navigation, has the building of our workplace as the destination. We first search Google Maps, which provides one route (A-B-C-D), as shown in Fig. 5a. CrowdNavi collects the routes from 100 volunteers who are familiar with the roads and drive with their own preferences. As shown in Fig. 5a, their routes match well with those recommended by Google Maps at the city's major road level. However, when we are near starting point A or close to the destination, the routes chosen by the volunteers become quite diverse and deviate significantly from the Google recommended routes. Route (A-B-C-D) in Fig. 5a is not a good choice with various weaknesses, including narrow roads with many crossroads, many people walking on the campus road, coinciding with bus lines and much reserved street parking for campus service vehicles. These volunteers are much more familiar with the exact road conditions on the campus, including the intersections, back streets, and roadside parking slots. The routes in CrowdNavi are therefore often better than the recommendation from Google Maps. As shown in Fig. 5b, CrowdNavi recommends a highway (A-E-F-D) on the edge of the campus, where our field check suggests that this route is highly practical and reasonable.

We have conducted a user questionnaire survey to further explore the results. Most of the existing studies on navigation services collect feedback from volunteers to derive the user experience. Trying to directly understand the quality of service (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 percent work or study on our campus, and 73.33 percent 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 gratifying: only 30 percent of users will not choose our route in Fig. 5b. Of the users who do not select our route, 78 percent of participants take buses daily to campus and are not aware of the route in Fig. 5b.

Conclusion

In this article, we have presented a retrospective view of past and present road navigation technologies, followed by very recent advances with crowd intelligence. We have presented the design principles of CrowdNavi, a practical crowdsourced navigation system, as a supplement to the current digital map services. The unique challenges therein have been illustrated, particularly in identifying the last segment in a route from the crowdsourced driving information and guiding drivers through the last segment.

Crowdsourced systems are inherently vulnerable to mischievous or malicious users that are seeking to disrupt the system. For example, malicious users supported by the competitors of navigation applications can forge their physical locations to mislead our system, or retail business owners can guide the potential customers to their stores by contributing false negative routes.

Acknowledgments

This research is supported by an NSERC Discovery Grant, a Strategic Project Grant, an E.W.R. Steacie Memorial Fellowship, and an Industrial Canada Technology Demonstration Program (TDP) grant. Zhi's work is partly supported by NSFC under Grant No. 61402247.

References

[1] V. Ceikute and C. S. Jensen, "Routing Service Quality — Local Driver Behavior versus Routing Services," *2013 IEEE 14th Int'l. Conf. Mobile Data Management*, 2013.
[2] X. Fan *et al.*, "Navigating the Last Mile with Crowdsourced Driving Information," *2016 IEEE INFOCOM Wksps.),* 2016, pp. 346–51.
[3] A. R. Pratama *et al.*., "Smartphone-Based Pedestrian Dead Reckoning as an Indoor Positioning System," *2012 Int'l. Conf. System Eng. and Tech.*, 2012, pp. 1–6.
[4] S. Grzonka *et al.*, "Activity-Based Estimation of Human Trajectories," *IEEE Trans. Robotics*, vol. 28, no. 1, 2012, pp. 234–45.
[5] E. Kanoulas *et al.*, "Finding Fastest Paths on a Road Network with Speed Patterns," *22nd Int'l. Conf. Data Engineering*, 2006, pp. 10–10.
[6] J. Yuan *et al.*, "Driving with Knowledge from the Physical World," *17th ACM SIGKDD Int'l. Conf. Knowledge Discovery and Data Mining.*, 2011.
[7] L.-Y. Wei, Y. Zheng, and W.-C. Peng, "Constructing Popular Routes from Uncertain Trajectories," *18th ACM SIGKDD Int'l. Conf. Knowledge Discovery and Data Mining*, 2012.
[8] Y. Zheng and X. Xie, "Learning Travel Recommendations from User-Generated GPS Traces," *ACM Trans. Intelligent Systems and Technology*, vol. 2, no. 1, 2011, p. 2.
[9] H. Su *et al.*, "Crowdplanner: A Crowd-Based Route Recommendation System," *2014 IEEE 30th Int'l. Conf. Data Eng.*, 2014.
[10] A. Rai *et al.*, "Zee: Zeroeffort Crowdsourcing for Indoor Localization," *Proc. 18th Annual Int'l. Conf. Mobile Computing and Networking*, 2012, pp. 293–304.
[11] S. Chen *et al.*, "Crowd Map: Accurate Reconstruction of Indoor Floor Plans from Crowdsourced Sensor-Rich Videos," *2015 IEEE 35th Int'l. Conf. Distrib. Computing Systems*, 2015, pp. 1–10.
[12] M. Alzantot and M. Youssef, "Crowdinside: Automatic Construction of Indoor Floorplans," *Proc. 20th ACM Int'l. Conf. Advances in Geographic Info. Systems*, 2012, pp. 99–108.
[13] Y. Wang *et al.*, "Crowdatlas: Self-Updating Maps for Cloud and Personal Use," *Proc. 11th ACM Annual Int'l. Conf. Mobile Systems, Applications, and Services*, 2013.
[14] H. Yoon *et al.*, "Social Itinerary Recommendation from User-Generated Digital Trails," *Personal and Ubiquitous Computing*, vol. 16, no. 5, 2012, pp. 469–84.
[15] G. Wang *et al.*, "Defending Against Sybil Devices in Crowdsourced Mapping Services," *Proc. 14th ACM Annual Int'l Conf. Mobile Systems, Applications, and Services*, 2016, pp. 179–91.

Biographies

Xiaoyi Fan [S'14] (xiaoyif@cs.sfu.ca) received his B.E. degree from Beijing University of Posts and Telecommunications, China, in 2013 and his M.Sc. degree from Simon Fraser University, British Columbia, Canada, in 2015. He is now a Ph.D. student in the School of Computing Science, Simon Fraser University. His areas of interest are cloud computing, big data, and mobile computing.

Jiangchuan Liu [S'01, M'03, SM'08, F'17] (jcliu@cs.sfu.ca) is a professor in the School of Computing Science, Simon Fraser University, and an NSERC E.W.R. Steacie Memorial Fellow. He was an EMC-Endowed Visiting Chair Professor at Tsinghua University, Beijing, China (2013–2016). From 2003 to 2004, he was an assistant professor at the Chinese University of Hong Kong. He received his B.Eng. degree (cum laude) from Tsinghua University in 1999 and his Ph.D. degree from the Hong Kong University of Science and Technology in 2003, both in computer science. He was a co-recipient of the inaugural Test of Time Paper Award of IEEE INFOCOM (2015), the ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), and the *ACM Multimedia* Best Paper Award (2012).

Zhi Wang [S'10 M'14] (wangzhi@sz.tsinghua.edu.cn) received his B.E. and Ph.D. degrees in computer science in 2008 and 2014 from Tsinghua University. He is currently an assistant professor in the Graduate School at Shenzhen, Tsinghua University. His research areas include online social networks, mobile cloud computing, and multimedia big data. He was a recipient of the China Computer Federation (CCF) Outstanding Doctoral Dissertation Award (2014), the *ACM Multimedia* Best Paper Award (2012), and the MMM Best Student Paper Award (2015).

Yong Jiang (jiangy@sz.tsinghua.edu.cn) is a professor in the Information Sciences Department of the Graduate School at Shenzhen, Tsinghua University. He received his Ph.D. from Tsinghua University in 2002. He was an assistant professor (2002–2005) and associate professor (2005–2013) of information sciences at the Graduate School at Shenzhen, Tsinghua University. His research interests range from computer network architecture to Internet applications, with current foci on future Internet architecture and big data applications.

Xue (Steve) Liu (xueliu@cs.mcgill.ca) is a William Dawson Scholar and associate professor in the School of Computer Science at McGill University. He received his Ph.D. in computer science (with multiple honors) from the University of Illinois at Urbana-Champaign. He has also worked as the Samuel R. Thompson Chaired Associate Professor in the University of Nebraska-Lincoln and as visiting faculty at HP Labs, Palo Alto, California. His research interests are in computer and communication networks, real-time and embedded systems, cyber-physical systems and IoT, green computing, and smart energy technologies. He has published over 200 research papers in major peer-reviewed international journals and conference proceedings in these areas and received several best paper awards. His research has been covered by various news media.