NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SMART SENSING AND MOBILITY ANALYSIS FOR IMPROVED TRANSPORTATION EFFICIENCY

CHU CAO
SUPERVISOR: Dr. MO LI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2019

(On the Spine) SENSING & ANALYSING FOR IMPROVED TRANSPORTATION EFFICIENCY C. CAO 2019

SMART SENSING AND MOBILITY ANALYSIS FOR IMPROVED TRANSPORTATION EFFICIENCY

CHU CAO

CHU CAO

School of Computer Science and Engineering

A dissertation submitted to Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University of Institution.

16 Jan 2019

Date

Chu Cao

Chu Cao

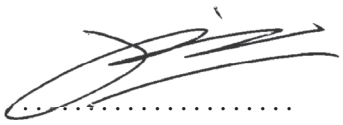
Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data presented honestly and without prejudice.

16 Jan 2019

.....

Date



.....

Li Mo

Authorship Attribution Statement

This thesis contains material from TWO paper(s) published in the following peer-reviewed journal(s) where I was the first and/or corresponding author.

Chapter 3 is published as C. Cao, Z. Li, P. Zhou and M. Li. Amateur: Augmented Reality based Vehicle Navigation System. In Proceedings of the ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT). Vol. 2, Issue 4, Dec. 2018. DOI: 10.1145/3287033

The contributions of the co-authors are as follows:

- A/Prof Mo Li provided the project direction and technical direction. He also improved the writing quality of the paper. Prof Mo Li discussed with me at each stage and each procedure of this paper.
- I prepared the manuscript drafts. The manuscript was revised by Asst Prof Zhenjiang Li and Dr. Pengfei Zhou.
- I designed the study and performed all the laboratory work at SCSE, NTU, Singapore. I also analyzed the data generated during our experiments.
- I proposed the basic solutions and experiment design. With the help and discussion of Dr.Z. Li, Dr. P. Zhou and A/Prof M. Li, we refined the solutions and improved the experiments and evaluations of this paper. The final design of methods and experiments were determined after several rounds of meeting and discussion with Dr. Z. Li, Dr. P. Zhou and Prof M. Li.

Chapter 4 is published as C. Cao, Z. Liu, M. Li, W. Wang and Z. Qin. Walkway Discovery from Large Scale Crowdsensing. In ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). April 2018. DOI:10.1109/IPSN.2018.00009

The contributions of the co-authors are as follows:

- A/Prof Mo Li provided the project direction and technical direction. He also improved the writing quality of the paper. Prof Mo Li discussed with me at each stage and each procedure of this paper.
- I prepared the manuscript drafts. The manuscript was revised by Asst Prof Zhidan Liu, Dr. W. Wang and Dr. Z. Qin.
- The basic solutions and experiment designs were drafted by me. With the help and discussion of Dr. Z. Liu and Prof M. Li, we refined the details of our solutions and improved the experiments and evaluations of this paper. The final design of methods and experiments were determined after several rounds of meeting and discussion with Dr. Z. Liu and Prof M. Li.
- I designed the study and performed all the laboratory work at SCSE, NTU, Singapore. I also analyzed the experimentally collected data.
- Dr. W. Wang and Dr. Z. Qin helped collect data and provided constructive advice during this work.

16 Jan 2019

.....

Date

Chu Cao

.....

Chu Cao

Acknowledgement

First and foremost, I would like to express my sincere appreciation to my PhD supervisor, Dr. Mo Li, for his support, patience and guidance. When joined WANDS as fresh PhD student, I got no experience on doing research and lacked the knowledge base to conduct cutting-edge research. I really want to thank Dr. Li for the career advice which he gives to me and helps me find my real research interests, the research skills that he teaches me and facilitate my capability to carry out influential research and the patience and freedom that he put on me and allow me to establish my knowledge base on mobile computing and urban computing before I can come up with my own ideas to update the state-of-the-art of the specific research domain. Dr. Li is also a nice guy who really cares for his student. I feel lucky to have such a mentor with me during the long journey pursuing my PhD degree.

I am also grateful to my group members in WANDS: Yuanqing Zheng, Zhenjiang Li, Wan Du, Zhidan Liu, Pengfei Zhou, Jiajue Ou, Yuxiao Hou, Shiqi Jiang, Jansen Christian Liando, Yaxiong Xie, Shuyu Shi, Panrong Tong, Xiaoyun Mo, Weiping Sun, Jinlong E, Sijie Ji, Yanbo Zhang, Agustinus Wellson Tengourtius, Amalinda Gamage. I will always remember the days we fight together for paper deadlines, the insightful discussions we had about diverse research topics. In particular, I want to thank Dr. Pengfei Zhou for his generous help. I do learn a lot from our cooperation.

I want to thank my collaborators, Professor Zhengjiang Li and Professor Zhidan Liu, for their help in my research projects. Their insightful opinions would always lead me to the correct direction when I met problem in my research projects. Their valuable comments help to greatly enhance the quality and readability of the writing. It is my great honor to work with those respectful professors.

Last but not least, I would like to thank my family for the support they give me and in particular, I must acknowledge my fiancee, Xiaoyu Miao. It is your love and support that make me get here.

Contents

List of Acronyms

AR	Augmented Reality
GPS	Global Positioning System
GNSSs	Global Navigation Satellite Systems
VLC	Visible Light Communication
CMOS	Complementary Metal Oxide Semiconductor
HMM	Hidden Markov Model
NSE	National Science Experiment
IMU	Inertial Measurement Unit
API	Application Program Interface
DTW	Dynamic Time Warping
FPS	Frames Per Second
MIS	Management Information System
MAC	Media Access Control
RSSI	Received Signal Strength Indicator
OSM	OpenStreeMap
HDB	Housing Development Board
GSV	Google Street View
LED	Light-Emitting Diode
CDR	Call Detail Records

List of Figures

List of Tables

Abstract

Sensors proliferate in human daily life and has changed the lifestyle of human beings. As the connection between human beings and digital devices, sensors not only bring new "ability" to users, such as localization, but also benefit users, such as activity monitoring. Therefore, one important research direction is to fully utilize the abilities of sensors built-in digital devices to provide further help for users. Many emerging applications leverage the sensors embedded in smart phones to provide service for users, *e.g.,* indoor navigation, indoor/outdoor detection, visible light communication. All these applications rely on the powerful built-in sensors on smart phones. Analyzing the data generated by sensors during the daily usage of human beings can also reveal the knowledge hidden behind, *e.g.,* tracking the daily activity to evaluate the health of users. Data collected from crowdsensing bring new opportunity to discover new knowledge that cannot be revealed using a small set of data.

We observe that current navigation systems show digital maps rather than real world scene to users. Thus we present Amateur, an augmented reality based vehicle navigation system using commodity smart phones. Amateur reads the navigation information from a digital map, matches it into live road condition video captured by smart phone, and directly annotates the navigation instructions on the video stream. The Amateur design entails two major challenges, including the lane identification and the intersection inference so as to correctly annotate navigation instructions for lane-changing and intersection-turning. In this project, we propose a particle filter based design, assisted by inertial motion sensors and lane markers, to tolerate incomplete and even erroneous detection of road conditions. We further leverage traffic lights as land markers to estimate the position of each intersection to accurately annotate the navigation instructions. We develop a prototype system on Android mobile phones and test our system in a total number of more than 300 *km* travel distance on different taxi cabs in Singapore. The evaluation results suggest that our system can timely provide correct instructions to navigate drivers. Our system can identify lanes in 2s with 92.7% accuracy and detect traffic lights with

95.29% accuracy. Overall, the accuracy of the navigation signs placement is less than 105 pixels on the screen throughout the experiments. The feedback from 50 taxi drivers indicates that Amateur provides an improved experience compared to traditional navigation systems.

When providing the navigation services to drivers, digital map plays a key role in navigation. The completeness of maps is significantly important. Most digital maps are designed for vehicles and miss a great number of walkways that can facilitate people's daily mobility as pedestrians. Despite of such a fact, most existing map updating approaches only focus on the motorways. To fill this gap, we present VitalAlley, a walkway discovery and verification framework with mobility data from large scale crowdsensing. VitalAlley aims to identify the uncharted walkways from the big but noisy personal mobility data and incorporate these findings into existing incomplete road maps. The implementation of VitalAlley faces the major challenges due to the unstructured nature of the walkways themselves and the noise from crowdsensing data. VitalAlley leverages different aspects of individual mobility to model and estimate the walkable areas, based on which representative walkways that connect known road segments or points of interest are extracted. To verify the new-found walkways, we further propose image based auto-verification with the help of publicly accessible street image database from GSV. VitalAlley is implemented and evaluated with real world crowdsensing data from the Singapore National Science Experiment. As a result, 736 walkways (totaling 161 km in distance) are identified from the mobility dataset collected from 108,337 students in Singapore. We manually verify 224 walkways totaling 32.4 km over a 9 km^2 district through on-site inspection. The results suggest over 96% accuracy of VitalAlley in discovering the walkways.

Chapter 1

Introduction

The emergence of sensors has changed and facilitated our daily life. Many kinds of sensors make our life more convenient. Mobile devices with rich built-in sensors, such as smartphones, tablets, and smart wearable-devices are becoming more and more popular in the world. One reason is that those mobile devices could bring new "abilities" to users, such as localization function using GPS sensor, playing a key role in daily life especially when users travel to new places. Combined with the digital maps and route planning algorithms, drivers can drive to wherever they are interested. For instance, with the help of sensors, anyone can be a taxi driver even she/he does not know the road very well as long as s/he has a driving license. More smartphone built-in sensors can be leveraged, such as accelerometer, camera sensor and GPS sensor. During my PhD career, I focus on the improvement of navigation service which is a key component in transportation system by exploiting a broad range of available sensors.

Modern navigation system consists of three main components: positioning, mapping and route planning. Take Google Maps Application [37] as an example. It is one of the popular navigation systems used by human beings living in modern cities. Google Maps can be installed on mobile phones and requires the permission to access the built-in sensors. In Google Maps, mapping is provided by Google as they have their own digital maps and basic route planning algorithms. The navigation system has to get the permission to access the GPS sensor in order to realize the positioning function for navigation systems. It is possible for researchers to improve the performance of navigation systems by enhancing the components individually or jointly. In this thesis, we first try to provide our solutions to improve the whole performance and experience of navigation systems by involving AR and to complete uncharted information

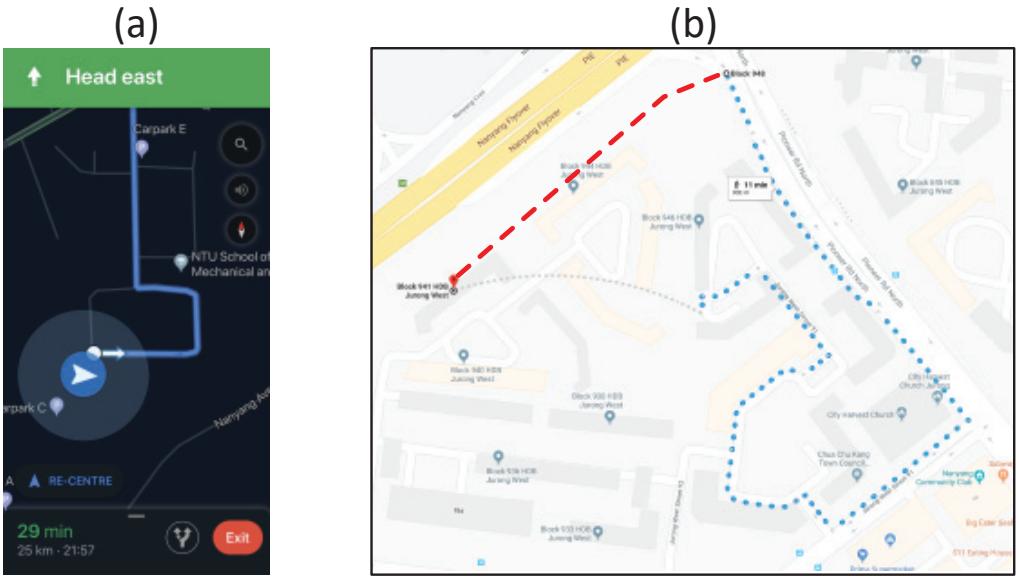


Figure 1.1: (a) User interface of one conventional navigation system during usage. (b) Comparison between recommended route from Google Maps when querying the route from Pioneer Roads North to BLK 941 and existed shortcut.

for existing digital maps by involving crowdsensing. We aim to provide a more user-friendly navigation service based on current navigation framework. The screen shot of one of the current navigation systems is shown in Figure 1.1 (a). It clearly illustrates that digital map is the underlying assisted map when users are using the navigation service. Users have to transfer the information shown on digital map to match with the real environment in front of vehicles, which may lead to safety issues especially for driver novices. We involve AR into navigation system and leverage video streaming to make underlying map and the real world identical. Thus, the information transferring procedure can be eliminated. Driver users of our improved navigation system can get what they see directly from the screen.

During the usage of navigation systems, we find that the digital map has uncharted routes especially walkways for pedestrians. In Figure 1.1 (b), Google Map recommends a route which is shown as blue dots when querying possible walking route plans from Pioneer North Road to BLK 941 in Singapore. This recommended route will take pedestrians about 11 minutes but there is a shortcut shown in red dashed line in Figure 1.1 (b). Such a shortcut walkway only takes 2 minutes to travel from Pioneer North Road to BLK 941. One observation is that pedestrians living nearly are quite familiar with those shortcuts while others are not. We try to

involve the power of crowdsensing to discover those uncharted walkways for pedestrians and complement those newly-found walkways to existing digital maps.

In my PhD career, my research includes how to improve the traffic efficiency on the aspect of navigation. Both vehicles and pedestrians use navigation service in daily life but they are used in different situations. We try to improve the navigation service for both vehicles and pedestrians. In project Amateur, we integrate augmented reality into the vehicle navigation service to provide a more user-friendly service for drivers. Drivers could easily get the instructions shown on smartphones. It is a basic philosophy regarding of what you see is what you get and what you need to do in our design. In project VitalAlley, we leverage the power of crowdsensing and the commutative dataset to discover walkways for pedestrians to provide a more efficient and more cost-effective walking routes. Walkers are provided with shorter routes and more walkable routes in the maps added with our new-found walkways.

1.1 Integrated Augmented Reality for Improved Vehicle Navigation

Navigation service on vehicles or cars can be provided by vehicle-mounted navigators or smart phones [18]. They display vehicle's instant GPS location and overlay the navigation information onto a digital map to assist the driving. The major limitation of existing navigation systems is that the navigation information is displayed on the digital map, while the driver's focus is on the traffic condition through the front window — the navigation information is displayed in an indirect and inconvenient way to the driver. The driver thus has to understand the navigation instructions presented on the device, and then map them to the front traffic view observed. In many complicated road conditions, *e.g.,* multiple entrances of nearby intersections in front, drivers may take a wrong turn due to the misunderstanding of displayed instructions or navigation errors. Concentration on translating the digital map and navigation instructions to the real world view may also cause potential safety issues, especially for the unfamiliar road conditions, long-term driving and novice drivers.

The major motivation of this project aims to improve the navigation service. To overcome the above limitations, we propose Amateur, *i.e.,* **A**ug**m**ented re**a**li**t**y (AR) [19] based v**e**hic**u**la**r** navigation system, with the help of nowadays smart phones. Instead of displaying instructions on a digital map, the smart phone directly displays the front traffic conditions, where all nav-

igation instructions are annotated as arrows on the live video stream to guide the driver, *e.g.,* to keep going straight, to change to the right/left lane, to turn right/left at an intersection, *etc.* Because the live video display is identical to the driver's front window view and navigation annotations are clearly marked on the video, such an augmented reality design minimizes the display-understanding gap for drivers, and enhances the navigation experience. To precisely place navigation instructions into the live video, the system, however, needs to continuously track the lane-level and navigation status at a high granularity to ensure navigation accuracy, *e.g.,* unlikely missing a turn or making a wrong turn, as well as to improve the navigation reliability with GPS errors. In particular, developing such a system in practice entails the following challenges:

1) Amateur needs to precisely identify the host lane, *i.e.,* the lane vehicle currently stays at. GPS cannot reliably locate vehicles at a lane level, (*e.g.,* 3 to 4 meters), due to high localization errors especially in urban areas. In the literature, some existing works have studied the host lane detection problem, but they mainly leverage extra sensors. For instance, the authors in [21] utilize the extra module to communicate with other vehicles. Jiang *et al.* [49] deploy hundreds of sensors on the vehicle to fulfill the design. High-definition map and dead-reckoning sensors are required to achieve lane-level map matching in [95]. In computer vision, there are also existing efforts made for the lane detection for the automatic driving. In [4], image processing techniques are used to detect the lane markers and roads. A comprehensive computer vision based algorithms are summarized in [42]. These solutions extract the complete lane information from video frames, but they rely on non-trivial deployment of spinning cameras or camera arrays around the vehicle, and incur high computation overhead which is not affordable by mobile devices directly.

2) To place navigation instructions for the turn at an intersection, the accurate position of the intersection needs to be identified in video frames. A straightforward solution is to use GPS locations of the vehicle to estimate the distance to the intersection, and then translate it to a proper position in the video frame. Such an approach is not a good choice due to GPS errors. Amateur tries to leverage traffic lights that are normally nearby intersections as indicators to place navigation instructions. Although some traffic light detection designs exist, they mainly rely on expensive computer vision techniques, which cannot be afforded by smart phones either, *e.g.,* the design in [33] applying machine learning algorithms, the approaches in [24] and [71] adopting expensive image processing techniques for detecting the traffic light's

shape and edge respectively, *etc.*

To address above issues in Amateur, we make the following contributions.

i) We propose Amateur, an augmented reality based vehicle navigation system, which directly displays the navigation instructions on the live road condition video captured by the smart phone. Compared with traditional methods that mainly display the navigation information on a digital map, Amateur could provide a more convenient and user-friendly navigation service.

ii) We propose effective techniques to address the lane identification and intersection inference challenges. For the first challenge, our key insight is that identifying which lane the vehicle currently stays on is sufficient for the Amateur design, so that the original lane detection problem can be simplified to the host lane identification problem. For the second challenge, we leverage traffic lights and a pin-hole model to estimate the position of the intersection for correctly placing the navigation instructions in the video.

iii) With above designs, Amateur is lightweight enough to smoothly execute on the mobile platforms, and we implement a prototype on Nexus 5X. We comprehensively evaluate its performance over road segments totaling over a 300km distance and also conduct a user study on 50 different taxi cabs in Singapore. The results demonstrate the efficacy of the Amateur design, which can improve the navigation service from four important aspects ease of use, perceived distraction, navigation experience and user-friendliness.

1.2 Walkway Discovery from Crowdsensing for Improved Pedestrian Navigation

Digital road maps are of significant importance for route planning and navigation in our daily life. Existing road maps, however, are mostly vehicle oriented and do not contain the information of many walkways that local pedestrians usually travel with. Many areas or walkways (*e.g.,* basement of buildings, interior of shopping malls, open fields, *etc.*) are used as shortcuts by people who are familiar with the local area but are not included in the digital road maps. These walkable areas or paths, although very useful, are uncharted on the maps and thus cannot be made of use by the public [17].

In this project, we apply the idea of crowdsensing [65, 121] and make use of the mobility data from a large number of local students to discover the uncharted walkways. Our study relies

on an ongoing crowdsensing project - Singapore National Science Experiment (NSE) [69] - a city wide initiative that enrolls more than 250,000 local students carrying smart devices that sense their surrounding environment and track their mobility [103] on daily basis. The smart devices periodically log and upload the locations, IMU readings, and other environmental parameters from participating students. Massive mobility data are collected from the students across the entire city, which provides us the opportunity to study and discover the walkways. The rationale of this study is that the local students are active users of the walkable areas or paths in their neighborhood, and by following their footprints we will be able to discover and summarize those walkways.

There have been existing efforts [61, 85, 98, 107] made to completing the digital road maps. Most of such works, however, only mainly focus on discovering vehicle-oriented motorways rather than the walkways for pedestrians. The motorways are structured where vehicles strictly follow the lanes and directions with almost 1-dimensional uncertainty in mobility. On the other hand, the walkable areas or paths are mostly unstructured where people travel with high freedom of 2-dimensional uncertainty (*e.g.,* lawns, shopping malls, open fields, *etc.*). Most previous works for motorway discovery leverage the structured property of motorways to infer missing road segments with the assumption that vehicle trajectories are constrained on the 1-dimensional roads. Techniques like trace clustering [98, 107] and location point clustering [61, 85] are directly applied with the GPS trajectories of vehicles. Due to the unstructured and pedestrian-oriented nature of walkways, neither previous map updating methods nor vehicle trajectory data can be applied to discover the walkways.

In this project, we present VitalAlley that uses the crowdsensing mobility data from NSE for walkway discovery. Unlike previous works for map completion, VitalAlley faces special challenges arising from the unstructured nature of walkways as well as the imperfect quality of the crowdsensing data. The location reports from participating students may scatter over a wide area of freedom, making it difficult to estimate walkways through the area. The imperfect quality of the crowdsensing data makes the situation more complicated - the smart device takes the locations every 15 seconds (or even longer in certain circumstances) with errors that range from tens to hundreds of meters (with WiFi hotspot based localization).

VitalAlley statistically looks at the big mobility data from many aspects, being able to tolerate errors and noise contained in the mobility data. With both the location reports and step counts derived from IMU readings, VitalAlley builds an ellipse model to estimate the probabil-

ity of how the individual walks between consecutive reported locations. Putting together such micro estimations from all students allows VitalAlley to statistically understand how likely different areas are walkable. With such knowledges, VitalAlley applies a two-phase clustering method to discover how the potential walkable areas are connected with nearby known road segments or points of interest and then identify walkways that are representative for people who walk through such areas.

VitalAlley further employs an auto-verification method to verify the correctness of the new-found walkways. By invoking the online Google Street View (GSV) APIs [35], VitalAlley is able to access an extensive image library containing street images from most road segments. VitalAlley retrieves the GSV images from where the discovered walkways join existing roads or points of interest and analyzes the key image descriptors with reference to a library of templates to verify whether the identified walkways are true or not.

To the best of our knowledge, this is the first study for digital map completion with a focus on walkways, which includes the ellipse based walkable area estimation and weighting, representative walkway identification, and GSV image based auto-verification. The systematic study with city scale mobility data from crowdsensing is of the largest scale. We extensively evaluate the performance of VitalAlley with the mobility data collected from 108,337 students in Singapore, which lasts 11 weeks. Based on the analysis of more than 400 million mobility data records, we discover 736 walkways (totaling 161 km in distance) and verify 224 walkways totaling 32.4 km through on-site inspection. Those verified walkways can be integrated into the digital map of Singapore from OpenStreetMaps [73]. The results show that VitalAlley has a 96% accuracy in discovering the walkways.

1.3 Organization of This Thesis

This thesis has five chapters. The literature review is presented in Chapter 2. We introduce a system named "Amateur" that involves AR into conventional navigation systems to provide a remarkably improved user-friendly experience in Chapter 3. In Chapter 4, we introduce a walkway discovery and auto-verification system that involves the power of crowdsensing to construct the uncharted representative walkways for pedestrians. Last in Chapter 5, we present the conclusion of this thesis and imagine our future work.

Chapter 2

Related Work

We review the related works of our projects. The related works are presented in NINE parts: vehicle navigation, AR-based navigation, lane identification, traffic light detection, digital map inference, digital map updating, map matching, geo-location data processing and crowdsensing.

2.1 Vehicle Navigation

In general, majority of the vehicle navigation systems match the information from a GPS receiver with a digital map. The most likely position of the vehicle is estimated. In urban environment, the satellite signals are blocked by skyscrapers, leading to a reduction of position estimation. Adding more sensors to the GNSS receiver is a feasible direction of giving navigation system higher accuracy [87]. The vehicle sensors provide more information such as acceleration, roll and pitch, depending on which types of sensors are used. Typical update rate of GNSSs receiver is less than 20 Hertz [1], while the sampling rate of modern low-cost accelerometer and gyroscopes is hundreds of Hertz. Thereby, the fusion of information from all sensors provides a better position estimation with a better accuracy for vehicles. In reality, the performance of navigation systems depends not only on the characteristics of the sensors, GPS receiver, vehicle model and map information but also on the trajectory dynamics and surrounding environment.

Many commercial navigation softwares have been developed for the smart phones and the most representative example is the Google Maps [37]. Although many of these products now

can support the "audio message reporting" function, drivers still need to manually understand the navigation information. The authors in [53] find that the acoustic information is preferred by the driver compared with the text and map. However, the research in [79] find that the long-time audio messages may significantly distract drivers' attention, causing the potential safety issues, especially with long auditory messages [28]. The automobile manufactures propose their own AI-based navigation, such as TESLA [94], but they have powerful sensors embedded, providing the capability of sensing nearby environment. Those sensed information are accumulated for the "brain" to make decisions in "auto-pilot" mode. Many companies give their conceptual products, like Navion [99], Hudify [43] and Exploride [26], while they do not have available devices currently.

2.2 AR-based Vehicle Navigation

Augmented Reality was proposed by Tom Caudell and David Mizell from Boeing to help workers assemble wires and cable for an aircraft [19]. In AR, 3D virtual objects are integrated into the real world [5]. AR is changing how humans interact with the digital contents [12]. So far, the AR technique has been used in many fields, such as game [76], medicine [57], education [67], user interface [83], *etc.* [82] proposed a method to accelerate the rendering process in a head-mounted-display (HMD). Researchers also employ AR-based indoor tour system on smart phones [46]. Researchers in [75] developed a simulated AR environment to study how virtual navigation cues help forklift operators locate pallets.

In the literature, AR [19] serves as a promising technique to advance the vehicle navigation design. AR is a technology that overlays the digital information on objects or places in the real world for the purpose of enhancing the user experience [8]. The existing attempts to utilize AR in the vehicle navigation designs have been made in both academia and industry.

The study in [29] is a pioneer work to introduce the visualization technique to the vehicle navigation designs, in which the authors compare three navigation modes (audio only, audio plus map, and visualization) on a simulated platform and find that drivers watch less frequently on the navigation screen with the visualization. The authors in [30] further examine an AR based design on a simulated platform and achieve further improved user experience. The authors in [50] report a user study also in a simulated environment to compare the driver's behaviors in different environments. These prior studies suggest that using AR is a more effective

and safer way to design a navigation system [81]. Following this trend, the authors in [112] build an in-vehicle navigation system using the AR idea. Vehicular AR is further proposed in [27] using a special camera. However, both of them cannot provide the lane-level navigation guidance. On the other hand, an LED array is introduced on the windshield of vehicles to provide affordable AR instructions [74], which again requires dedicated hardware to support. Some researchers combine the real world video with expensive computer graphical models to provide AR navigation information [82]. Different from these works, we propose a mobile-affordable AR navigation system, without installing dedicated hardware on the car, to provide a fine-grained lane-level navigation service.

There are also existing attempts made from the industry. Sygic [91] overlays the navigation information on a real or virtual environment, without the actual navigation provided. Other products like Navdy try to integrate incoming calls, messages and navigation together and project those information onto the windshield. Such AR services can only supplement the navigation design. On May 9 2018, Google started to provide the AR function for their own map application on the Google I/O [36] event in the latest Android system. The captured scenes by camera can be used to match the Google Street View so as to localize the user and provide proper navigation instructions. While with such a design, it still cannot provide the lane-level instructions due to the localization error. Meanwhile, if the driver's Google Street View is not the latest version, the matching error can be large or even the matching failure occurs. As far as we know, Amateur is not the first work that introduces AR in the navigation system design but it is the first system realizing this vision on commodity mobile devices by addressing unsolved design challenges.

2.3 Lane Identification

Many approaches in the literature are proposed to extract lanes from the video streaming for automatic driving designs [42], which can be used to avoid the lane weaving or drifting [116], or detect whether a remote object can potentially collide with the vehicle. As these existing works aim to achieve a complete lane search, labeling entire lanes on the frame [93] by leveraging the lane edge detection [16] involves excessive computations [22]. Generally, those methods take each frame in video streaming as input. Every RGB frame is converted into gray scale frame, after which an edge detection algorithm is used to identify high frequency components

in the gray image. All the frames need to be transferred into bird-eye view to eliminate the perspective effect from camera sensors, where the orientation information of camera should be known. Thus, the corner and edge are detected as possible straight lines or corners of lane markers. Hoffman transformation is used to find the points that are likely on same straight line. The straight lines are identified by connecting those points, which are the straight edges of lane markers in reality. The whole image processing progress involves too many computer vision algorithms, causing too much computing overhead. So those methods cannot be afforded by the mobile devices. Although the authors in [51] propose a down-sampling technique to reduce the number of frames to be processed, the computation overhead of image processing is still beyond the capacity of mobile devices. Different from these existing works above, we focus on the host lane identification problem in the Amateur design and explicitly address the incomplete and erroneous input issue, which is not been solved yet.

2.4 Light Detection

Traffic light detection is a key technique in automatic driving designs, which can be grouped into three categories [24], namely, color segmentation, shape detection and direct classifier. The former two types of techniques involve extensive computer vision computations. The study in [71] proposes to identify the traffic lights by detecting the edge of a circle and also the colors of the lights. Authors in [92] determine the position of traffic lights in the video by their colors and the central point. The extensive computer vision computations make these methods not suitable for VitalAlley. The direct classifier technique, on the other hand, requires a large volume of training data [54]. The researchers in [33] apply the machine learning algorithm to detect traffic lights. Such a design is also computationally expensive and requires a training phase. To avoid both the high computation complexity and training overhead, we observe that traffic lights consist of LED bulbs, which are powered by alternating current. We thus propose to utilize the flickering feature of LED bulbs to detect the position of traffic light on the frame, and further leverage the pin-hole model to estimate the position of the intersection for correctly placing navigation instructions in the video. The idea was inspired by the researchers in VLC field, where they leverage the flickering feature of LED bulbs to transfer information. Before the transferring, they use different frequencies to drive the LED bulbs, resulting in different flickering frequencies that can carry diverse information in modulated phase. Receivers (cam-

era sensors) can demodulate the signal by identifying the frequency due to the rolling shutter effect of CMOS sensors. Although there are also some existing efforts made to identify visible lights on images, *e.g.,* [56], they strive to the indoor positioning designs, which requires an off-line processing from servers to achieve a high-precision light location identification. Many cities, such as Singapore, Shenzhen have the LED traffic lights and CMOS cameras dominate the smart phone market currently, leveraging the flickering feature of LED bulbs to detect traffic lights on smart phones is a feasible and low-cost solution.

2.5 Digital Map Inference

Map inference aims to automatically generate the whole map from the satellite images or location trajectory data. The aerial imagery methods [84, 122] employ image processing techniques to draw only the main roads due to the limit of image resolutions, and thus cannot be used for discovering small trails like the walkways. Aerial imagery methods are labor-intensive and low efficiency. There are three categories of methods proposed for the GPS trajectory data based map inference, *i.e.,* K-means [2, 105], Kernel Density Estimation [11, 90], trace merging and clustering [52, 64]. The basic idea is to treat one piece of data (containing multiple geo-location observations) as a high-dimensional point and cluster those points in high-dimensional space. The distance measurement of different high-dimensional points may be diverse. Authors of [85] proposed to leverage the Hausdorff distance, which can tolerant error to some extent. Due to the definition of Hausdorff distance, however, it does not perform well when there are identical points in the two trajectories. On the other side, to achieve the results for different applications, most of these methods, however, build on various impractical assumptions of the GPS data, including low noise and high sampling frequency (*e.g.,* 1 Hertz). They perform poorly once the assumptions are not hold [10, 64]. In contrast, our method preserves no assumption on the input data, and is more robust against noisy data like NSE mobility data. Our method is fed with consecutive location observations and also general enough for all geo-location datasets.

2.6 Digital Map Updating

Map updating aims at completing a given road map by updating missing roads from geo-location data. Several recent works, *i.e., CrowdAtlas* [98], *COBWEB* [85], and *GLUE* [107], have been proposed to find missing roads, which are in particular the well-structured motorways, based on GPS data collected from vehicles. Those GPS datasets are generated by vehicles whose moving trajectories are constrained by motorways. These works extract new-found roads mainly relying on trajectory clustering [98] or location point clustering [85, 107] coupled with some well-tuned thresholds, which thus make them be prone to failure when involving noisy data. Authors in [61] consider the moving direction of vehicles to identify different moving directions on the same route segment. The direction information is derived from raw dataset containing locations and time information. In this paper, we consider to complete a given road map by updating the missing walkways from personal mobility data, which are noisy and random in nature. The unstructured and pedestrian-oriented characteristics of walkways implicitly make the problem more difficult and essentially distinguish our work from the existing map updating works. Walkers are moving in a two-dimension space compared to vehicles moving in one-dimension following the curvature of motorways. To the best of our knowledge, this is the first work to discover and update walkways from noisy mobility data at large scale.

2.7 Map Matching

Map matching is a technique to match location data to existing road networks that minimizes the influences of localization errors. There are plenty number of research papers on matching geo-location observations to a digital map. Existing map matching methods can be classified into three categories: local/incremental methods [20, 39, 100, 101], global methods [3, 14, 70, 115] and statistical methods [44, 68, 77]. The local/incremental methods find the local matching of geometries (candidate edges) in terms of similarity measurement between location observations and edges. The global methods aim to match the whole trajectory with the road networks, minimizing the distance between the trajectory and the matched road segments. The authors in [77] propose to leverage Bayesian classifier and HMM to model topological constraints of the connection between road segments. An enhanced method based on extended Kalman filter and cubic spline interpolations is proposed [44]. In order to compensate for noise

and gaps of individual trajectory, leveraging a collection of trajectories to do map matching has been proposed in [62], which increases the density of trajectories, leading to a better matching results, which is similar to [98]. Since the third one globally considers all location observations and achieves the best accuracy [89], in this project we choose one representative algorithm in [68] to perform map matching on the mobility data. We may adapt some of those works in our paper to construct map matching phase. On the other hand, these works are parallel with our work and can benefit from a full road map completed by our system.

2.8 Geo-location Data Processing

Geo-location data contains the location information of contributors, which is vitally valuable information. Many researchers try to discover the hidden information behind the data leveraging geo-location dataset. The data processing is usually target-oriented. Removing outliers is a frequent target in data pre-processing. Taking the GPS data as an example, vehicles generating the data should move at a reasonable speed, based on which one could remove the outliers that are very far away from the reasonable range. [61] removes the outliers that are outside a predefined threshold. Authors in [62] propose to concatenate trajectories from different vehicles to form a more representative trajectories in order to figure out the frequently used and most possible routes. Processing methods are differently designed in various situations. We also define our own data pre-processing methods in VitalAlley project. Noisy data and sojourn data are removed in our processing. The main reason is that those data may have negative effect on our walkway discovery, leading to wrongly identified walkways areas which may result in incorrect walkways. In our NSE dataset, sojourn data are generated when users are playing or staying at some specific regions.

2.9 Crowdsensing

Crowdsensing, sometimes refereed as mobile crowdsensing, is a technique where a large group of individuals having mobile devices capable of sensing and computing, such as smartphones, tablet computers, wearables, collectively share data and extract information to measure, map, analyze, estimate or infer any processes of common interest [102]. Authors in [31] have well defined the idea of scrowdsensing. With the help of crowdsensing, the large project may be di-

vided into small tasks for different individuals. A famous project built based on crowdsensing is called OpenStreeMap [73], which is an open-sourced digital map made by millions of contributors. The volunteers around the world contribute only part of the map in their residential area. OSM combine all the data and form a world wide digital map.

How to motivate people to take part in a crowdsensing project has been discussed in [113] where authors leverage game theory to analyze the relation of human behaviors and the proposition of auction-based approaches for incentives. Koutsopoulos [55] inspired by game theory and proposed the calculation of optimal payment for participants that minimizes the total cost of compensation. However, compensations are not always available. For some specific research applications [41] of crowdsensing rely on volunteers helping in the projects with non-financial incentives. Many games are designed for participants to motivate their activities such as [45], or the service quality can be improved when users provide the required data, such as a more accurate navigation service [78].

When users are willing to contribute in a crowdsensing project, a energy-efficient scheme would be better for users such that they have little effect on their normal utilization of tools, such as mobile phones. This energy awareness problem has been discussed in [58]. Authors proposed a piggyback mechanism to collect data when applications invoke the sensors. This approach significantly reduces the usage of energy but the number of measurements is too small to track the mobility. An energy-efficient location API in [25] predicts the location assuming that the paths are repeatedly took every day. This decreases the accuracy and involves large errors when users do not follow the routine paths. A method proposed in [72] allows to enable GPS for localization based on accelerometers and could achieve a 27% energy saving. In a series of works [97, 108–110, 120], authors discuss the energy usage when transferring data in mobile crowdsensing system. There are methods optimizing participant selection in crowdsensing projects so that the overall work is minimized while the coverage of geographic area of interest is guaranteed. The authors also proposed to minimize the number of tasks allocated to users. Authors in [111] propose to minimize the energy consumption by selecting which portion of users should enable sensing tasks. Kyon-mo *et al.* [114] develop a context-aware system leveraging Bayesian networks to lower the power consuming.

Researchers also leverage the power of crowdsensing to solve problems. In [65, 121], authors leverage the crowdsensed data from the mobile phones of commuters on different buses to provide a service estimating the bus arrival time. The mobility measurement involving crowd-

sensing is proposed in [40], where reposting, tagging and sharing public information are used in the FlierMeet project. In our project VitalAlley, the dataset are from mobile crowdsensing. Students in NSE project wear a sensor named SENSg to sense the surrounding environment.

Chapter 3

Augmented Reality based Vehicle Navigation[1]

3.1 System Design

3.1.1 Design Overview

Fig. 3.1 illustrates the system architecture of Amateur. Supposing Alice plans to drive from position A to position B, using Amateur. After Alice inputs the origin and the destination, Amateur first launches the underlying navigation services, *e.g.,* Google Maps [38] or OpenStreetMap [73], to obtain detailed lane-level[2] navigation information from A to B. Note that Amateur is not positioned to redesign the entire navigation stack. Instead, it reuses the existing navigation functions and focuses on automatically matching the navigation information to the live road condition video to achieve a much augmented user experience.

When Alice's car is moving, Amateur periodically reads GPS to acquire a rough location of the car for determining the road segment the car is currently on. According to the navigation plan provided by the underlying navigation service, Amateur then knows which lane should be taken and how many lanes on current road in total. On the other hand, Amateur executes the *Lane Identification* module in Fig. 3.1 to determine the host lane. If the host lane differs from the lane suggested by the navigation service, the *Instructional Sign Placement* module

[1]This Chapter is partially published on IMWUT 2018, presented on UbiComp 2019 [17].

[2]The lane-level road information is already available for most cities in commercial navigation products, like Google Maps [38]. However, due to the limited resolution of GPS positioning, drivers cannot directly leverage such detailed information in existing systems throughout the navigation procedure.

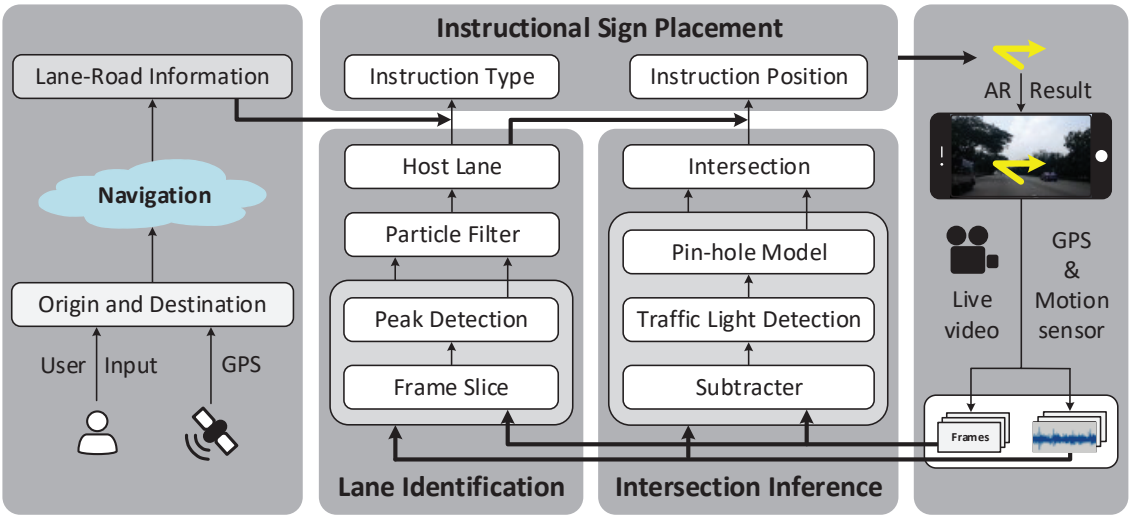


Figure 3.1: System architecture of the Amateur design.

displays an annotation arrow on the live road condition video to inform drivers changing to the correct lane. To fulfill above design, video frames first undergo a slicing operation to extract the interested points from each frame. Amateur then executes peak detection to obtain all lane markers (boundaries) on the road. The observed lane markers however can be incomplete or mixed with detection errors. The lane detection module further adopts a particle filter to statistically infer the most likely position of host lane through consecutive observations, by matching the observed lane marker patterns and accelerometer readings with correspondingly expected distribution.

When Alice needs to take a turn at one intersection, Amateur can specify the intersection's position in the video stream for displaying the corresponding turning instruction. To this end, the *Intersection Inference* module first reads the GPS data. If the car gets close to the intersection, *e.g.,* 50 meters in front, it conducts the frame subtraction operation to unveil the flicking feature of traffic lights to identify the position of traffic lights in video frames, based on which the module further applies a pin-hole model to estimate the position of the intersection, so that the turning instruction arrow can be correctly placed.

With above visualized navigation design, Alice could simply follow the displayed annotations to reach the destination. We detail the design of lane identification and intersection interference modules in Amateur.

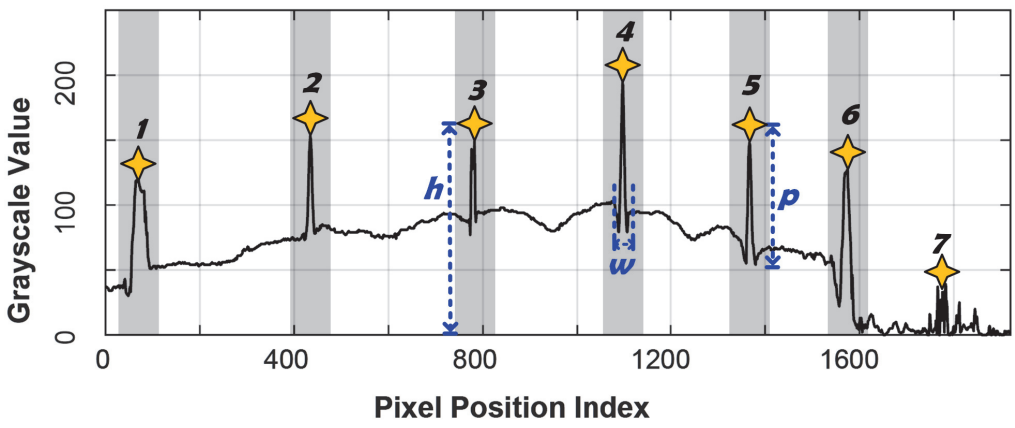


Figure 3.2: Gray scale values of one row of pixels.

3.1.2 Lane Identification

To determine the lane markers on the road, we have the following key observation: as lane markers are usually painted with bright colors, *e.g.,* white and yellow, after we transform the video frames to the gray scale, the pixels on the lane markers will exhibit large pixel values, *i.e.,* peaks. Fig. 3.2 illustrates the gray scale values of the pixels along a straight horizontal line extracted from the frame in Fig. 3.3(a). After manually checking the ground truth of the lane marker positions on the frame, we find that each highlighted peak in Fig. 3.2 indeed corresponds to the boundaries of all the lanes.

Although we propose a design, in Section 18, to precisely quantify and extract each individual peak from the converted gray scale pixel values, the obtained overall peak pattern is not always reliable due to two types of unavoidable errors:

- False alarm peaks: bright reflections, paintings and objects from the road surface can cause the detection of additional peaks, not corresponding the lane boundaries.
- Missing peaks: certain lane boundaries may also be miss-detected due to light conditions or blocking by nearby cars.

If we directly use such noisy or erroneous peak patterns, host lane cannot be reliably detected. In Amateur, we introduce a particle filter based host lane identification design to tackle this issue.

Particle filter based host lane identification

The core idea for the host lane identification, *i.e.,* identifying the lane number of host lane on the road, is to conduct a template matching based on the observed gray scale value peaks as in Fig. 3.2.

To facilitate the discussion, we first introduce necessary notations used in the design. Fig. 3.3(a) depicts the live video from the smart phone and we convert the pixels from the highlighted narrow stripe to the gray scale values for the peak detection (detailed in Section 18). For all the detected peaks from one frame, we find that the two peaks that are closest to the center line of the video frame are usually from the two markers of the host lane (yet the lane number is still unknown), denoted as X_{mid}. This is because smart phone is normally placed close to the middle of the car. For the example in Fig. 3.3(a), the host lane is lane-3 (with respect to the left). The two markers of this lane in Fig. 3.3(b) indeed enclose the frame's center line. On the other hand, because drivers need to keep a distance to the car in front, we find that the two markers of the host lane can reliably be detected.

As a result, we can place an x-axis along the narrow stripe as in Fig. 3.3(b), where the origin is the center of these two lane markers, denoted as $[0]$. In addition, we can treat these two lane markers as a delimiter. The locations of other peaks detected on the left (and right) hand side of this delimiter, denoted as X_{obs}, can serve as an observation to infer the host lane's location. In Fig. 3.3(b), X_{obs} contains two peaks, whose coordinates along x-axis are -895 and -526 respectively, on the left and other two, 407 and 614, on the right. Viewing X_{obs} as the constraints, we can then guess the location of the host lane with the following design.

Particle filter framework. Supposing the current road segment has 4 lanes with 5 lane markers (obtained from the underlying digital map) illustrated in Fig. 3.4(a). The host lane, *e.g.,* surrounded by the middle two lanes in X_{mid}, is one of these four. If the host lane's lane number is one, the expected positions of other three lane markers or peaks are depicted as X_1 in Fig. 3.4(b), and we call it a template[3]. Similarly, we can generate different templates for other three possible cases. In addition to the four templates, we also have a real observation $X_{obs} = [-895, -526, 407, 614]$ with one extra peak detected, as shown in Fig. 3.4(c). We thus want to know which template in Fig. 3.4(b) that X_{obs} matches best.

[3]We adopt a default width, *e.g.,* 316 pixels, between two consecutive peaks in each template. As all templates adopt the same width, even it is different from the actual width on the current road, it does not affect the matching result.

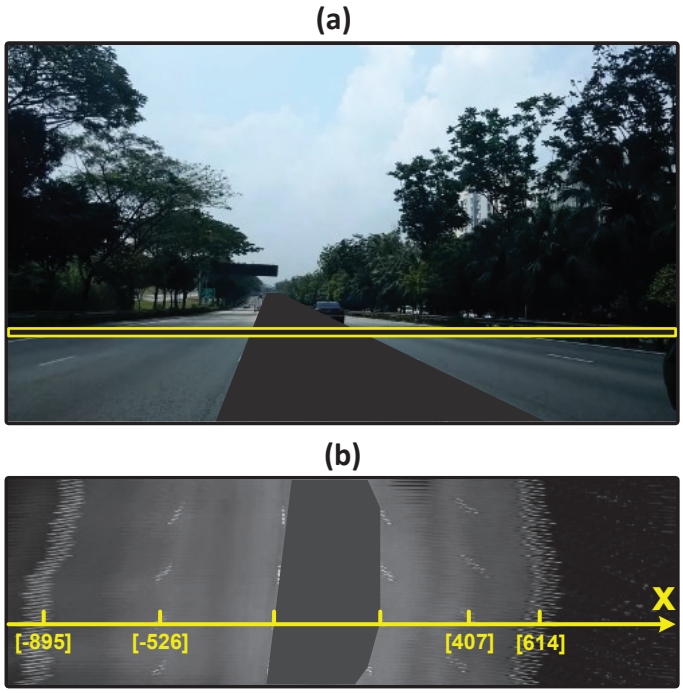


Figure 3.3: Illustration of peak detections from video frames. (a) Selected horizontal pixels at the same position from one frame. (b) Peak positions along the x-axis. The ticks inside brackets are the positions of lane markers.

If each peak in the observation X_{obs} corresponds to one lane marker, *i.e.,* containing no errors, the matching is trivial. However, as aforementioned, X_{obs} may contain the peaks from other bright objects' reflection, *e.g.,* one more peak in X_{obs} in Fig. 3.4(c). X_{obs} may also miss "legal" peaks due to the blocking from surrounding cars. To address this issue, we leverage particle filter which can tolerate observation errors and statistically compute the possibility for each template. After a series of consecutive observations, the probabilistic belief on the real host lane could rapidly accumulate and lead to the highest confidence.

By this principle, we design a particle filter based method. Supposing there are M particles, *e.g.,* $M = 100$. We treat each lane on the current road (obtained from the digital map) as a bucket and then throw particles into these buckets. Initially, the probability or weight into each bucket is identical. Whenever one observation X_{obs} is generated from peak detection, we update weight w_l for particles into lane (*i.e.,* bucket) l. The update is based on the similarity between observed vector X_{obs} and each template. Due to the observation errors, the number of elements in X_{obs} may be different from each template. To measure their similarity with variant lengths,

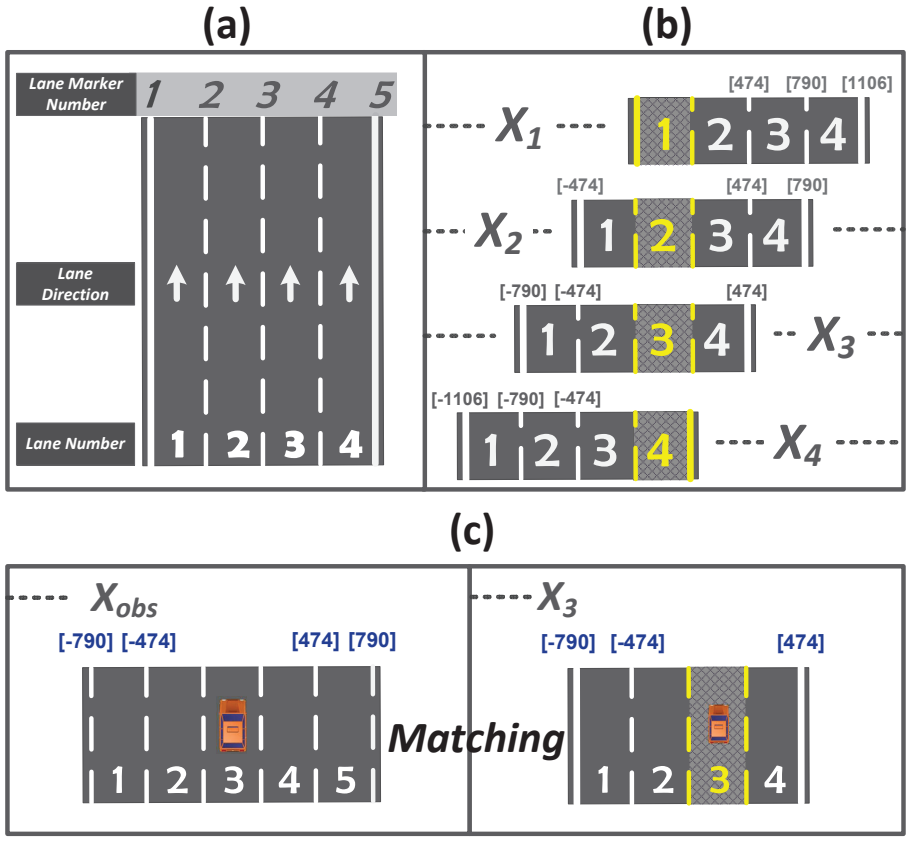


Figure 3.4: Illustration of the particle filter design. (a) One road segment has four lanes with five lane markers. (b) Four templates by assuming the host lane is at each of these four lanes. (c) Template matching based on the observed peak pattern, *e.g.*, X_{obs} matches X_3 best.

we adopt the DTW technique [86]. Given two discrete sequences of different lengths, *e.g.*, X_{obs} and X_1, DTW searches for the best alignment between these two sequences by minimizing the mutual distance using dynamic programming. After the DTW computation for each template, the weight updating can be expressed as:

$$w_l = e^{-d_l/k}, \qquad \text{(Eq. 3.1)}$$

where d_l is the DTW distance of X_{obs} to the l-th lane template and k is a scaling factor. From each video frame i, we can calculate $w_l^{(i)}$ by Eq. 3.1 and further conduct a weighted average

with the previous $w_l^{(i-1)}$ to tolerate peak pattern errors from individual observations:

$$w_l^{(i)} = \alpha \cdot w_l^{(i)} + (1-\alpha) \cdot w_l^{(i-1)}, \quad \text{(Eq. 3.2)}$$

where α is set as 0.5 in our current implementation to balance the contributions from both the current and past observations for avoiding the sudden parameter changes caused by the observation errors. The weight average will restart from the scratch after the car makes a turn. After Eq. 3.2, we further normalize each $w_l^{(i)}$ so that their summation equals to 1, *i.e.,* $\sum_l w_l^{(i)} = 1$. After the weight updating, all particles will be redistributed cross all possible lanes with a new probability. After several rounds of updating, most particles will converge and we select the lane occupied by particles whose weight is larger than 0.5 as the host lane.

Enhanced particle filtering. With this primary design, in Amateur, we also propose an enhancing technique to further improve the host lane identification performance, by detecting and leveraging the event of car's lane-changing. Supposing Amateur detects the car has changed to the right lane. In this case, the left-most lane on the road is unlikely to be the current host lane, as it contradicts to this lane-changing event. Therefore, the weights of particles in each bucket, after the awareness of such an event, can be further adapted to speedup the convergence and improve the reliability.

To leverage such an opportunity, we observe that if the car goes straight, the two markers in X_{mid}, enclosing the host lane, are on the different sides of the frame's center line, as shown in Fig. 3.3(b). When it starts to turn right, the right marker starts to move towards the center line. After the car turns to the right lane, this marker will move cross the center line. The same principle can be applied when the car turns to the left lane. Therefore, we can utilize such a center line crossing phenomenon to detect the lane-changing event. On the other hand, when a car changes its lane, it may change s lanes, where s can be equal or greater than 1. For instance, if the car changes from lane 1 to lane 3, s equals to 2. With different s values, the weights of particles should be adapted differently. However, when the car changes multiple lanes, its heading direction becomes not parallel to the lane's direction. We cannot determine the exact value of s based on the center line crossing phenomenon. To this end, we find that the IMU sensor readings from smart phone can facilitate to determine the value of s.

In Amateur, smart phone is tightly mounted on the car and its IMU sensor readings (along the direction that is perpendicular to car's moving direction) suffice to infer the car status, *e.g.,* moving straight or making a turn. Fig. 3.5 illustrates such accelerometer readings when the

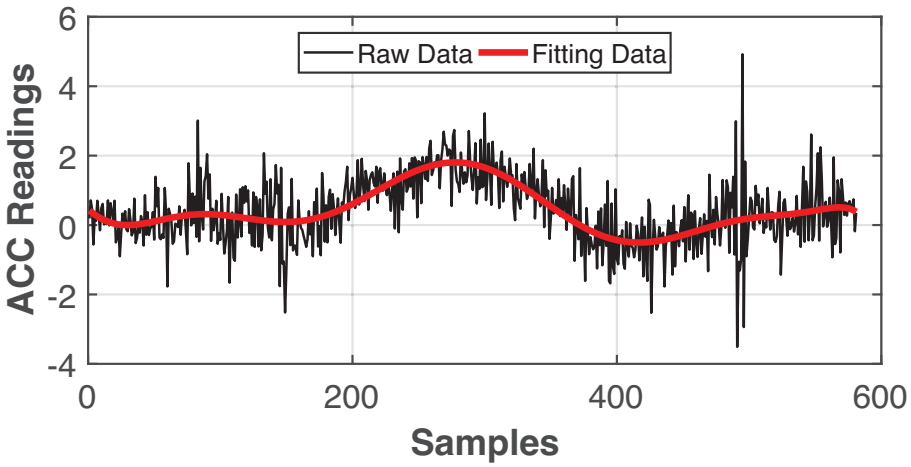


Figure 3.5: Accelerometer readings when the car makes a right turn.

car makes a right turn. It clearly shows a convex bump. We conduct a moving average for accelerometer readings. Normally, the readings are near zero. However, when the car turns to a new lane, its readings exhibit an obvious change. The shape of such reading patterns is highly related to the lane-changing details. More precisely, given the acceleration readings (perpendicular to car's moving direction) under the lane-changing event (detected by the center line crossing phenomenon in the video), *i.e.,* observation $\boldsymbol{O}$, we aim to compute the posterior distribution $\boldsymbol{P}(\boldsymbol{L_s}|\boldsymbol{O})$, where $\boldsymbol{L_s}$ represents the lane-changing event with $\boldsymbol{s}$ lane(s) being changed. Therefore, the Bayesian inference can be expressed as:

$$\boldsymbol{P}(\boldsymbol{L_s}|\boldsymbol{O}) \quad = \quad \boldsymbol{P}(\boldsymbol{O}|\boldsymbol{L_s}) \cdot \boldsymbol{P}(\boldsymbol{L_s}) / \boldsymbol{P}(\boldsymbol{O}). \qquad \text{(Eq. 3.3)}$$

The likelihood for the occurrence of the lane-changing event $\boldsymbol{P}(\boldsymbol{L_s})$ and the observed sensor readings $\boldsymbol{P}(\boldsymbol{O})$ can be estimated from the off-line analysis of the collected live video and sensor data traces. But later we find that their exact values are not needed, and we denote Eq. 3.3 as $\boldsymbol{P}(\boldsymbol{L_s}|\boldsymbol{O}) = \boldsymbol{P}(\boldsymbol{O}|\boldsymbol{L_s}) \cdot \beta$, where $\beta = \boldsymbol{P}(\boldsymbol{L_s})/\boldsymbol{P}(\boldsymbol{O})$. To calculate $\boldsymbol{P}(\boldsymbol{O}|\boldsymbol{L_s})$, we record different IMU readings under the events $\boldsymbol{L_s}$ as templates in advance. Those templates can be utilized to match the actual sensor readings during the driving to compute $\boldsymbol{P}(\boldsymbol{O}|\boldsymbol{L_s})$. For this computation, we further apply DTW to measure the similarity between the observed $\boldsymbol{O}$ and each template, and compute $\boldsymbol{P}(\boldsymbol{O}|\boldsymbol{L_s})$ as $\boldsymbol{P}(\boldsymbol{O}|\boldsymbol{L_s}) = e^{-d_s'/k'}$, when d_s' is the cost of $\boldsymbol{O}$ to each template in DTW match, and k' is a scaling factor. Then we can incorporate $\boldsymbol{P}(\boldsymbol{L_s}|\boldsymbol{O}) = \beta \cdot \boldsymbol{P}(\boldsymbol{O}|\boldsymbol{L_s})$ to update

particle filter's weights in Eq. 3.1 when the lane-changing event is detected:

$$\hat{w}_l = \sum_{s+i=l} \boldsymbol{P}(\boldsymbol{L}_s|\boldsymbol{O}) \cdot w_i = \sum_{s+i=l} \beta \cdot \boldsymbol{P}(\boldsymbol{O}|\boldsymbol{L}_s) \cdot w_i, \qquad \text{(Eq. 3.4)}$$

where $\boldsymbol{P}(\boldsymbol{L}_0|\boldsymbol{O})$ is defined as 1. The rationale of Eq. 3.4 is: when $s = 0$, w_i is the original weight w_l and only the peak observations contribute to the weight update in line 9 of Algorithm 1; Otherwise, the term $\boldsymbol{P}(\boldsymbol{L}_s|\boldsymbol{O})$ describes the likelihood that the car changes from the i-th lane to the l-th lane by changing s lanes in total, *i.e.,* $s + i = l$. In this case, the peak observations and accelerometer readings of the lane switching both contribute to the weight update in line 7 of Algorithm 1. In other words, in addition to the original weight w_l, Eq. 3.4 also considers the transitions from other lanes. After Eq. 3.4, we further normalize all $\hat{w}_l$, so that their summation equals to 1, and then apply the normalized $\hat{w}_l$ in Eq. 3.2 to throw particles to identify host lane. Note that the factor β will be canceled out during normalization and we do not need to know its exact value because given the same observation, they are equal in different lane-changing events.

A formal description of the weight updating is given in Algorithm 1. In particular, the algorithm maintains a set of particles and each particle corresponds to a possible host lane. Then, the algorithm considers to associate each pair of peak observation and accelerometer readings with every particle to calculate how likely (probability) each lane could be the host lane (line 7). When there is no lane-switching event, the corresponding probability is calculated in line 9. After the probability contributed by new peak observations and accelerometer readings is calculated, the algorithm updates the weight of each particle (line 11) and normalizes their weights (line 15). The algorithm utilizes DTW function (Algorithm 2) to calculate the weight of each particle, which finds the best match between two time sequences (line 11–16). In DTW, the algorithm usually utilizes the Euclidean distance (line 13) as the cost function. Finally, the DTW function returns the cost value (Euclidean distance under best match) of two input time sequences (line 17), which is used in the weight updating in line 7 and 9 in Algorithm 1.

Lane-changing arrow placement. As stated before, after the weight update, all particles will be redistributed cross all possible lanes with a new distribution. After several rounds of updating, most particles will converge and we select the lane occupied by particles whose weight is larger than 0.5 as the host lane. After the host lane is identified, if it differs from the suggested lane by the navigation service, a lane-changing annotation is displayed in the live

Algorithm 1: Weight update for particles in the host lane identification

Input: peak template X_b, peak observation X_{obs}, accelerometer template $\boldsymbol{L}_s$ and observation $\boldsymbol{O}$;
Output: updated weights for all the particles;
for *each template* X_b **do**
  for *each template* $\boldsymbol{L}_s$ **do**
    for *each particle* p **do**
      if $\boldsymbol{L}_s$ *event happens* **then**
        $w_{l\pm s} = e^{-DTW[X_b,X_{obs}]/k - DTW(\boldsymbol{L}_s,\boldsymbol{O})/k'}$; `// Both peak patterns and accelerometer readings contribute to the weights of the particles`
      else
        $w_l = e^{-DTW[X_b,X_{obs}]/k}$; `// Only peak patterns contribute to the weights of the particles`
      end
      Update the weight of particle p following equation Eq. 3.2 ; `// Updating of the weights`
    end
  end
end
Normalize the weights of all particles ; `// Normalization of the weights. Function` DTW `is depicted in Algorithm 2`

road condition video, where the annotation arrow design is introduced in Section 3.1.3.

Peak quality control

Although the particle filter could automatically tolerate observation errors, as a matter of fact, many peak errors can be excluded before the particle filtering to speedup the host lane identification.

To this end, for any peak pattern as depicted in Fig. 3.2, peaks are empirically characterized by three features:

- Height h: the gray scale value of a peak.
- Width w: the width of the detected peak.
- Prominence p: the prominence of a peak within range w.

Algorithm 2: Function of DTW used in Algorithm 1

Input: series $\boldsymbol{L}_s$ and series $\boldsymbol{O}$;
Output: cost under DTW matching;
Function `DTW(`$\boldsymbol{T}$, $\boldsymbol{O}$`)`**:**
    for $i = 1$ *to* $lenth(\boldsymbol{T})$ **do**
        $DTW[i,0] = \infty$; `// Initialization`
    end
    for $j = 1$ *to* $lenth(\boldsymbol{O})$ **do**
        $DTW[0,j] = \infty$; `// Initialization`
    end
    $DTW[0,0] = 0$;
    for $i = 1$ *to* $lenth(\boldsymbol{T})$ **do**
        for $j = 1$ *to* $lenth(\boldsymbol{O})$ **do**
            $cost = d(\boldsymbol{T}[i], \boldsymbol{O}[j])$; `// The cost is measured by the Euclidean distance`
            $DTW[i,j] = cost + \min\{DTW[i-1,j], DTW[i,j-1], DTW[i-1,j-1]\}$; `// DTW calculation`
        end
    end
    return $DTW[length(\boldsymbol{T}), length(\boldsymbol{O})]$; `// Return the final DTW value`
    End Function

Supposing the width (pixel-wise) of one frame is N. Along a row extracted from one frame, the gray scale values of the N pixels are denoted as $G = \{g_i\}_{i=1}^{N}$. Through our extensive investigations, we find the following empirical criteria could well characterize the peaks corresponding to lane markers:

$$h \geq \frac{1}{2} \cdot max\{g_i\}_{i=1}^{N}, \tag{Eq. 3.5}$$

$$p = h - min\{g_i\}_{i=g^k-\frac{w}{2}}^{g^k+\frac{w}{2}}, \tag{Eq. 3.6}$$

$$\frac{1}{3} \cdot p \leq p - G_{10th}. \tag{Eq. 3.7}$$

where G_{10th} is the 10^{th} percentile in set $\{g_i\}_{i=g^k-\frac{w}{2}}^{g^k+\frac{w}{2}}$ and g^k is the position of the k-th peak. Eq. 3.5 ensures the detected peaks should be bright enough. Eq. 3.6 and Eq. 3.7 ensure that the detected peak should be sharp enough. The three features, *e.g.,* h, w and p, are carefully investigated.

Due to the perspective effect in camera sensors, these parameter settings may vary if the

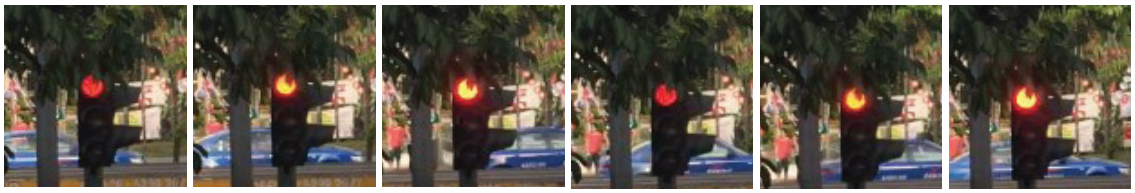


Figure 3.6: Cropped parts of six continuous frames containing the traffic light from live video.

straight horizontal line in Fig. 3.3(a) is selected differently. To collect consistent observations, we adopt a fixed offset in Amateur, *e.g.,* 230 pixels to the frame bottom on our current development platform Nexus 5X. The 230 pixels on Nexus 5X correspond to a horizontal line about 5 meters in front of the car. This distance is rarely blocked by the front car, as the driver needs to keep certain distance to the front car during the driving. Therefore, the pixels from this position could provide a reliable lane marker detection on our current implementation platform. In Section 3.3, we further discuss how to set this parameter on different smart phones. On the other hand, as the pixels adopted to detect lane markers correspond to a horizontal line that is not far away from the car, when a car moves along a curvy road, the observed bending of all the lanes is also similar to each other. As a result, after we use DTW to find the similarity between the observed peak pattern and the peak templates, the curve lanes have the same impact on each template. The calculated similarity can still update the weights of the participle filter as on a straight road.

3.1.3 Intersection Inference

With the host lane identification design in Section 3.1.2, the lane switching annotation arrow can be displayed in the live road condition video. To further display an annotation arrow for the intersection turning, the position of the intersection needs to be specified in the video frames and we leverage traffic lights as an anchor to estimate this position for the arrow placement.

Traffic lights detection

To determine the position of traffic lights, we utilize the flickering feature of LED bulbs. As traffic lights are powered by alternating current with a time varying intensity, *i.e.,* of 50 or 60 Hz frequency, the light intensity varies accordingly, as shown in Fig. 3.6. Such a difference can be unveiled by subtracting consecutive frames. According to the readings of GPS, Amateur

can detect whether the car gets close to an intersection, *e.g.,* 50 meters in front. If so, the intersection inference module conducts the frame subtraction to unveil the flicking feature of traffic lights to identify the position of traffic lights in video frames, based on which the module further applies a pin-hole model (Section 3.1.3) to estimate the position of the intersection, so that the turning instruction arrow can be correctly placed.

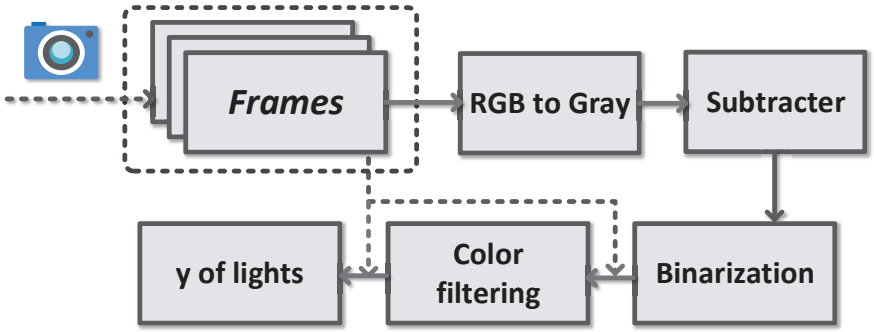


Figure 3.7: The flowchart of traffic lights detection for the intersection inference in live video.

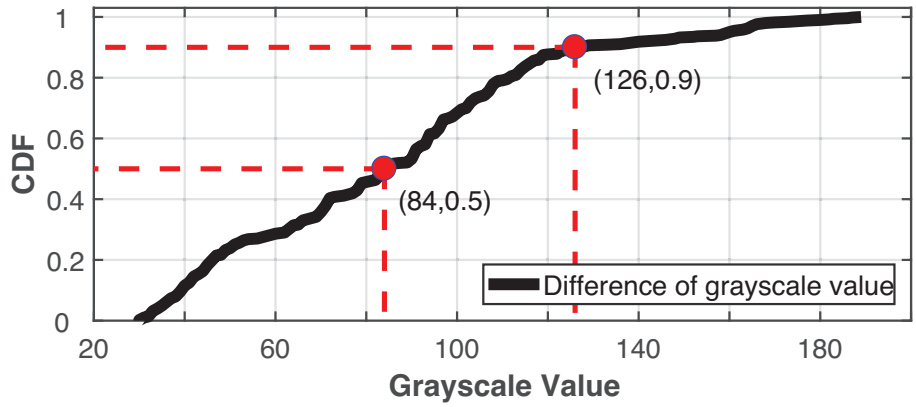


Figure 3.8: Statistical results of differences of pixel at traffic lights' positions in gray scale domain.

Fig. 3.7 illustrates the flow chart of the traffic lights detection. Frames are streamed into a subtracter. It outputs the subtracted results of two consecutive video frames. As the camera captures the real-time video with 30 FPS and the frequency of the traffic light's intensity varying is 50 or 60 Hz, the area with significantly changed pixel values likely indicates traffic lights due to the flickering feature of LED bulbs, while other areas will not exhibit such a high intensity varying frequency because they reflect the sun light, which can be viewed as a constant

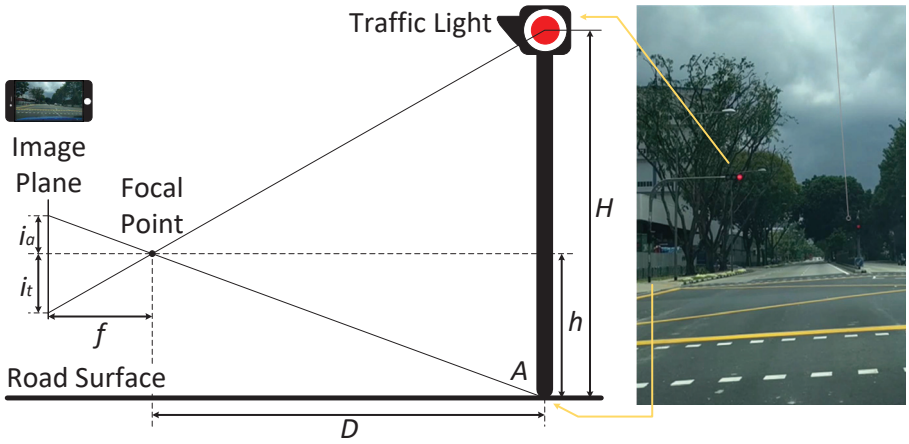


Figure 3.9: Pin-hole model to calculate the position of an intersection in a video frame.

approximately within a short time span. Fig. 3.8 shows that the statistic difference of pixels representing traffic lights varies from 31 to 189 in the gray scale domain. In particular, 70% of the pixels at traffic lights' position is larger than 60 in the gray scale domain. On the contrary, the pixel changing in the gray scale domain for other areas is rarely larger than 50. Therefore, we use the threshold 50 to binarize the subtraction result in our current implementation.

After the binarization, the preserved area in the frame mainly indicates the traffic lights. We then apply the color filter [54] to extract the positions of pixels representing the traffic lights and use their averaged position to denote the traffic light. After the identification, the y coordinate of the traffic light, *i.e.,* its height, is passed to the pin-hole model, as illustrated in in Fig. 3.9.

Pin-hole model

In Amateur, we utilize the position of traffic light y in the frame to infer the position of the intersection. We observe that the supporting point of a traffic light, *e.g.,* point A in Fig. 3.9, could serve as a good reference to indicate the nearby intersection because it approximately shares a similar depth as the intersection to the car in the video frame.

Therefore, we introduce a pin-hole model as shown in Fig. 3.9 to fulfill the calculation. We first determine the position of the traffic light in the frame in Section 3.1.3 and denote this position as i_t, *i.e.,* $i_t = y$ in Fig. 3.9. When there are multiple traffic lights are detected, Amateur selects the one with the largest y value (*i.e.,* the highest one on the screen) as the indicator. This is because when a car is approaching a crossroad, the traffic light that the driver needs to follow

is in front of the car (maybe far away but the direction is almost upright). Therefore, according to the perspective relation, such traffic light has the highest y value on the screen. On the other hand, the physical height of traffic light is normally fixed in a city, which can be denoted as H. After the smart phone is mounted to the car, we can also measure its distance to the ground h. With these three parameters, the position of the supporting point, denoted as i_a, in the video frame can be calculated by:

$$i_a = \frac{h}{H-h} * i_t. \qquad \text{(Eq. 3.8)}$$

After i_a is obtained, the upper edge of the annotation arrow is aligned with this height to indicate a turning at the intersection.

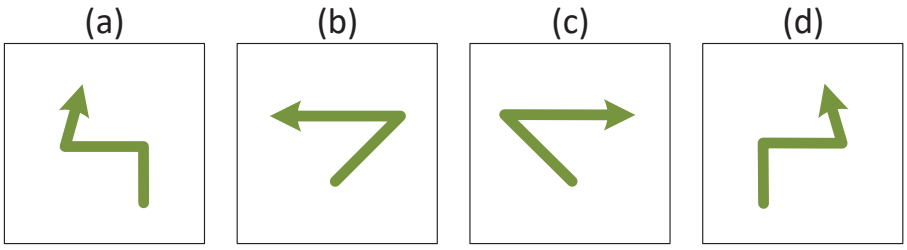


Figure 3.10: Instructional annotation arrows designed in Amateur. (a) Left-lane changing. (b) Left turning. (c) Right turning. (d) Right-lane changing.

Annotation arrow design and display. The primary arrow set to navigate the driving consists of four basic elements. The first two are *turning left or right*, as shown in Fig. 3.10 (b,c), which inform the driver to turn left or right at the crossroad. The other two arrows are *left- or right-lane changing*, as Fig. 3.10 (a,d) depicts. These two arrows inform the driver how to change the lane, where the lane changing number also appears on the top of the arrow to facilitate the driver to recognize. We show concrete use cases of these annotation arrows in the evaluation part (Section 3.2). One more arrow might be included in Amateur, *i.e., going straight*. If there is no turning or lane-changing event, the default operation is going straight. Therefore, we skip this arrow to reduce the amount of arrows displayed that may distract the driver's attention.

To display the annotation arrows, in addition to the smart phone's screen, another possible solution is to project them on the dashboard or windshield of the car. However, this may require the car release the permission to let third-party navigation system access its dashboard, which needs the car manufacturer's cooperation and may also have security risk concerns. In addition, displaying information on the dashboard or windshield requires the transparent windshield to support as well, which is more expensive. Therefore, we provide a lightweight and easy-to-

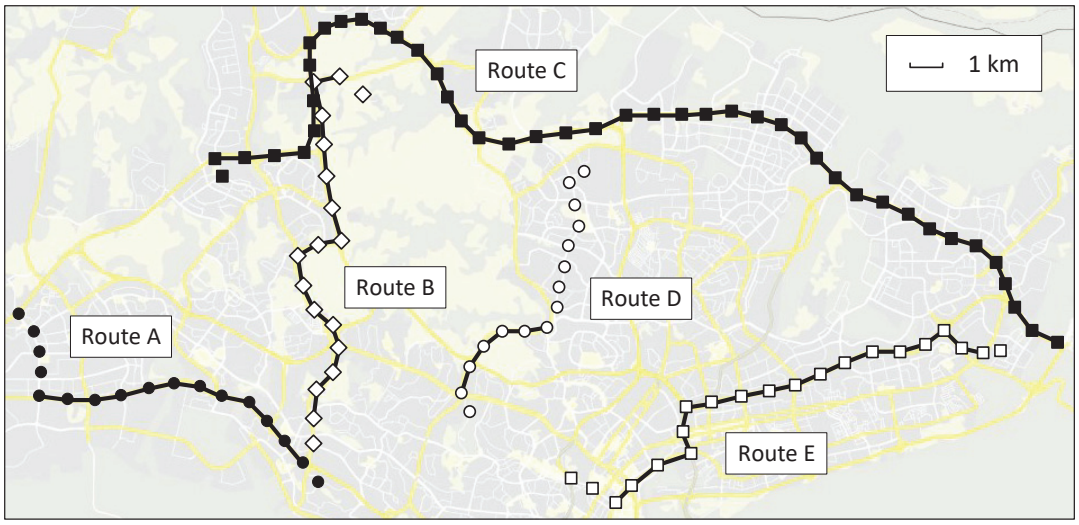


Figure 3.11: Experimental routes in the evaluation and their total length is about 96*km*.

deploy solution which is transferable regardless of car models and windshield types by using the smart phone, due to its independence to car's internal system and the wide availability to most drivers.

3.2 Implementation and Evaluation

3.2.1 Experiment Methodology

We implement Amateur on Nexus 5X smart phone, which is equipped with the IMU sensors, two cameras and a $1920 * 1080$ resolution screen. It has a 1.8 GHz hexa core 64-bit ARMv8-A processor and 2GB RAM. In the implementation, we invoke the OpenCV library for the image processing. The sampling rate of GPS is set to be $1hz$ during this experiment.

To evaluate the performance, we hire taxi cabs in Singapore and attach the mobile phone to the windshield with a mobile phone holder. The phone is plugged with the USB power and there is thus no energy concern to the usage of Amateur. Prior to the experiment on each taxi, we measure the distance (one time effort for each taxi) from the phone holder to the ground, *i.e., h* in the Eq. 3.8 used by the pin-hole model. The evaluation lasts several weeks and covers 5 experimental routes (depicted in Fig. 3.11) cross the city with the total length of over 96 *km*, where the connected markers are the expressway and isolated markers are the

Table 3.1: Detailed information of the 5 experimental routes in Fig. 3.11.

Route	A	B	C	D	E
Length (km)	12.7	17.4	36.8	12.5	16.8
Length of Expressway (km)	8.73	15.35	36	4.46	14.28
Length of Highway (km)	3.97	2.05	0.8	8.04	2.52
Number of Traffic Lights	15	19	5	18	17
Average Velocity (km/h)	49.7	51.2	56.1	45.4	53.7

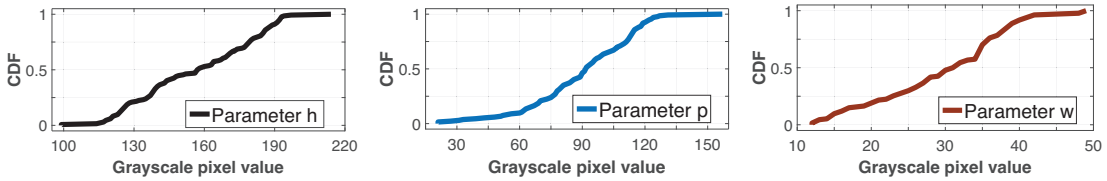
highway in reality. These five routes contain both the highways and expressways, which also cover both the east-west and north-south directions in the city. Their more detailed information is summarized in Table 3.1. In particular, the longest testing route (Route C denoted as ■ in Fig. 3.11) lasts about 36.8 *km* and only 0.8 *km* of them is on the highway. The shortest route is Route D of 12.5km, in which the length of its highway lasts longer than its expressway.

Throughout the experiments, each route is evaluated multiple times, in 1) different times of a day, *e.g.,* in the daytime, at night, at sunset and at nightfall with different traffic patterns and light conditions on the road, and 2) also with different weather conditions, like sunny and rainy, to achieve a comprehensive performance evaluation. The total traveling distance is more than 300km in the evaluation, and we report Amateur's performance from the following four detailed aspects in this section.

As Amateur consists of two major modules, *i.e.,* host lane identification and intersection inference, in this section, we first evaluate the performance of these two modules individually (Sections 3.2.2 and 3.2.3) as follows.

Host lane identification. For the host lane identification, we adopt the accuracy rate to evaluate its performance, which is defined as the ratio between the correctly placed lane-changing arrows and the total number of such arrows displayed during the navigation. We check the accuracy of host lane identification under different conditions. If host lanes are determined correctly, lane switching arrows can be properly displayed. In addition to the accuracy, we also investigate the parameter settings and the particle filter convergence speed for this module.

Intersection inference. For the intersection inference module, we also use the accuracy to quantify its performance. As we record all the navigation videos for obtaining the ground truth, we can manually check the traffic light detection accuracy for the evaluation. On the other hand, we also examine the position difference between the placed turning arrows at each



3.12.a: Peak height h 3.12.b: Peak prominence p 3.12.c: Peak width w

Figure 3.12: Investigation of the three features, *i.e.,* h, p and w in Fig. 3.2, to detect the gray-scale pixel value peaks.

intersection, which reflects the effectiveness of the pin-hole model. Intersection inference is evaluated on highways, since there are no traffic lights on expressways. On expressways, identifying the correct lane alone is sufficient to navigate the driving.

In addition to the investigation of these two individual modules, we further evaluate the overall instruction correctness performance of Amateur in Section 3.2.4 and also conduct a concrete user study in Section 3.2.5.

Instruction correctness. We then test whether the pop-up instructions from Amateur are correct or not, comparing with the ground truth obtained from our manual investigation of the live video off-line. For each trip in the experiment, all the annotation instructions from Amateur can logically form a vector, in which each element is one of the four annotation arrows depicted in Fig. 3.10.

User feedback. Finally, we conduct the survey from each taxi driver to get their feedback to Amateur compared with the traditional navigation service.

In the following, we report the evaluation results for each of the four aspects above.

3.2.2 Host Lane Identification

In the host lane identification module (Section 18), we utilize three features, *i.e.,* h, p and w in Fig. 3.2, to detect the gray-scale pixel value peaks, which correspond to the candidate lane boundaries. In this section, we first examine each of these features to verify the design efficacy. In Fig. 3.12.a, the height of each detected peak, h, varies from 100 to 218. The difference comes from the light intensities and we observe that the pixel value increases dramatically on the lane boundary markers. Fig. 3.12.b further indicates that the prominence of a peak, p, is also sharp, *e.g.,* varying from 20 to 159. Finally, Fig. 3.12.c shows that the width of the peak,

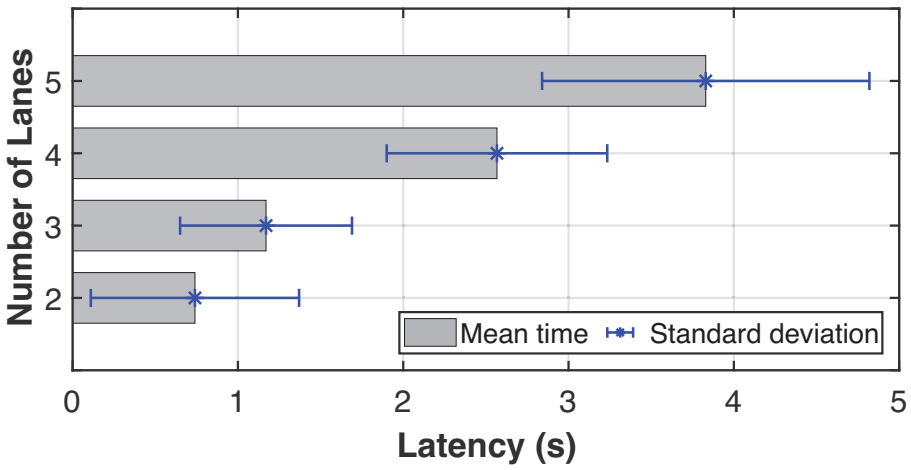


Figure 3.13: Latency to correctly identify the host lane on the roads with different numbers of lanes.

w, is sufficiently wide enough, ranging from 12 to 49.

The host lane identification in Amateur is based on the particle filter. In Fig. 3.13, we first investigate the latency spent by this particle filter design. In particular, we test this latency performance for different types of roads (with two lanes up to five lanes). We record the time until the particle filter correctly identifies the host lane by comparing with the ground truth. From the result, we can see that the average latency varies from $0.74s$ to $3.83s$ when the number of lane increases from 2 to 5. Fig. 3.13 implies that the host lane identification in Amateur is reliable as long as enough time is given. However, in practice, it may not be possible to wait for a relatively long time, *e.g.,* up to $4.85s$ on the road of five lanes, to identify the host lane for only once. We thus consider a more stringent setting to evaluate the host lane identification next.

In Fig. 3.14, we further examine the host lane identification performance within a short time window, *e.g.,* $2s$. In particular, after the particle filter starts a host lane identification, if the weight is larger than 0.5 (according to the design in Section 3.2.1) on one lane within $2s$, it is reported as the current host lane. If the $2s$ time window is reached but there is no lane accumulating a larger weight, the filter outputs the lane with the highest weight at this moment as the host lane. Then for different types of roads (with two lanes up to five lanes on the x-axis), we report the average latency for the host lane identification (left y-axis) and the identification accuracy (right y-axis) in Fig. 3.14. When the number of lanes is small, *e.g.,* 2, the particle filter converges in nearly $0.5s$ and achieves 100% accuracy. As the number of lanes increases,

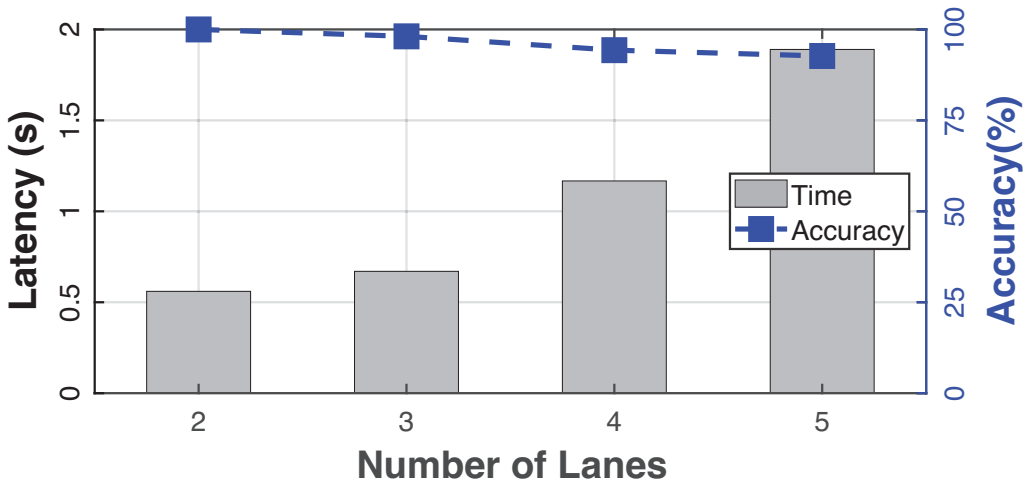


Figure 3.14: Average delay to identify the host lane (left y-axis) and the identification accuracy (right y-axis) on the roads with different number of lanes.

the accuracy decreases, *e.g.,* for a road of five lanes, the accuracy is 92.7%. The performance slightly degrades for more lanes because the distribution of particles is more dispersed and the peak amount increases as well. In this case, the confidence of each peak is reduced (spreading to other peaks). As a car normally stays on a road segment longer than 2 seconds, it explains the good performance achieved by our host lane identification design in practice.

3.2.3 Intersection Inference

For the intersection inference module, we first examine the accuracy of the traffic light detection. Since there are no traffic lights on the expressway, the results in Fig. 3.15 are all from the common roads. As the red light has the longest wave length than other two lights (green and yellow), it has a better penetration and Amateur mainly detects the red light on the screen. From the result, we can see that the detection rate for the red light is highly accurate, where the precision and recall are 99.72% and 99.28%, respectively. As a comparison, if we detect all other two lights at the same time, the precision and recall could slightly degrade to 99.61% and 95.29%, respectively. The reason is that the vehicles are static when detecting red lights. In fact, Amateur can tolerate certain miss-detected traffic lights. We only need to find out the positions of traffic lights on screen when vehicles are distant to lights. Those lights locating at the next intersection can be detected when cars are about 50 meters away from them, which is enough for drivers taking actions and for Amateur displaying instructional annotations on the

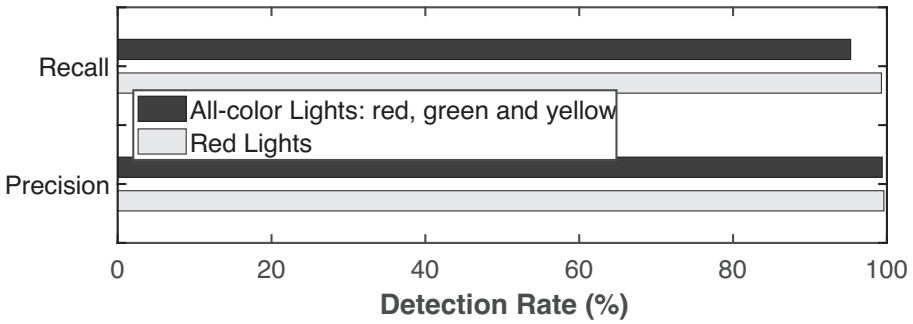


Figure 3.15: Precision and recall of the traffic light detection. We compare the detection for red light only and the detection for all three lights together.

Figure 3.16: Illustration of the turning arrow placement offset.

screen.

After the traffic light is detected, Amateur applies the pin-hole model to determine the position on the screen to display the turning arrow. In our current implementation, the end of a turning arrow is displayed near the middle of the screen's center line and its upper part should ideally align with the intersection in front of the car. However, due to the measurement error, the turning arrow may exhibit an offset with respect to the actual intersection position in the screen, which is denoted as Δ in Fig. 3.16.

In Fig. 3.17, we plot the CDF of Δ for all the turnings encountered in our experiment. In addition, we also compare with the performance achieved by the traditional navigation service based on GPS (its result is computed for the evaluation only, not displayed on the screen).

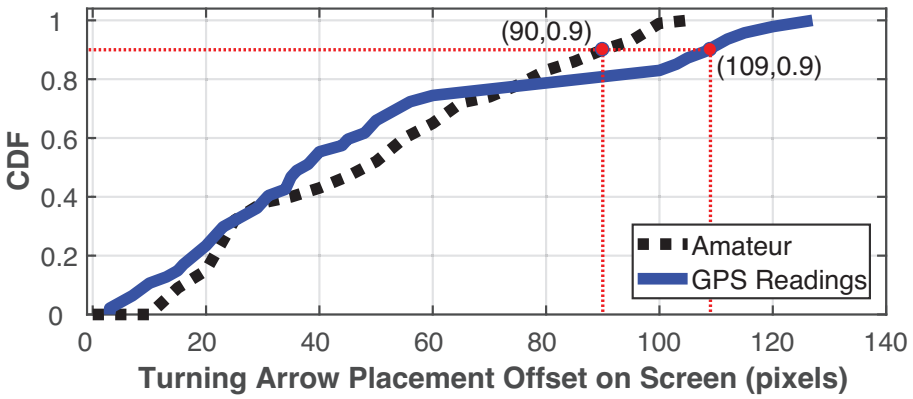


Figure 3.17: CDF comparison of the offsets of turning arrow placement on the scree between Amateur and GPS readings in the unit of pixel.

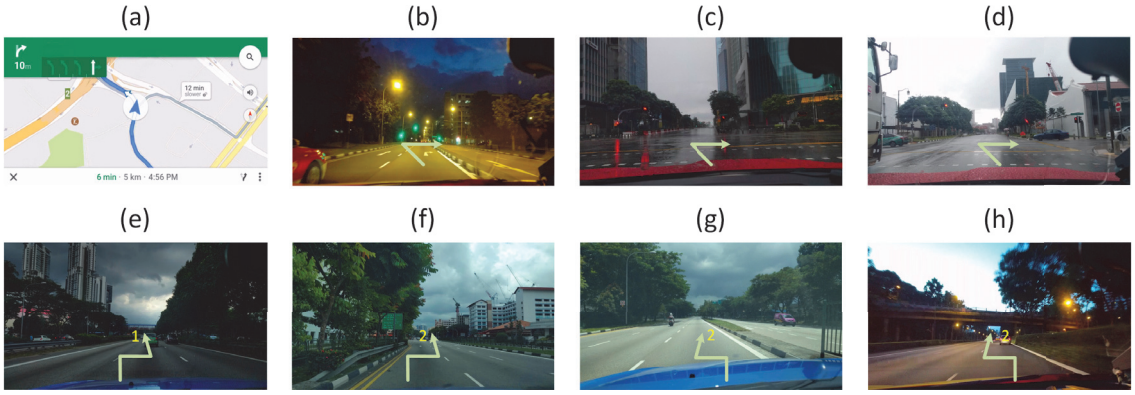


Figure 3.18: Screen shots of Amateur and traditional navigation service. (a) Traditional navigation service; (b) Turning arrows at night; (c) Turning arrows at rainy sunset; (d) Turning arrow in rainy daytime; (e) Right 1-lane changing; (f) Right 2-lane changing; (g) Left 2-lane changing; (h) Left 2-lane changing at sunset.

From the result, we can see that the average and maximum placement offsets of GPS are 49 (12m in terms of distance) and 127 (41.2m) pixels, respectively. Amateur could reduce them to 44.7 (9.6m) and 105 (26.5m) pixels, respectively. After we analyze the captured video for each turning, we find that two methods perform comparably when the GPS signals are strong, while Amateur can leverage the pin-hole model to further improve the GPS's performance in the areas with more skyscrapers nearby.

In Fig. 3.18, we show a series of concrete Amateur's navigation cases in different scenarios. Compared with the traditional navigation service based on the digit map as in Fig. 3.18(a), Amateur is more convenient and user friendly. We also find that Amateur performs well even

in the rainy and dim environments. In particular, in the dim conditions, such as in Fig. 3.18(b), (c) and (e), cameras automatically increase the exposure time to improve the frame quality. However, the increased exposure time may affect the detection of flickering of LED bulbs. To address this issue, we currently divide each second into two $0.5s$ parts — one for the flickering capture and the other part after a longer exposure time for getting clear frames. We can also utilize the latest smart phone like Samsung Galaxy S9, Huawei P20 Pro for better image quality in dimmer environment. Fig. 3.18(e) to (h) further show the lane switching arrows after the lane-changing event is detected. The number above the arrow indicates the number of lanes that the car should change.

3.2.4 Instruction Correctness

To evaluate the overall instruction correctness, we leverage the Levenshtein distance metric [60] to measure the difference between the ground truth vector and the generated vector by Amateur. It counts how many instructions in the generated vector by Amateur need to be modified, so that it becomes the same as the ground truth vector. Supposing the generated vector is $V_{gen} = \{LC2, L, RC1, R\}$ and ground truth vector is $V_{tru} = \{LC1, L, RC1, R\}$. In these two vectors, LC2 means "left changing by 2 lanes"; RC1 means "right changing by 1 lane"; L and R mean "turning left" and "turning right", respectively. The Levenshtein distance between V_{gen} and V_{tru} is 1, which means editing one element in V_{gen}, from $LC2$ to $LC1$, can make these two vectors equal. In general, the smaller the Levenshtein distance indicates higher instruction correctness.

Fig. 3.19 illustrates the statistical results of Levenshtein distance on each road and examine its performance under different time slots when there are different traffic patterns, *i.e.,* morning peak, evening peak and off traffic peak. From the result, we observe that Route C has the largest Levenshtein distance 9 when running Amateur during the evening peak, while the smallest one is on Route A without the traffic peak. In Fig. 3.19, compared with the total number of annotations made for each road, *i.e.,* 43, 74, 101, 67 and 86 for Routes A to E respectively, the instruction correctness of Amateur is high, ranging from 88.4% to 97.3%.

Even the accuracy may vary in different traffic conditions, we find that **none** of instruction errors lead to the wrong navigation destination. To further understand the reason of these annotation errors, we refer to the ground truth and observe that the errors are all caused by the lane-changing event. When the current host lane is detected wrongly, Amateur will recommend

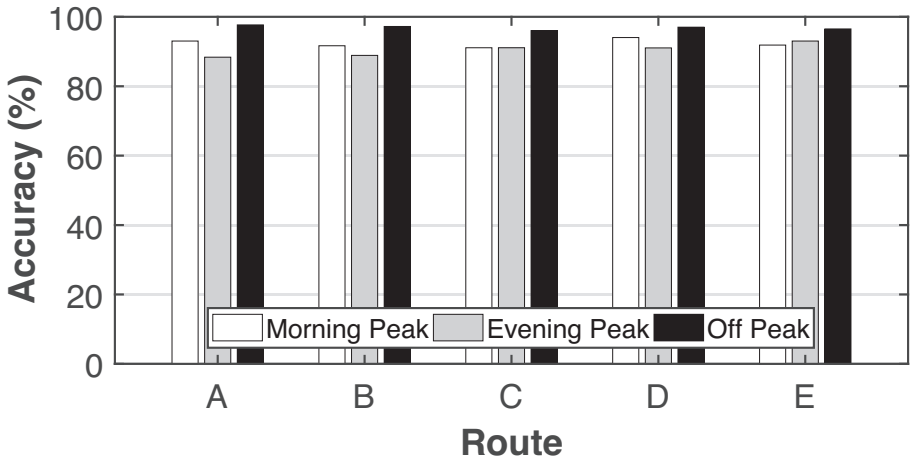


Figure 3.19: Accuracy of arrow placement on the different routes with different traffics.

drivers to change the lane, which actually is not needed. However, Amateur keeps tracking the host lane by our particle filter design and this unnecessary lane-changing can be corrected by changing back to the real interested-lane, which is essentially error-tolerant. After we examine the ground truth, we find that the reason to cause the host lane detection error is mainly due to the shield of lane markers by nearby vehicles or some bright reflections on the road (*i.e.,* noise). Fortunately, nearby cars do not cover the lane markers at the same position for very long time. Their impact is thus temporary. Moreover, the error rate itself is not high, *e.g.,* less than 10%. Therefore, we observe that such errors did not cause the navigation to a wrong destination throughout our experiments. Please note that in case that Amateur made a wrong navigation, similar as the existing navigation systems, it will re-calculate the route and treat it as a new navigation from the current location to the destination. In summary, Amateur can provide convenient and reliable navigation experience in practice.

3.2.5 User Study

Study design

Finally, we conduct a with-in subject user study with 50 taxi drivers to understand the user's experience of Amateur in practice and we also compare it with the traditional navigation method, *e.g.,* Google Maps. In the user study, taxi driver uses one of two navigation systems (Amateur or Google Maps) first on route D of Figure 3.11. After the taxi reaches the destination, we ask the driver to come back to the origin and drive again using another one. For the first

25 taxi drivers (Group A), they use Amateur first and then Google Maps. The other 25 taxi drivers (Group B) switch the order of the system usage. This is a standard technique named Latin Square [80] to counterbalance the order effect. After arriving the destination each time, the taxi driver rates the system just used for the navigation. The questionnaire is detailed in Table 3.2.

Table 3.2: Likert scale rating questions in our user study.

No.	Rating Question Statement
Q1	It was easy to navigate using this navigation service.
Q2	I need to pay extra attention on this navigation service when driving.
Q3	This navigation service provided user-friendly guidance.
Q4	This navigation service was useful in helping me navigate properly.
Q5	It was easy for me to learn how to use this navigation service.
Q6	I paid most of my attention on driving using this navigation service.
Q7	The guidance was user-friendly to interact with.
Q8	This navigation service provided me with effective guidance.

1) Question design. For the question design, we consider the following four key points that we believe are important to evaluate a navigation system.

- **Ease of use**: we propose Q1 and Q5 in Table 3.2.
- **Perceived distraction**: we propose Q2 and Q6 in Table 3.2.
- **Navigation experience**: we propose Q4 and Q8 in Table 3.2.
- **User-friendliness**: we propose Q3 and Q7 in Table 3.2.

In particular, "ease of use" and "navigation experience" have been adopted in a prior simulated AR-based navigation design [50]. Since they have different survey targets, we only utilize these two points instead of their exact questions. In addition, we further propose "perceived distraction" and "user-frienliness" in our user study, which are closely related to our design. Based on the theory from MIS (Management Information System) research domain, two questions for each aspect could improve the reliability and validity of the questionnaire [23].

2) Participants. Table 3.3 summarizes the detailed information of the participanting taxi drivers. In Group A, there are 21 male and 4 female drivers. Their ages are from 32 and 61

years old (Mean: 40.3 years old) and driving experiences are within 2 to 27 years. In Group B, there are 19 male and 6 female drivers. Their ages are from 31 and 57 years old (Mean: 42.8 years old) and driving experiences are within 1 to 30 years.

Table 3.3: Detailed information of the participating drivers.

Property	Description of Group A	Description of Group B
Age	32-61 (Mean 40.3 Year)	31-57 (Mean 42.8 Year)
Driving Experience	2-27 (Mean 15.8 Year)	1-30 (Mean 17.6 Year)
Gender	21 Male (84%), 4 Female (16%)	19 Male (76%), 6 Female (24%)

3) Rating methodology. For each question, we adopt the standard 7-point Likert scale rating mechanism, where each question will be rated by a scaling score from 1 (strongly disagree) to 7 (strongly agree). The results from these 50 taxi drivers are summarized in Fig. 3.20.

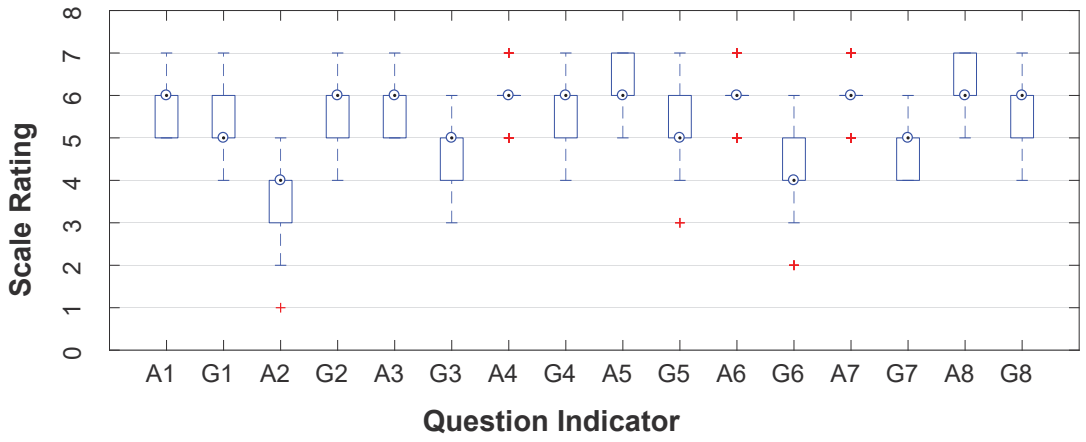


Figure 3.20: Results of Likert scale rating (1: strongly disagree $\sim$ 7: strongly agree). On x-axis, 'A' means Amateur; 'G' means Google Maps and the following number is the question number in Table 3.2. Median is shown as a $\odot$. The ratings for A4, A6 and A7 have different means and standard deviations. However, due to the illustration, they look to be identical in the figure.

4) Numeric analysis. With the 7-point Likert scale rating mechanism and the results in the Fig. 3.20, we can then apply the standard method called Wilcoxon Signed Rank Tests to mathematically analyze the user study, where the significance level is set as $\alpha = 0.05$. The results of inferential statistics analysis and the observations are detailed in the following.

Results

With the 7-point Likert scale rating mechanism, we then apply the standard method to mathematically analyze the user study and obtain the insight for each question design considerations. In particular, the p-value results are summarized in Table 3.4 and we have the following observations.

Table 3.4: The p-values for the 7-point Likert scale rating questions.

Question No.	Q1	Q2	Q3	Q4
$p-value$	0.0016	< 0.0001	< 0.0001	0.0003
Question No.	Q5	Q6	Q7	Q8
$p-value$	< 0.0001	< 0.0001	< 0.0001	< 0.0001

- For the ease of use, the results show that there was a significant difference between the two systems ($p = 0.0016$) for Q1. Fig. 3.20 depicts that Amateur is easier to use comparing with Google Maps, where the mean of our system A5 (*i.e.,* $mean_{A1} = 5.78$) is larger than that of Google Maps G5 (*i.e.,* $mean_{G1} = 5.44$) . In addition, there was a significant difference between two systems ($p < 0.0001$) for Q5. The result indicates that Amateur is easier for drivers to learn how to use the navigation system than the traditional service Google Maps as in Fig. 3.20, where $mean_{A5} = 6.08$ and $mean_{G5} = 5.42$.

- For the perceived distraction, the result indicates a significant difference of two systems for both Q2 and Q6 ($p < 0.0001$ for both). The drivers pay more attention on the traditional service (*i.e.,* $mean_{G2} = 5.5$) than that of Amateur (*i.e.,* $mean_{A2} = 3.72$). The rating results also show that drivers focus more on driving when utilizing Amateur while they might focus more on the navigation display when utilizing the traditional service. The averaged ratings are $mean_{A6} = 5.9$ and $mean_{G6} = 4.26$. Therefore, Amateur can cause much less distractions than the traditional service.

- For the navigation experience, the results of Q4 and Q8 showed that there were significant differences between two systems, with p-value being $p = 0.0003$ and $p < 0.0001$ for Q4 and Q8 respectively. The drivers gave better ratings for Amateur ($mean_{A4} = 5.94$, $mean_{A8} = 6.2$) than the traditional service ($mean_{G4} = 5.44$, $mean_{G8} = 5.56$).

- For the user-friendliness, two systems also exhibit quite different results (Q3: $p < 0.0001$ and Q7: $p < 0.0001$). Compared to the traditional system ($mean_{G3} = 4.76$, $mean_{G7} = 4.92$), Amateur is more user-friendly, implied by rating results with $mean_{A3} = 5.76 > mean_{G3}$ and $mean_{A7} = 6 > mean_{G7}$. The main reason is that the AR display is much easier to understand for the drivers, which is also our original motivation.

In summary, the results indicate that Amateur can provide a better navigation service than the traditional method for the ease of use, perceived distraction, navigation experience and user-friendliness four aspects.

3.3 Limitations and Discussions

In this section, we discuss some limitations of our system and the potential future works.

1) Brightness requirement. The primary requirement of Amateur is that the environment cannot be completely dark. We have tested Amateur at different time periods of a day. The system works well in the daytime and dimmer environments. Since Amateur utilizes commodity camera, if the environment is completely dark, Amateur cannot extract information from the captured images. Nowadays, the camera advances rapidly on smart phones and one major effort of smart phone companies is to improve the picturing performance in the dark. In the near future, once camera can capture more meaningful contents in a dark environment, this limitation can be alleviated and the performance of our system can be improved accordingly.

2) System setup. To deploy a smart phone with Amateur on a car, we need the following two setups. First, the phone needs to be placed along the gravity direction, which is required by the pin-hole model. Fortunately, the motion sensor (more precisely the gyroscope) on the smart phone provides accurate measurement of the posture which we can leverage in the setup phase. The phone posture may not strictly align with the gravity direction even after the setup phase. Through our experiment, we find that the detected position of intersection on the screen could differ from its real position on the screen in tens of pixels, which is still acceptable. Second, we need to select the position of a narrow strip for Amateur to perform the lane boundary detection. In our implementation on Nexus 5X, the current setting is 230 pixels, corresponding to a horizontal line about 5 meters in front of the car. In the future, we plan to introduce a narrow strip appearing on the Amateur's UI and let the driver drag this strip on the screen and move to the appropriate position, which gives the driver freedom to configure this system

parameter. We note that these setups are one-time overhead for installing Amateur.

3) Availability of traffic light information. In Amateur design, we leverage traffic lights as an indicator of the position of intersection using a pin-hole model. This model needs the traffic light height as one input. However we may lack the specification for this information in some cities. In some scenarios, *e.g.,* usually outside of the city, the traffic lights may even miss at intersections due to lack of the infrastructure. In these cases, GPS always serves as a fall back solution to provide the estimated distance to the front crossroad. Such a distance can also help determine the arrow's position in the video. As a matter of fact, our design incorporates with GPS (instead of being independent with GPS) and takes the GPS reading as one system input. However, GPS could become inaccurate in many urban areas, and the detection of the traffic light essentially provides an opportunity to leverage the pin-hole model to improve the distance estimation. To further improve the availability of the traffic light height information for more cities, such a database could be constructed through crowdsourcing, which is one feasible solution to overcome this limitation in the near future.

4) Camera's field of view. For the commodity camera, we find that it can capture five lanes at the same time. If a road has more than five lanes, as long as Amateur has detected one of the left most or right most lanes initially (most likely since a car rarely starts at the center of one road), the particle filter can continuously track the host lane. In case that the host lane tracking is lost or inaccurate for a long time, the driver can explicitly change the lane to let Amateur detect one of the left most or right most lanes to track the host lane again.

5) Camera sampling rate. In our current design, we set the sampling rate of the camera sensor at 30 FPS. We find that the difference between consecutive frames is acceptably small for the normal driving speed, *e.g.,* $< 30km/h$. A larger speed may cause larger difference in subtracted image, leading to longer time to detect the traffic light. Higher sampling rate (*e.g.,* 120 FPS or 240 FPS) can ensure a reliable detection even under a high speed, but it increases the computing overhead. This is a trade-off between the efficiency and the computing overhead. We believe that as smart phones have more powerful CPUs in near future, Amateur can support higher sampling rates, which can further enhance the performance of Amateur.

Chapter 4

Walkway Discovery from Crowdsensing for Pedestrian Navigation[1]

4.1 Preliminary and Motivation

4.1.1 Objective

A road map can be represented as a directional graph $\mathtt{G}(\mathtt{V},\mathtt{E})$, where $\mathtt{E}$ refers to the set of edges that correspond to roads for vehicles and/or pedestrians and $\mathtt{V}$ refers to the set of vertices that correspond to intersection points or terminal points of road segments. We define a **road segment** as follows.

Definition 4.1 ***Road segment.*** *A road segment is a directed edge in graph* $\mathtt{G}$*, that is associated with a deterministic traveling direction and two terminal points.*

Existing digital maps do not contain all the road segments and are thus incomplete. There have been some recent efforts, *e.g., CrowdAtlas* [98], *COBWEB* [85], and *GLUE* [107], made to complete such maps. By comparing the GPS trajectories of vehicles with an existing road map, those works mainly focus on discovering those vehicle-oriented road segments. They mainly perform clustering on the vehicle GPS data not matched to existing road segments based on trajectory distance [98], moving direction [61, 107] or distance between location samples [85]. New-found road segments are constructed either by extracting the centric line of clustered trajectories [98] or connecting the centric points of clustered location samples [61, 85, 107].

[1] This Chapter is partially published on IPSN 2018 [18].

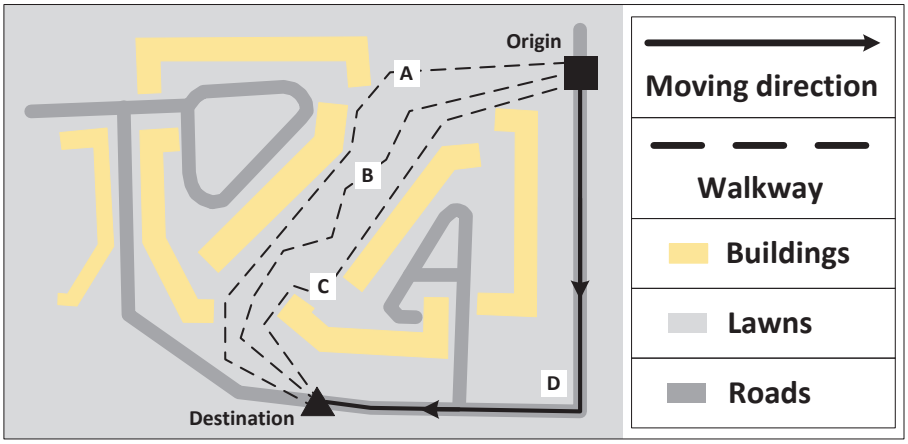


Figure 4.1: The comparisons of motorway and walkway.

Those methods perform well with the motorways which are well structured and with vehicles that strictly follow the directions of the road segments. Different from vehicle-oriented road segments, the walkways for pedestrians are often unstructured and different people may travel with very different trajectories even across the same walkable area. We define a **walkable area** as follows.

Definition 4.2 ***Walkable area.*** *A walkable area is an area bounded by nearby road segments or points of interest (POIs, e.g., residential buildings, schools, etc.). Unconstrained movements of people are allowed within the area.*

Figure 4.1 presents an illustrative example to compare the motorways and walkways. Given the origin of a trip (shown as the symbol ■) and the destination (shown as the symbol ▲), a user can either take the route on road segments (shown as the black solid line, *i.e.,* route D) or choose the walkways (shown as the black dash line, *e.g.,* route A) for this travel. For route D on road segments, she needs to strictly follow the directions of road segments. By contrast, she owns high freedoms on the walkways. For example, she can take route A, B or C for this travel, and even go with any walking path between the two buildings. Putting all these walking paths together gradually forms a walkable area, where people can freely walk at any direction. Such walkable areas widely exist, *e.g.,* lawns, basement of buildings, floors of shopping malls, open fields, *etc.*, and they may provide significant convenience and save the walking time when compared to using existing road segments on the map. Due to the high travel freedom within such walkable areas, however, pedestrians' movements could be irregular with high

uncertainty. As a result, all existing map completion approaches based on direct clustering of locations or trajectories will fail in obtaining the sketch of a walkable area.

For practical utility, discovering and identifying the entire walkable area is not necessary and thus is not the final goal of this paper. We observe that although different pedestrians may choose different paths when walking through the same walkable area, there are typical walkways that connect certain existing road segments or POIs and are frequently used. We thus define the **representative walkway** as follows.

Definition 4.3 ***Representative walkway.*** *A representative walkway represents the connectivity a walkable area serves between two known road segments/POIs. If we specify the intersection points between the road segments/POIs and the walkable area, the representative walkway can be denoted as a polyline connecting the two intersection points and integrated into the road graph* `G` *as an edge. There may be multiple representative walkways connecting different road segments/POIs adjacent to the same walkable area.*

Therefore, instead of precisely discovering the entire walkable area which we may not have sufficient mobility data to support, in this paper we aim at extracting representative walkways that sketch the accessibility of walkable areas in facilitating pedestrians' travel needs. The representative walkways, once identified, are compatible with the current digital road map and can be easily integrated into the map.

4.1.2 NSE Mobility Data

We make use of a crowdsensing mobility data from the National Science Experiment (NSE) [69] of Singapore to support our goal of walkway discovery. NSE is a nationwide project initiated by the National Research Foundation and supported by the Ministry of Education of Singapore. This project involves more than 250,000 students from primary, secondary, high school and junior colleges. Each student carries a smart device called SENSg [103] with various sensors embedded to record their mobility and sense their surrounding environment everyday. The SENSg device collects sensing data every 15 seconds when it is in active mode. The sensor readings are uploaded to the server through nearby wireless hotspots whenever there are (Wireless@SG [104] provides WiFi coverage with over 4,000 hotspots in Singapore and offers free data bundle to the SENSg devices in NSE. WiFi deployed at schools also can be used to upload sensor readings and are the most frequently used). All data used in the project

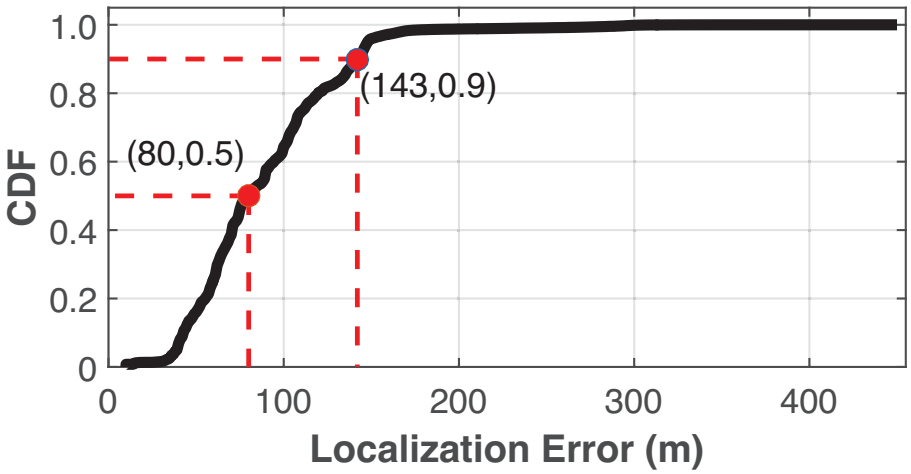


Figure 4.2: CDF of localization error in NSE mobility data.

are anonymized to protect the students' privacy.

Each uploaded record contains 16 attributes, including raw sensor readings and some derived results. Specifically, each record contains the *ID* of the SENSg device, and two timestamps, *i.e., ts* referring the time the readings are taken at device side and *rts* referring the time data are received at the server side. Each record has following mobility attributes:

- **Location.** The location is represented as a pair of latitude and longitude, indicating the current position when the sensing data is taken. The location is derived from the MAC addresses and RSSIs of nearby WiFi hotspots and through a third-party localization service called SKYHOOK [88]. In addition to the location itself a localization error estimation is also provided. As shown in Figure 4.2, the localization service based on WiFi hotspots has varied localization error ranging from tens to hundreds of meters. The 50-percentile and 90-percentile localization errors are 80 m and 143 m, respectively.

- **Step count.** The step count is inferred from the IMU sensor readings in SENSg. The server side keeps tracking the accumulated steps the SENSg user takes.

- **Travel mode.** It indicates the mode of transportation (*i.e.,* walking, bus, MRT train, or private car) the SENSg user takes when the sensing data is taken, which is also inferred from the IMU sensors through an online classification algorithm.

Besides the mobility attributes, each record includes other environmental attributes, *e.g.,* temperature, atmospheric pressure, relative humidity, sound pressure level, light intensity, *etc.*. In

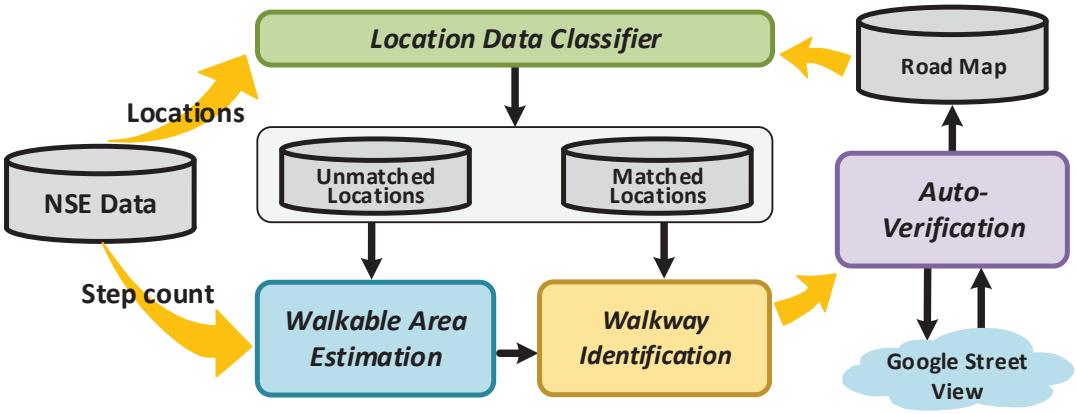


Figure 4.3: The system architecture of VitalAlley.

our study, we rely on mobility attributes from the records of each student which give the student's daily footprint in the city. This study utilizes the mobility data from a total number of 108,337 students for 11 weeks in 2016, which correspond to a distance of trajectories totaling millions of kilometers.

As the natives, students are most familiar with the neighborhood around their home and school. They often make use of the uncharted walkable areas or paths to save their commuting time and their walkway choices contain novel knowledge to existing digital road maps (*e.g.,* open-sourced OpenStreetMaps [73] and commercial maps like Google Maps [34]). In this paper, we primarily exploit the walking trajectories of students to discover the missing walkways.

4.2 System Design

In this section, we present the system overview and then elaborate each component in the following subsections.

4.2.1 System Overview

The system architecture of VitalAlley is illustrated in Figure 4.3. At a high level, VitalAlley consumes the location and step count data from NSE mobility dataset and offers representative walkways to complete existing road maps. Due to localization errors and characteristics of students' mobility (*e.g.,* mainly being active in school and home), not all location data are

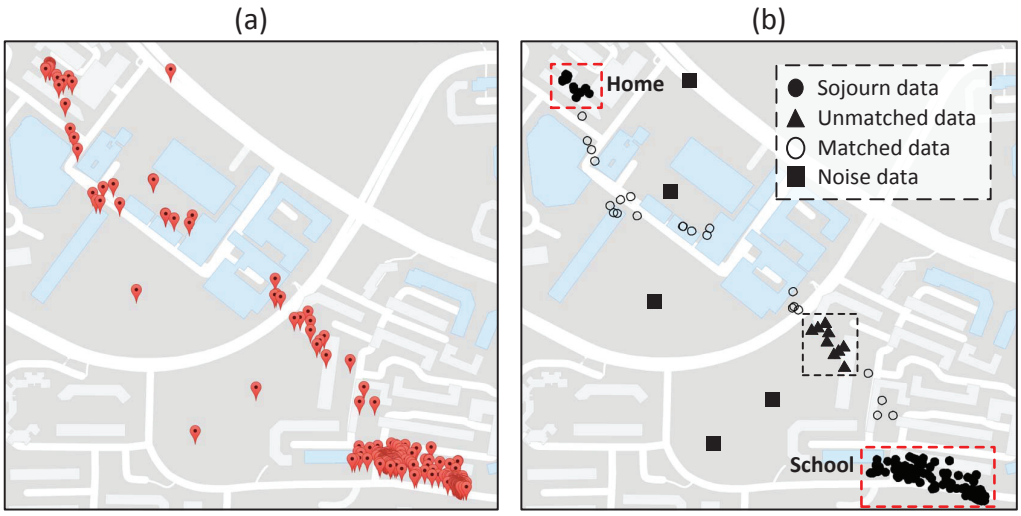
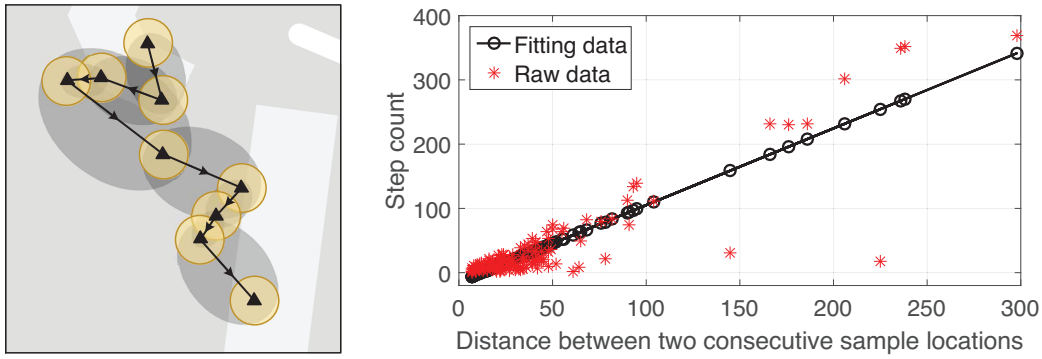


Figure 4.4: Illustration of location data classification on the mobility data of one student in a typical school day.

useful for the walkway discovery. At the very beginning, VitalAlley filters out the location data which are not in *walking* status according to the travel mode in the records. Then it invokes the *Location Data Classifier* module to label the location data as "Noise", "Sojourn", "Matched" and "Unmatched". This module firstly labels the locations with large localization error estimations (*e.g.,* $\geq 143\,m$) as noise data, which are far away form the student's trajectory. The common sojourn places for students are schools, homes, shopping malls, *etc.*, which leads to a much higher data density than usual at a specific place. Thus the module labels such data taken at some sojourn places as sojourn data by exploiting the HDBSCAN algorithm [15]. After filtering out noise and sojourn data, this module classifies the remained data into "Matched" and "Unmatched" through a map matching algorithm [68]. Matched data are those that can well match with existing roads, while the unmatched data cannot well match with any road. Only "Matched" and "Unmatched" locations are helpful for VitalAlley as they may contain knowledges about the missing walkways.

Figure 4.4 illustrates the four types of location data. Figure 4.4(a) plots the raw mobility data of one student in a typical school day, and Figure 4.4(b) shows the classification results. The two clusters correspond to home (top left corner) and school (right bottom corner), respectively. The noises are far away from the student's actual trajectory, and the matched locations are distributed near some existing road segments while the unmatched ones locate away from the roads.



4.5.a: 4.5.b:

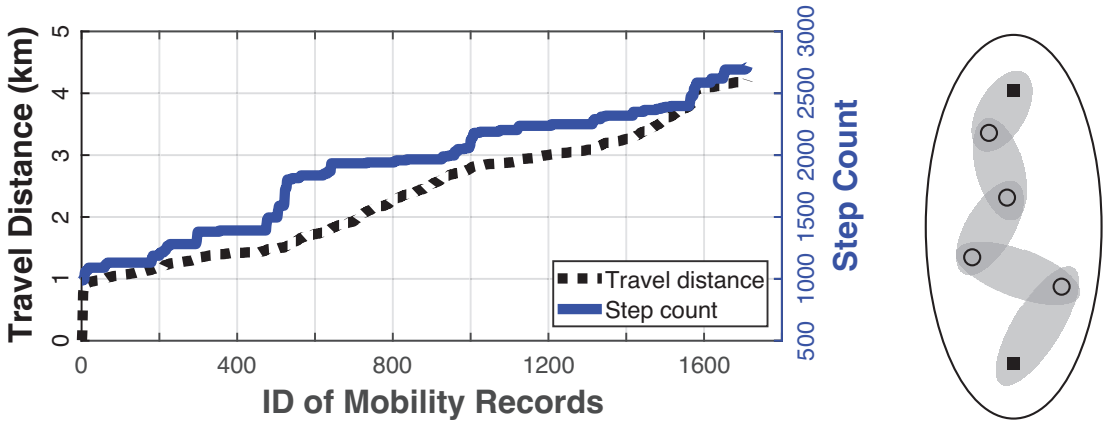
Figure 4.5: (a) The ellipse model based walkable area estimation *v.s.* individual sample based estimation; (b) Linear relationship between steps taken and sample distance.

Based on the results of *Location Data Classifier*, VitalAlley will identify and verify the representative walkways through several modules. Specifically, the *Walkable Area Estimation* module (in Section 4.2.2) approximates possible walkable areas, within which the *Walkable Path Identification* module (in Section 4.2.3) extracts some representative walkways. Finally, the *Auto-Verification* module (in Section 4.2.4) verifies the new-found walkways by analyzing their images retrieved through GSV APIs. All discovered walkaways are integrated into current road map for the public uses.

4.2.2 Walkable Area Estimation

A straightforward approach to deriving walkable area from coarse locations is to calculate the possible coverage of each mobility sample given a typical WiFi hotspots based localization error (*e.g.,* $80m$) in urban city and take the whole covered area as the walkable area estimation. Such an approach simply treats mobility data and fails to exploit the multi-modality sensor data and trajectory information between consecutive locations. Instead, we proposes an ellipse model to estimate the walkable area between any two consecutive mobility data, jointly leveraging the location and step count data.

Benefits of step counts. When considering the total steps taken from one location to the next, we can estimate the length of walking path between the two locations as the product of total steps and stride length. Such an estimation of walking distance along with the locations themselves together bound the possible walkable area, which form an elliptical region. Based



4.6.a: 4.6.b:

Figure 4.6: (a) The step counts and accumulated traveling distance of one student over a school day; (b) Fine-grained estimation derived from step count estimator, where ◯ denotes locations whose step counts remain and ■ denotes locations whose step counts increase.

on this observation, We thus propose an ellipse model to estimate the walkable area. In theory, an ellipse is determined by two parameters: 1) positions of two focal points; and 2) sum of distances d from every point on boundary of ellipse to the two focal points. The locations of two consecutive mobility data indicate the focal points and we determine d using the step count data in records. VitalAlley calculates d as $\Delta s \times \lambda$, where Δs is the total steps taken between the two locations (*i.e.,* difference of step counts in the two records) and λ is stride length and set as the average stride length of young as $0.78m$ according to a recent report [6]. Though the stride length can be accurately estimated for each individual, it requires extra efforts and hardware [48], which is infeasible in the NSE project. With such settings, VitalAlley can produce an oval area to capture the walkable area between two locations. Figure 4.5.a presents an example of the walkable area estimation from a series of unmatched mobility data. Compared with the disjoint walkable area estimation (*i.e.,* yellow area) derived from coverage of each individual location, the ellipse model can provide more reasonable walkable area estimation (*i.e.,* grey area).

Step estimator. Although step counts benefit the walkable area estimation, they are not perfect in the NSE mobility data. To reduce computation overhead of the SENSg device, step counts as the derived results are updated with a longer period than the updating period of locations. Figure 4.6.a plots the step count data and accumulated travel distance of a student over one typical school day, where travel distance is calculated from the locations. From this figure,

we can see the travel distance keeps increasing while the step count data are discontinuously updated, where flatten lines indicate that the step counts remain the same as previous one. Due to this fact, we can only use the records, whose step counts and locations are simultaneously updated, for walkable area estimation. As a result, the ellipse model can only provide coarser estimations using a few valid records.

To improve the performance of the ellipse model, we propose a step estimator which can estimate the total steps taken between two consecutive locations. We find an interesting relationship between the geographical distance of two consecutive locations and the total steps Δs between them. We extensively study their relationship and plot the statistics in Figure 4.5.b, which clearly shows that the distance of two consecutive locations and Δs are linearly correlated. Therefore, we can build a step estimator for each student by training a linear regression model based on her records with both step count data and location data available. Since the ellipse model is related to d which is determined by total steps Δs, we thus are able to determine an ellipse only based on the locations of two consecutive records even when their step counts are not updated. The step estimator enables us to derive more fine-grained walkable area estimation, just as demonstrated by gray region in Figure 4.6.b. We can only generate a large oval area with two records having both updated locations and step counts, while a fine-grained walkable area can be derived by exploiting the step estimator.

Weighting scheme. We further refine the walkable area estimation through a weighting scheme. For an oval area derived from two consecutive records, different parts inside the region may be traveled by pedestrians with varying probabilities. For example, when someone walks from one position to another, it is highly possible for her to walk along the straight line. Based on this intuition, we propose a weighting scheme to assess the probability of being traveled for different parts inside the oval area. To achieve that, we use a bivariate Gaussian model [106], which has been frequently used for home range estimation of animal movements in the biological domain [47]. Researchers in biological domain have limited periodical observations of where the animals are and they need to estimate the moving region of animals. They adopt this model since it can provide the probability of being a part of home range for a specific region. Similarly, we also have limited observations from pedestrians and need to estimate their possible movement area. In our scenario, the bivariate Gaussian model measures the probability density of an area being traveled by the pedestrians. We partition the space of interest into small cells and assess the probability of each cell inside an ellipse. We set the

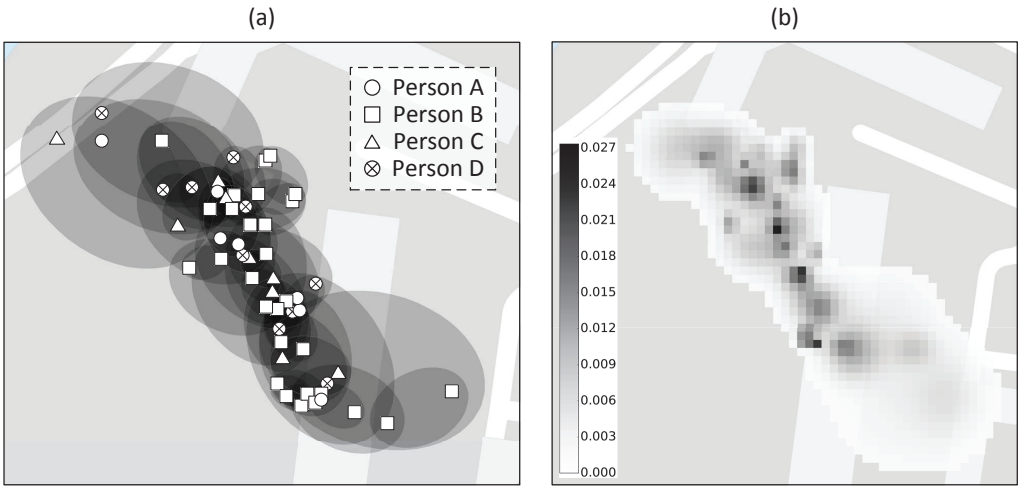


Figure 4.7: (a) The walkable area estimation from multiple trajectories. (b) The corresponding score map of (a).

cell size as $2.0m \times 2.0m$ for a better resolution. The probability density of being traveled by pedestrians for location X is measured by:

$$f(X) = \frac{1}{2\pi\sqrt{|\Sigma|}} exp(-\frac{1}{2}X^T\Sigma^{-1}X), \qquad \text{(Eq. 4.1)}$$

where $X = [x_1, x_2]^T$ and $\Sigma \in S^2_{++}$ is the covariance matrix of X. Σ is calculated as $\Sigma = [a/3, 0; 0, b/3]$, where a and b are major semi-axes and minor semi-axes of the ellipse edges. The probability of each cell being passed by the pedestrians is the integral of Eq. 4.1 on the cell's area. The probabilities of cells outside ellipse are negligible.

We can easily extend the scheme to all mobility data collected from many students. For any two consecutive records, VitalAlley generates an ellipse and assesses the probability of each cell inside the ellipse. An area covered by more ellipses should be walkable with a higher probability. In other words, these areas are "scored" by the mobility data via ellipses. For any cell, its final score is the sum of probabilities assessed by oval areas covering on it. A cell with larger score implies that it is more likely to be walkable. Finally we can derive a score map of the space of interest. Figure 4.7(a) presents the ellipses derived from the mobility data of four persons, and Figure 4.7(b) shows the corresponding score map based on the weighting scheme. We can see that the cells covered by more oval areas have larger scores, *i.e.,* in darker color.

4.2.3 Representative Walkway Identification

We propose a two-phase clustering to identify some representative walkways from each estimated walkable area. The representative walkways sketch the accessibility of a walkable area and usually connect certain road segments or POIs at some intersections for convenient mobility. Such intersections act as the entries/exits of a walkable area. Therefore, we conduct *location clustering* on some matched location data to identify the intersections, and then run *trajectory clustering* on unmatched location data to extract representative walkways that link the derived intersections. In the following, we present the details of each clustering phase.

Phase 1: location clustering to identify the intersections. To derive the entries/exits of a walkable area, we make use of the last matched locations collected near the walkable area. A last matched location data is the data that can be well matched with known road segments, after/before which other mobility data cannot be matched. We put the last matched data matching with the same road segment (or POI) together, which may correspond to one or several potential entries/exits. A walkable area may intersect with the same road segment/POI at several points, resulting in multiple entries/exits. Thus location clustering firstly groups the last matched data matching with the same road segment/POIs and then runs the HDBSCAN algorithm [15] on each location group to further classify those locations into one or several clusters, each of which indicates an intersection connecting one road segment/POI and the walkable area. We calculate the position of an intersection through all locations in the same cluster. It is the location on a road segment that has the minimum distance to all last matched locations of the cluster; or just as the average location when the location cluster falls in a POI, *e.g.,* residential building.

Figure 4.8(a-e) illustrates the location clustering. Figure 4.8(a-d) plot the mobility data collected from four persons A, B, C, and D, in the same area. Their unmatched locations are denoted by ○, □, △ and ◊, respectively. Meanwhile, their last matched locations are denoted by ●, ■, ▲ and ♦, and the corresponding matched locations are denoted by ★. Figure 4.8(e) shows that location clustering groups the last matched data into three location clusters, *i.e.,* "SC-1", "SC-2", and "SC-3", which correspond to three entries/exits of the walkable area. "SC-1" and "SC-3" locate on different road segments while "SC-2" locates on one residential building. Through the location clusters, we can calculate the positions of intersections.

Phase 2: trajectory clustering to extract representative walkways. The trajectory clustering takes advantage of the results from the first phase to group trajectories. In practice, each

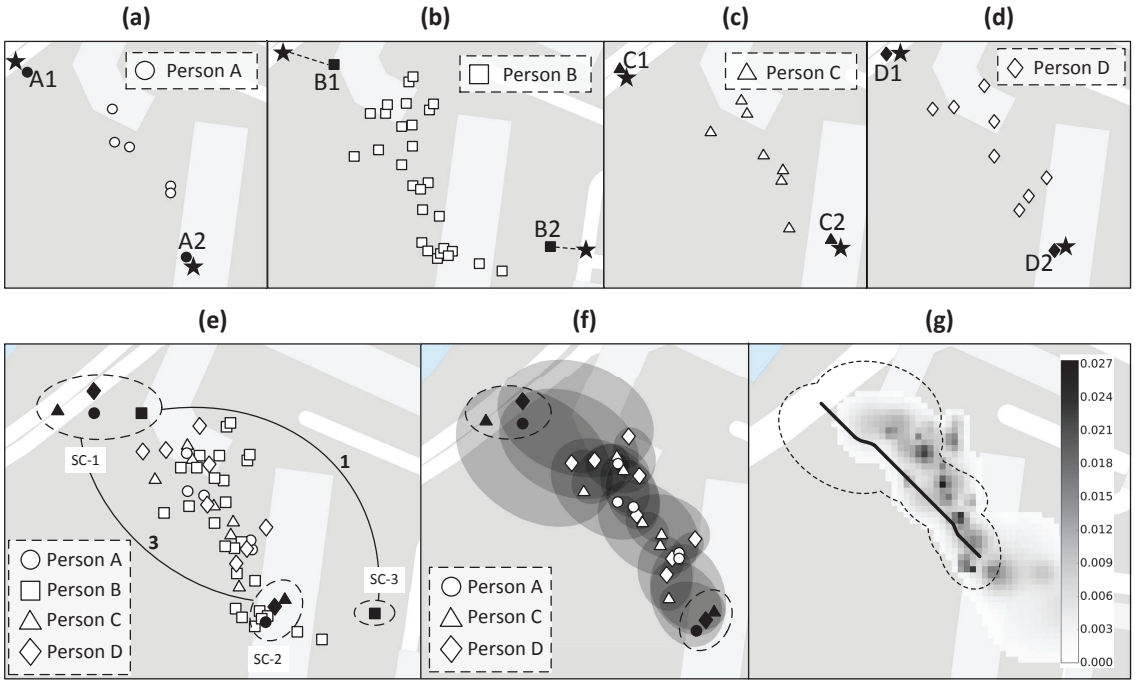


Figure 4.8: An illustrative example for representative walkway identification. (a-d) plot the mobility data of person A, B, C, and D, respectively. (e) demonstrates the two-phase clustering. (f) shows the covered area by ellipses of trajectory cluster <SC-1, SC-2>, and (g) presents the searching space and derived representative walkaway of trajectory cluster <SC-1, SC-2>.

trajectory is associated with two last matched data, and we thus can use their corresponding location clusters to annotate the trajectory. For example in Figure 4.8(b), location B1 and B2 are paired last matched data of trajectory B, and we annotate this trajectory as <SC-1, SC-3>. We annotate all trajectories following the same rule, and thus can simply group trajectories with the same annotation. Even for the same pair of intersections (*i.e.,* entries/exits), different trajectories exist and may correspond to different representative walkways. Thus we still need to run a typical clustering algorithm [98] to further classify the trajectories in the same group into different clusters according to the metric of Hausdorff distance. Each trajectory cluster corresponds to a representative walkway, and we use its size (named as *support*) to assess whether the potential walkway is frequently used or not. In Figure 4.8(e), there are two trajectory clusters and their supports are 3 and 1, respectively.

In principle, the potential representative walkway of a trajectory cluster should be a polyline on the score map of a walkable area, traversing the cells with high scores. To find the walkway

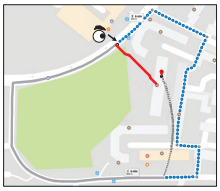
most likely to be traveled in reality, we transform the score map into a weighted graph and model our representative walkaway identification problem on this graph. For the score map of a walkable area, we generate a graph, where cells are represented as the vertices and edges are formed between any two immediately neighboring cells. In the graph, we set the weight of each vertex as the reciprocal of the score of corresponding cell. For cells with zero score, we set their weights as an maximum value, which implies that those cells are barely traveled by the pedestrians. Each vertex also inherits the position of the corresponding cell. In such a graph, we define a real-valued weight function $f : V \rightarrow \mathbb{R}$ that returns the score of a given vertex v_i, and we will find a representative walkaway $P = (v_s, v_1, v_2, \cdots v_n, v_d)$ between the two intersections v_s and v_d over all possible n vertices such that the total score of its constituent vertices $\sum_{i=1}^{n} f(v_i)$ is minimized. We can run the A^* algorithm to find the optimal solution. Finally, mapping vertices back to the cells on original map obtains the representative walkway.

Optimizations. VitalAlley also incorporates some optimizations to speedup the identification of representative walkways. For each trajectory cluster, we extract a subgraph from the original weighted graph by only keeping the vertices (and the associated edges) covered by oval areas of records belonging to this cluster. We thus reduce the searching space for a specific representative walkway. In addition, VitalAlley simply discards the trajectory clusters owning small supports, *e.g.,* $< \alpha$, just because they are barely traveled by people. For the example in Figure 4.8, we plot the covered area by ellipses of trajectory cluster <SC-1, SC-2> in Figure 4.8(f), which is derived from the unmatched trajectories of person A, C, and D. To identify the representative walkway, VitalAlley shrinks the searching space by filtering out irrelevant cells. The final searching space is shown as the dashed region in Figure 4.8(g). The black line in Figure 4.8(g) is the representative walkway, connecting a road segment and a residential building.

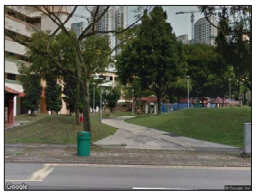
4.2.4 Auto-Verification

We propose an auto-verification method to verify the discovered walkways by invoking the GSV APIs [35]. GSV offers panoramic street views from different positions along the streets in a city, and we are able to retrieve images for a walkway at its entry/exit, which usually locates at some main road. As an independent data source, GSV provides a complementary angle to investigate new-found walkways.

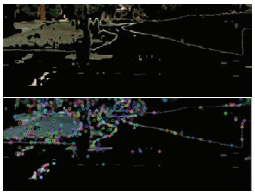
We find that the entries/exits of walkways in a city are relatively regular and pertain a



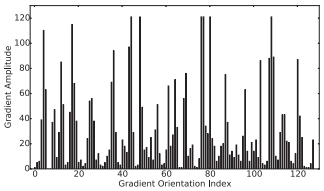
4.9.a: An example of one newfound representative walkway.



4.9.b: The GSV image captured from the position of intersection.



4.9.c: One of the clustered images and its key points for the image in (b).



4.9.d: One typical key point descriptor illustration.

Figure 4.9: The illustrations of image processing on the GSV images of one discovered walkway.

limited number of types. Thus it is possible for us to collect images of typical entries/exits from GSV in advance and treat them as the templates stored in a local library for the auto-verification of walkways.

Retrieving images of walkways from GSV. For each candidate walkway, we project its entry/exit (*i.e.,* the intersection derived from location clustering) to the main road and retrieve multiple images at the projected points by invoking GSV APIs, which need the parameters of location (*i.e.,* projected points), size (*i.e.,* 640×640), field of view, and heading directions. We take the projected point as the origin and generate two rays from this point to include all location points of the last matched data that generate the intersection. The two rays form an angle θ and we retrieve $\lceil \theta/30^\circ \rceil$ images from GSV by varying the heading direction every 30° within the angle. As a concrete example, Figure 4.9.a presents a new-found representative walkway that connects a road segment and a residential building, and Figure 4.9.b shows the retrieved GSV image from the position where there is an eye in Figure 4.9.a. From this GSV image, we can clearly see a walkway.

Image matching. For each walkway candidate, we extract the features (such as corner, boundary) from GSV images as the key descriptor using conventional image processing techniques [63]. For one GSV image, we first segment the image into κ clusters based on the pixel values, where we set $\kappa = 5$ for better performances. Second, we remove the upper part of the segmented image because walkways usually locate at the bottom part of an image. Then from the cut image, we extract the key points, which are around the borders of segmentations and describe the image [66]. Then we calculate the image gradient magnitudes and orientations of key points to form an image descriptor, which is denoted as a histogram. We match it with

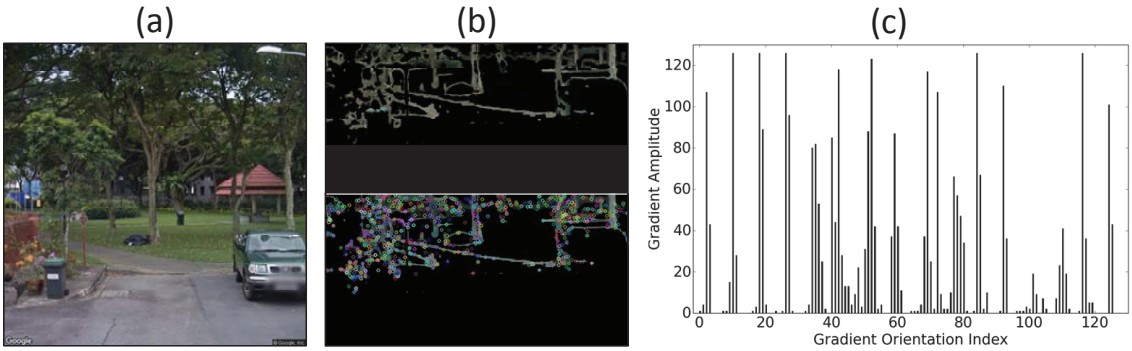


Figure 4.10: The matched template image in the library.

template images by comparing their image descriptors in χ^2 distance, which returns a measure of difference. Once the difference of at least one GSV image is lower enough, *e.g.,* $\leq \rho$, we declare that this candidate walkway is verified and mark it as "*confirmed*"; otherwise as "*suspicious*". We set $\rho = 10$ as the default setting in this work.

For the GSV image in Figure 4.9.b, we derive the image processing results shown in Figure 4.9.c and Figure 4.9.d. In Figure 4.9.c, the upper figure is the bottom part of one of the 5 clustered images and the bottom figure is the corresponding key points highlighted by circles. Figure 4.9.d shows the image descriptor of one key point in Figure 4.9.c, where bins on x-axis are the gradient orientations and y-axis indicates the amplitude of associated gradient. We match this image with the templates in the library and find one template image shown in Figure4.10(a). Its clustered bottom part and key points are shown in Figure 4.10(b). Figure 4.10(c) is a descriptor of one key point. After the calculations of their descriptors on the χ^2 distance, we find they have low difference with $\chi^2 < 10$, which indicates that the new-found walkway is verified and really exists.

Since GSV does not have images for the whole city, we may retrieve no image at some locations. For such cases, we mark the candidate walkways as "*suspicious*" for prudent use by the publics. We integrate all discovered walkways into existing road map and provide some hints to remind the users whether a walkway is verified or not.

Building the template library. We can build a local template library in advance for the auto-verification. Building a library containing all the templates is out of the problem due to infinite cases. We argue that, however, the types of different walkways in a specific city are limited so we only consider the most frequently used walkways. This library consists of four categories of walkways in total. Each category has many template images and the correspond-

ing image features, which describe the typical characteristics of entries/exits of walkways in a city. Specifically, we review the images of intersections between walkways and main streets from GSV and select some regular entries/exits to collect their images as the templates for building the local library. For each typical intersection, we capture 3 images from the street with heading directions at the medial axis of the walkaway and 30 degrees offset of the medial axis. For each image, we extract its image descriptor using the same image processing techniques. We can manually update this library by adding more GSV images of new kinds of entries/exits.

4.3 Evaluation

In this section, we implement VitalAlley and extensively evaluate its performances using the NSE mobility dataset. We make our source code publicly accessible through GitHub [96].

4.3.1 Experimental Setup

We implement VitalAlley in Python and run the system in a powerful HP Z440 workstation that has 12 3.5GHz Intel Xeon CPU cores and 32GB memory. For data preprocessing, we realize a custom HDBSCAN algorithm [15] to filter out the sojourn mobility data and implement an hidden Markov model based map matching algorithm [68] to accurately match mobility data with a base road map. We leverage some open-sourced Python libraries, *e.g.,* matplotlib, networkx, numpy, *etc.* to compute, visualize, and verify the results on a base road map, as well as capture the screenshots in this paper. We also implement the automatic verification module by invoking the GSV API [35]. We use the NSE mobility dataset and an open-sourced road map for the evaluations of VitalAlley.

NSE mobility dataset. The dataset contains mobility data collected from 108,337 students for 11 weeks in 2016 in the NSE project, containing more than 400 million records of totaling millions of kilometers in distance. As mentioned in Section 4.1.2, each record in the dataset includes 16 attributes about the mobility and environmental parameters during a student's daily activity. We mainly use the timestamps, step counts, and locations for walkway discovery.

Base road map. We obtain our base road map from the open-sourced OSM [73], which is the largest crowdsourced mapping project with more than 2 million registered contributors. In this paper, we use the OSM Singapore map of 20-August-2016 for map matching and eval-

Figure 4.11: All new-found walkways discovered from the NSE mobility data.

uations, which already includes many passable roads in Singapore. We complete this map by supplementing the missing walkways discovered from daily trajectories of local students. To accelerate the walkway discovery for the whole city, we divide the OSM Singapore map into four regions as shown in Figure 4.11, *i.e.,* $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, and $\mathscr{D}$, and run VitalAlley for each region in parallel using the multi-threading technique.

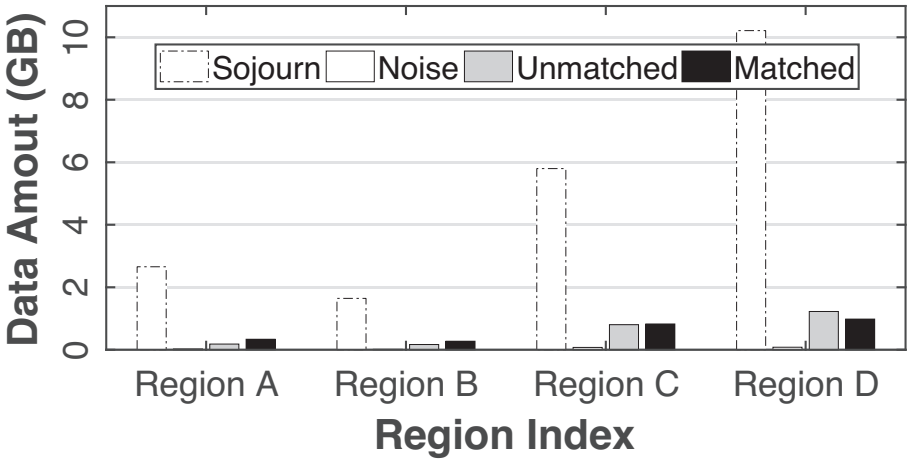


Figure 4.12: Statistics of data types in each region.

After the location data classification, VitalAlley divides mobility data into four types: *so-*

journ data that are sampled when students are staying at some specific place, *e.g.,* school or home; *noise* data that are with large estimated localization errors and thus far away from the student's trajectory; *matched* data that can well match with the OSM road map; and *unmatched* data that cannot well match with existing roads. Since the schools and homes of students are non-uniformly distributed in Singapore, the records are dispersed in the map. Figure 4.12 presents the concrete proportions of each data type collected in each region. According to our statistics, the average portion of the four data types (*i.e.,* sojourn, noise, unmatched, and matched) are 80.1%, 0.8%, 8.6%, and 10.5%, respectively. For all regions, the sojourn data occupy the largest portion, which is reasonable as students spend most of their time at school and home. For VitalAlley, unmatched data are the most valuable input, which provides implicit information about the missing walkways. The region $\mathscr{D}$ owns the most unmatched mobility data, where we thus may discover more walkways.

Performance metric. We define $accuracy = \frac{N_{true \bigcap new}}{N_{new}}$, where $N_{true \bigcap new}$ denotes the number of new-found walkways that" truly exist and N_{new} means the total number of walkways discovered by our system. The larger *accuracy* is, the better performances VitalAlley has.

4.3.2 Evaluation Results

In this subsection, we first present and analyze the overall results of walkway discovery on the whole OSM Singapore road map, and then conduct a detailed evaluation in a study area to carefully evaluate the accuracy of VitalAlley and the effects of system parameters.

Overall performances

Figure 4.11 depicts all the new-found walkways in Singapore by leveraging the NSE mobility dataset. The discovered walkways scatter over the road map and connect the motorways and POIs. In total, we have discovered 736 walkways, with accumulated distance more than 150km. The number of discovered walkways for region $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, and $\mathscr{D}$ are 110, 109, 147, and 370, respectively. In principle, a region with more mobility data will probably have more discovered walkways. In region $\mathscr{D}$ which is near the downtown area of Singapore, we have the most new-found walkaways as we have the most mobility data there. From Figure 4.11, we see the walkways are in different lengths, and we present the statistics of lengths of the new-found walkways in Figure 4.13. Almost all of the new-found walkways are with lengths less than 1km, and 90% of the walkways are shorter than 598m. This is reasonable as walkways usually

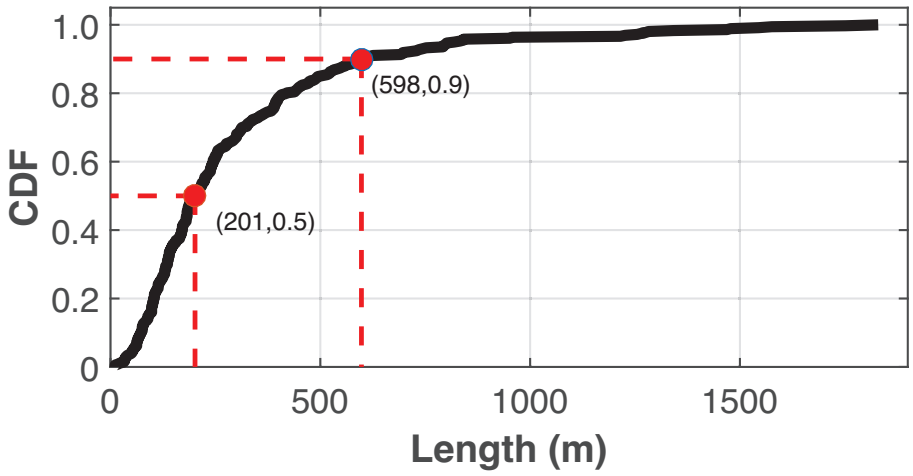


Figure 4.13: The length distribution of all new-found walkways.

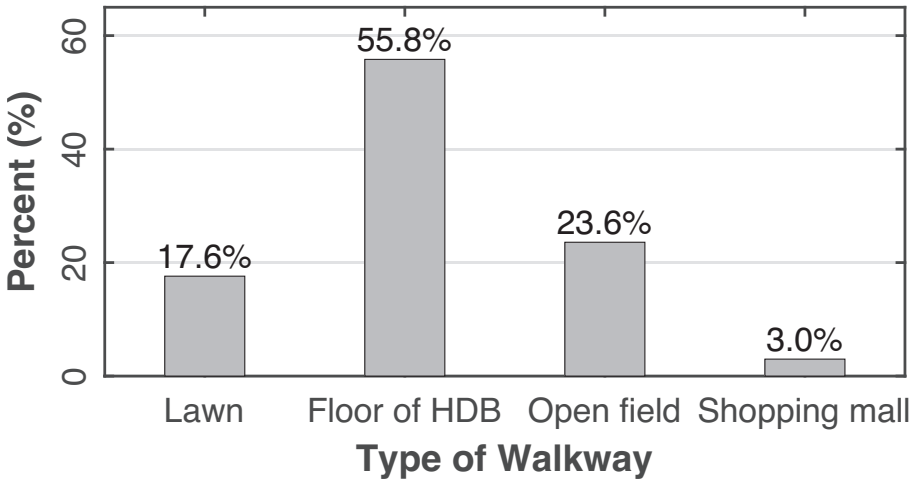


Figure 4.14: Different types of all discovered walkways.

act as the small shortcuts between road segments/POIs. According to our statistics, the shortest walkway is only 11 m, while the longest walkway achieves 1829 m.

For each discovered walkway, we invoke the GSV APIs to retrieve its images for auto-verifications, and also check the type each walkway belongs to. Among all the new-found walkways, we mainly have four types of walkways, namely lawn, floor of HDB (the typical residential buildings in Singapore), open field and shopping mall, which account for 17.6%, 55.8%, 23.6% and 3.0%, respectively, as the statistics shown in Figure 4.14.

Walkways derived from the floors of HDB account for the largest proportion mainly be-

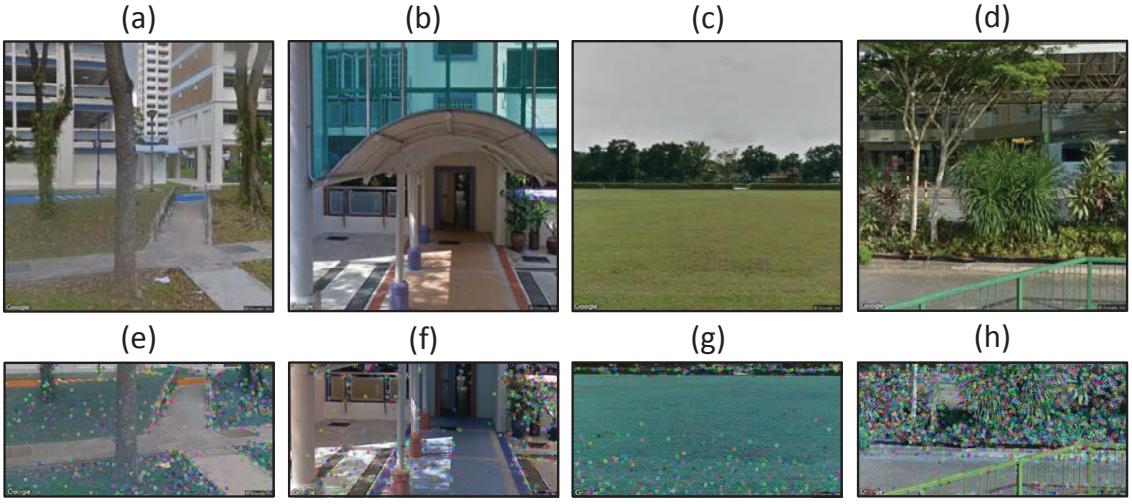


Figure 4.15: GSV images of the new-found walkways.

cause the HDB building widely exist in Singapore and many of such residential buildings have the first floor empty where people can walk through. Besides, the lawns and open fields are also frequently used by the local residents as the shortcuts for accessing to other neighborhood. The interiors of shopping malls are also occasionally used as walkways by some students who are familiar with the inner connectivities of those malls. Figure 4.15 shows the images of some typical walkways we find in Singapore. Figure 4.15 (a) plot the walkways in the lawn, where we can see some clear walking tracks used by the pedestrians. Figure 4.15 (b) shows the walkway around the HDB buildings, which is a small trail connecting a road segment and the HDB residential building. Figure 4.15 (c) shows a walkway in an open field which is a flat ground, and Figure 4.15 (d) shows a walkway within a shopping mall, which connects two parallel road segments. Figure 4.15 (e-h) illustrate the key points of each type of walkways above. The features of walkway in shopping mall are complicated, as shown in Figure 4.15 (h). We identify walkways passing through shopping malls by exploiting additional information (*e.g.,* positions and names) from OSM. Such walkways widely exist in Singapore and facilitate people's daily mobility.

Accuracy evaluation in a study area.

To evaluate the accuracy of our system, we select a small region, as shown in Figure 4.16, around the central area of Singapore (*i.e.,* the rectangle region in Figure 4.11) covering about $9\,km^2$ area to conduct a detailed study. Within this region, we have discovered 224 walkways in total. For each new-found walkway in the study area, we manually investigate whether it

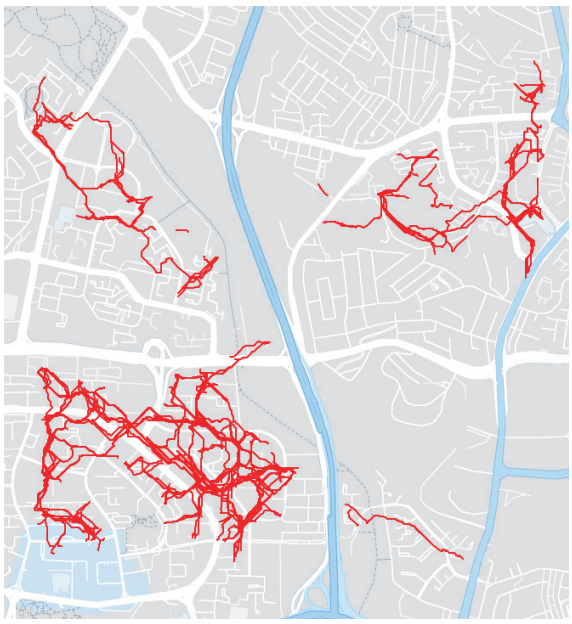


Figure 4.16: The new-found walkways in study area shown as black box in Figure 4.11.

is true. If we are able to find a walkway according to the positions calculated by our system and the walkway indeed connects two road segments/POIs indicated by our location clustering, then we consider this walkway as true in reality; otherwise we think it is a false walkway and cannot be practically used by the pedestrians. For this specific study area, we finally find 209 true walkways with manual verifications, with the accuracy as high as 93%.

We explore the accuracies for walkways of different lengths and present the results in Figure 4.17. Specifically, we classify all walkways into four groups according to their lengths (in m), *i.e.,* $(0,200]$, $(200,400]$, $(400,600]$, and $(600,\infty)$. Among the four groups, we find a large number of walkways are with short length, *i.e.,* $\leq 200m$. The distribution of walkways on length is in accordance with the overall distribution in Figure 4.13. Most walkways are in median lengths and there are few walkways too long, *e.g.,* > 600. With respect to accuracy, group $(200,400]$ has the lowest accuracy as 88%, and group $(0,200]$ has the highest accuracy as 94%.

We also study the accuracies for different types of walkway, and present the results in Figure 4.18. For lawns and shopping malls, VitalAlley achieves 100% accuracy, which means that all the new-found walkways on lawns and shopping malls are verified as truly "walkable" after manual investigations. For walkways on lawns, the auto-verification module can accurately

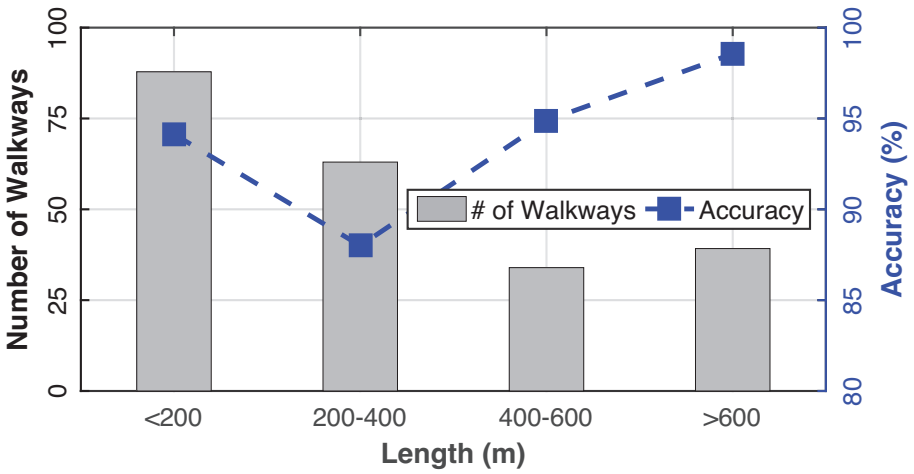


Figure 4.17: Accuracies for walkways of different lengths.

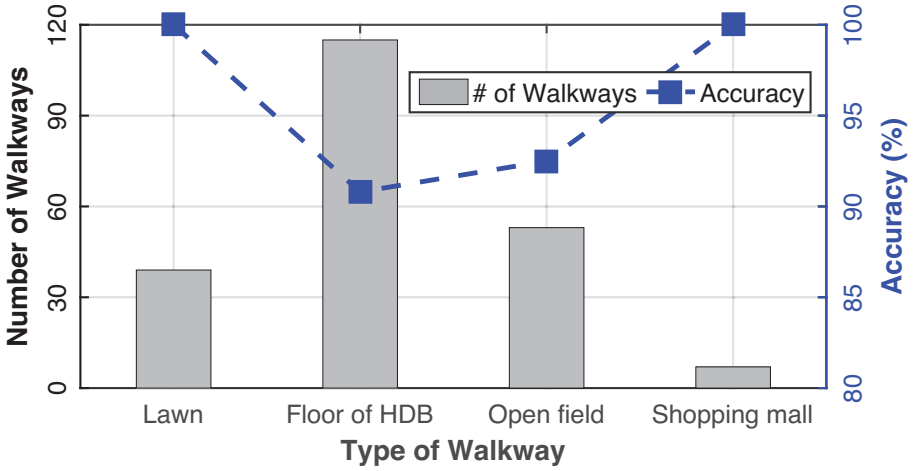


Figure 4.18: Accuracy of different types of walkways.

identify them as the features of lawns are prominent in images. Besides, our base road map contains the information about shopping malls and once a walkway passing through a shopping mall can be easily identified. As long as those features are extracted, it is highly possible that the walkways endorsed by the GSV-based auto-verification module exist in reality. For walkways belonging to types of "floor of HDB" and "open field", VitalAlley still achieves accuracies of as 91% and 93%, respectively.

Utility study of discovered walkways. The motivation for walkway discovery is that those widely existing walkways can provide significant convenience and largely save the walking time when compared to using existing road segments. Thus we conduct a utility study for the

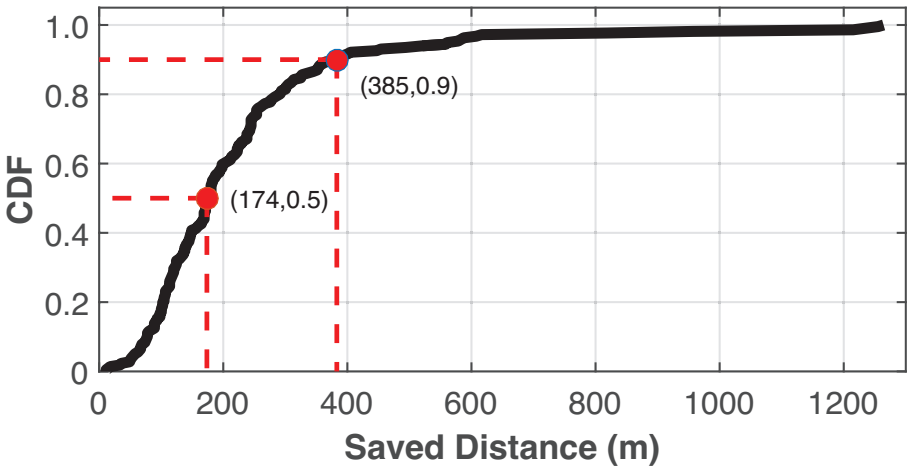


Figure 4.19: Utility of walkways for route planning.

discovered walkways in the study area. We initiate 100 travel demands with trip origins and destinations being the students' homes and schools. We implement a shortest path based route planning algorithm, which will return a shortest walking route based on the available road map. For each pair of trip origin and destination, we generate two routes based on the original OSM road map and the new road map supplemented with our new-found walkways. We calculate the distance difference between the two routes and adopt the saved walking distance when compared to the one planned on the original road map as the measure of utility. The larger the saved walking distance is, the more efficiency the walkways can provide. We calculate the distance differences of the 100 trip queries, and plot the statistics in Figure 4.19. For all trip planing queries, the routes derived from the completed road map are shorter than the ones walking on existing roads. Among all queries, 50% can save walking distance than $174\,m$, and the 90-percentile saved walking distance is $385\,m$. The results in Figure 4.19 demonstrate that it is necessary to discover those missing walkways as they really facilitate people's mobility.

Parameter Setting

We evaluate the impacts of system parameters and design choices in the study area.

Impacts of support value. In the representative walkway identification module, we use a threshold to filter out some trajectory clusters with small support. In fact, the support reflects how likely a trajectory cluster is to generate a representative walkway used by enough pedestrians. Figure 4.20 shows the effects of choosing different supports as the threshold on the

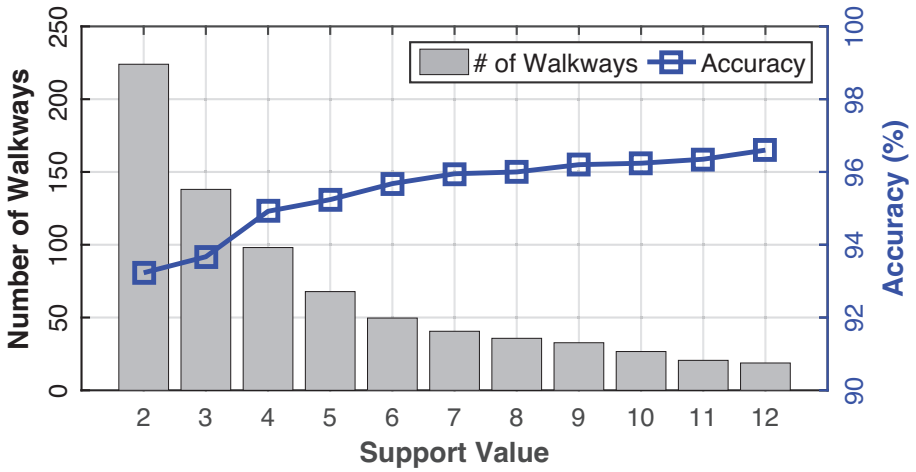


Figure 4.20: The impacts of support value on number of discovered walkways and accuracy.

number of discovered walkways and accuracy. The support value is ranged from 2 to 12. When $support = 2$, nearly 224 walkways are correctly found in the specific region with an accuracy of 93%. Larger support requires a walkway traveled by more pedestrians and thus leads to a decreased number of discovered walkways. On the other hand, the accuracy increases along with increase of support values. Only 31 walkways are found with a high accuracy as 97% when we set $support = 12$. According to Figure 4.20, we find $support = 3$ can balance both number of discovered walkways and the accuracy, and we thus set $support = 3$ as the default setting.

Impacts of data amount used. The data amount used will also affect the effectiveness of walkway discovery. With more data, we can collect the observations about more potential walkways and it is more possible to accurately identify them. We vary the number of weeks of mobility data used, and present the resulting number of discovered walkways and accuracy in Figure 4.21. When we increase the amount of used mobility data, the number of discovered walkways gradually increase as well. It is reasonable since more mobility data will include the trajectories from more walking paths and we thus can identify more walkways. When we use data more than 9 weeks, the benefit becomes negligible as the frequently used walkways have been covered by existing data and extra data may provide little new knowledges. On the other hand, the impact of data amount on the accuracy is less significant. More data lead to slight increase on the accuracy. When we use more than 6 weeks of data, the accuracy becomes stable around 98.5%. Extra data bring no increase on accuracy. Therefore, 9 weeks of mobility data

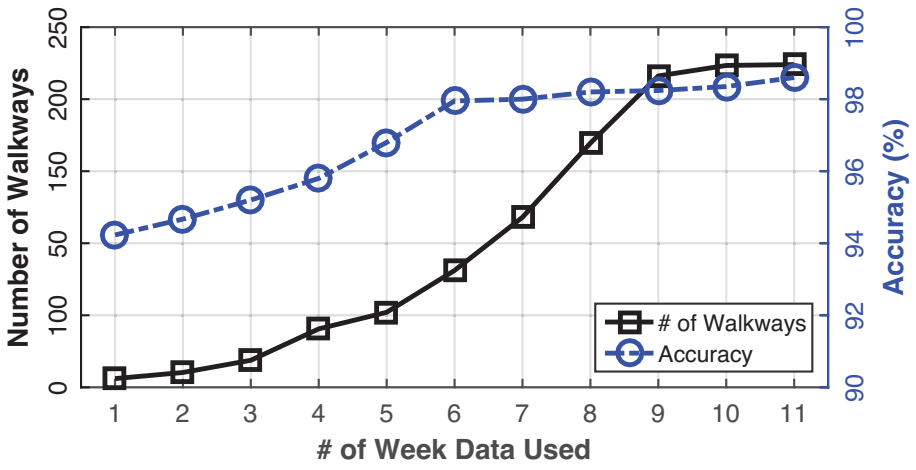


Figure 4.21: Impacts of data amount used on the number of discovered walkways and accuracy.

seem to be a good amount for accurate and comprehensive walkway discovery in our situation.

Table 4.1: Effect of GSV based auto-verification on the accuracy of walkway discovery.

Support	2	4	6	8
w/ GSV	93.2%	94.8%	95.7%	96.0%
w/o GSV	80.9%	88.6%	93.5%	95.8%

Impacts of GSV based auto-verification. We invoke the GSV APIs to retrieve images of the discovered walkways and automatically verify them based on computer vision techniques. To evaluate the performance of this module, we conduct experiments *with* and *without* the GSV-based auto-verification module. We use 9 weeks of mobility data during the comparison and vary support from 2 to 8. The results are shown in Table 4.1. Note that for both with and without this module, the system can find the same number of walkways. By enabling this module, the system can always achieve higher accuracy than the case without this function, having the largest performance gain of 12.3% on accuracy when $support = 2$. When we increase the support value, the gain benefited from GSV becomes small as the parameter of support value will filter out many invalid walkways.

Impacts of step estimator. In the system design, we propose a step estimator to enhance the walkable area estimation. Thus we also run experiments based on only available step count data (denoted as *basic method)* and enabling the step estimator (denoted as *enhanced method*). Table 4.2 shows the comparison results. For both methods, they achieve similar accuracy while

Table 4.2: Effect of step estimator.

	# of walkways	Accuracy
Basic method	95	93%
Enhanced method	224	94%

the enhanced method still has a bit higher accuracy. On the other hand, VitalAlley can discover much more walkways with the step estimator when compared using the step count data only, *i.e.,* 224 *vs* 95 with an improvement of 136%. Since the step count data in the NSE project may not be continuously updated, thus some mobility data cannot be directly exploited by the basic method. Once the step estimator is trained for a student, all the mobility data can be used by the enhanced method and as a result more walkways are found.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Smart city is an inevitable trend for city development, with more and more electrically mobile devices being available on the market. Many countries have planned to build smart cities. Among all the related components of a smart city, transportation is vitally important to a city. In this thesis, we describe two possible directions to enhance the transportation systems from the people's points of view. For the navigation service, we provide an AR-based one which could improve the user experience. We also leverage the crowdsensed mobility information to complete the missing information for digital maps.

We propose Amateur in the first work. It is an augmented reality based vehicle navigation system, using commodity smart phones. Amateur can automatically display navigation instructions as a series of annotative arrows on the live road condition video stream to assist the driving. The system design encounters host-lane identification and intersection determination challenges. We observe that the lane markers are painted in yellow or white in reality, leading to large values in gray scale domain. Taking those large values that look like peaks as input to a customized particle filter, the host-lane is identified in a computing affordable way for mobile platform. We leverage the flickering feature of LED bulbs to detect traffic lights in video streaming. The position of traffic lights in the video is a good indicator of the position of intersections in the real world. Our solution can comfortably work with lightweight smart phone platforms. To demonstrate the efficacy of the Amateur design, we implement the system on Android phones. The experimental results indicate that Amateur can improve the navigation

service from four important aspects: ease of use, perceived distraction, navigation experience and user-friendliness.

We build another system named VitalAlley for discovering and verifying the missing walkways leveraging NSE mobility data. Compared to motorways where vehicles are constrained by the road, walkways are not structured and pedestrians have higher freedom to move towards any direction. VitalAlley proposes an ellipse model to estimate the range of walkable area and a novel bivariate Gaussian model based weighting scheme to assess the workability of walkable areas. We then identify representative walkways from each walkable area through a two-phase clustering method. Another challenge for new-found walkways is the verification. No previous method has been proposed to evaluate the validity of new-found motorways or walkways. We proposed to resort to the help of an independent data source – GSV. The GSV is involved to automatically verify each new-found walkways by comparing the representative features of intersections between walkways and main roads. Experimental results using the large-scale NSE mobility data demonstrate that our system VitalAlley finds 736 valid walkways (totaling 161 km in distance) for the OSM Singapore road map. A detailed evaluation in a study area shows that VitalAlley can achieve accuracy as high as 96%.

5.2 Future Work

Based on our studies about smart city and urban computing, we think we are going to keep researching on the following directions.

Demand-oriented bus dispatch strategy. Our project focus on the improvement of user-experience and digital maps of the city, which is one aspect of the smart cities. Bus dispatch strategy is another critical aspect for smart cities with large population density. Usually, bus company has a regularly planned time schedule to run buses but the travel demand is not considered in the time schedule, leading to bus bunching [7] phenomenon that two same buses run together and one is almost empty. The main reason may be that the travel demand is less than the carriage capacity of provided buses. The travel demand could be analyzed through EZ-link cards that are the cards used to take public transportation in Singapore [13]. We may design more efficient and cost-effective strategy to improve the usage of buses by taking account the travel demand of commuters. In our vision, the bus scheduling should be adapted and matched with the travel demand.

Transportation efficiency analysis. Automatic fare collection devices are deployed in the public transportation system, namely subway system [59], bus system [9] and taxi system [119], each of which plays a key role in transportation efficiency. The performance of any of the three systems has great impact on transportation efficiency. In normal situation, each system takes corresponding travel demand but the travel demand may tremendously increase when there are emergent events such as subway breakdowns as subway system carries majority travel demand in daily commuting. We may resort to the bus system to take the extra travel demand. Buses and subways have complementary characteristics of capacities, speeds and costs [117] and for some important road segments, both bus line and subway line exist. Thus, providing special-designed bus lines is a feasible solution for subway breakdowns. We wish to analyze the effect of subway breakdowns at different stations. The breakdown will generate new travel demand, based on which we have further optimized bus dispatch strategy. Thus the breakdown effect may be weakened by proper bus arrangement, which will largely shorten the waiting time of commuters affected by the breakdown.

Traffic monitoring through taxi information. In smart cities, we can provide better service by combining more information from connected devices. For instance, we could provide accurate waiting time for users if the traffic speed is known. Taxis are good "devices" equipped with on-board GPS chips [32]. We can regard taxis as probing sensors of the traffic speed. From the dataset of taxi, we could estimate the speed of taxis [65]. Thus, we have a sampling of traffic speed for the whole city. We will derive methods to infer the traffic speed of the road segments without taxis running on. The commuters may have a better plan and time schedule before they are waiting at the picking spots or bus stops.

Multi-source datasets. We have involved multi-source datasets in the design of VitalAlley. GSV dataset, as an independent dataset, is leveraged to verify the existence of new-found walkways. Obviously, multi-source datasets could provide complementary information comparing single-source dataset. To infer the mobility of human beings, mPat [118] leverages CDR data to compensate the missing part of transit dataset. We are going to leverage the information of EZ-link dataset, taxi dataset and NSE dataset to approximate the transaction of commuters. Commuter transaction is essential for building efficient wireless protocols and mobile applications. The EZ-link dataset, however, only covers partial commuters who take public transportation systems: subway and bus system. Taxi dataset only covers the commuters taking taxis. The transactions of human beings who driver their own vehicles cannot be estimated. We

believe some of them driver their kids to school in the morning, resulting in a overlap with NSE dataset. Using NSE dataset to infer the transactions of private vehicles is a feasible solution.

Appendix A

Author's Publications

(i) **Chu Cao**, Zhenjiang Li, Pengfei Zhou, Mo Li, "Amateur: Augmented Reality based Vehicle Navigation System ", in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* Volume 2, Issue 4, December 2018. To be presented in *ACM International Joint Conference on Pervasive and Ubiquitous Computing* 2019 (UbiComp'19).

(ii) **Chu Cao**, Zhidan Liu, Mo Li, Wenqiang Wang, Zheng Qin, "Walkway Discovery from Large Scale Crowdsensing ", in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2018.

(iii) **Chu Cao**, Zhidan Liu, Mo Li, Wenqiang Wang, Zheng Qin, "Demo: Walkway Discovery from Large Scale Crowdsensing ", in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2018.

References

[1] E. Abbott and D. Powell. Land-vehicle navigation using gps. *Proceedings of the IEEE*, 87(1):145–162, 1999.

[2] G. Agamennoni, J. I. Nieto, and E. M. Nebot. Robust inference of principal road paths for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):298–308, 2011.

[3] H. Alt, A. Efrat, G. Rote, and C. Wenk. Matching planar maps. *Journal of Algorithms*, 49(2):262–283, 2003.

[4] M. Aly. Real time detection of lane markers in urban streets. In *Proceedings of IEEE IV*, pages 7–12, 2008.

[5] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, (4):355–385, 1997.

[6] T. V. Barreira, D. A. Rowe, and M. Kang. Parameters of walking and jogging in healthy young adults. *International Journal of Exercise Science*, 3(1), 2010.

[7] J. J. Bartholdi III and D. D. Eisenstein. A self-coördinating bus route to resist bus bunching. *Transportation Research Part B: Methodological*, 46(4):481–491, 2012.

[8] D. R. Berryman. Augmented reality: a review. *Taylor & Francis Medical reference services quarterly*, (2):212–218, 2012.

[9] S. Bhattacharya, S. Phithakkitnukoon, P. Nurmi, A. Klami, M. Veloso, and C. Bento. Gaussian process-based predictive modeling for bus ridership. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1189–1198. ACM, 2013.

[10] J. Biagioni and J. Eriksson. Inferring road maps from GPS traces: survey and comparative evaluation. *Transportation Research Board Annual*, 2012.

[11] J. Biagioni and J. Eriksson. Map inference in the face of noise and disparity. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 79–88. ACM, 2012.

[12] M. Billinghurst, A. Clark, G. Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, (2-3):73–272, 2015.

[13] D. Bo and P.-A. Dublanche. Bus bunching identification using smart card data. In *Proceedings of The 24th International Conference on Parallel and Distributed Systems*. IEEE, 2018.

[14] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st international conference on Very large data bases*, pages 853–864. VLDB Endowment, 2005.

[15] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):5, 2015.

[16] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[17] C. Cao, Z. Li, P. Zhou, and M. Li. Amateur: Augmented reality based vehicle navigation system. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):155, 2018.

[18] C. Cao, Z. Liu, M. Li, W. Wang, and Z. Qin. Walkway discovery from large scale crowdsensing. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 13–24. IEEE Press, 2018.

[19] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic. Augmented reality technologies, systems and applications. *Springer Multimedia tools and applications*, (1):341–377, 2011.

[20] S. S. Chawathe. Segment-based map matching. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 1190–1197. IEEE, 2007.

[21] T.-S. Dao, K. Y. K. Leung, C. M. Clark, and J. P. Huissoon. Markov-based lane positioning using intervehicle communication. *IEEE Transactions on Intelligent Transportation Systems*, (4):641–650, 2007.

[22] A. David and P. Jean. Computer vision: a modern approach. 2002.

[23] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[24] M. Diaz, P. Cerri, G. Pirlo, M. A. Ferrer, and D. Impedovo. A survey on traffic light detection. In *Proceedings of Springer ICIAP*, pages 201–208, 2015.

[25] J. Duribreux, R. Rouvoy, and M. Monperrus. An energy-efficient location provider for daily trips. 2014.

[26] Exploride. Exploride website, Jan. 2018.

[27] E. Foxlin, T. Calloway, and H. Zhang. Improved registration for vehicular ar using auto-harmonization. In *Proceedings of IEEE ISMAR*, pages 105–112, 2014.

[28] P. Froehlich, R. Schatz, P. Leitner, S. Mantler, and M. Baldauf. Evaluating realistic visualizations for safety-related in-car information systems. In *Proceedings of ACM CHI*, pages 3847–3852, 2010.

[29] P. Fröhlich, M. Baldauf, M. Hagen, S. Suette, D. Schabus, and A. L. Kun. Investigating safety services on the motorway: the role of realistic visualization. In *Proceedings of ACM AutomotiveUI*, pages 143–150, 2011.

[30] P. Fröhlich, R. Schatz, P. Leitner, M. Baldauf, and S. Mantler. Augmenting the driver's view with realtime safety-related information. In *Proceedings of ACM AH*, page 11, 2010.

[31] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.

[32] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, and R. Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal—The International Journal on Very Large Data Bases*, 20(5):695–719, 2011.

[33] J. Gong, Y. Jiang, G. Xiong, C. Guan, G. Tao, and H. Chen. The recognition and tracking of traffic lights based on color segmentation and camshift for intelligent vehicles. In *Proceedings of IEEE IV*, pages 431–435, 2010.

[34] Google. Google maps, Aug. 2017.

[35] Google. Google street view, Aug. 2017.

[36] Google. Google i/o event, May 2018.

[37] Google. Google maps app, Jan. 2018.

[38] Google. Google maps direction api, Jan. 2018.

[39] J. S. Greenfeld. Matching gps observations to locations on a digital map. In *81th annual meeting of the transportation research board*, volume 1, pages 164–173, 2002.

[40] B. Guo, H. Chen, Z. Yu, X. Xie, S. Huangfu, and Z. Wang. Fliermeet: Cross-space public information reposting with mobile crowd sensing. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 59–62. ACM, 2014.

[41] N. Haderer, R. Rouvoy, and L. Seinturier. A preliminary investigation of user incentives to leverage crowdsensing activities. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 199–204. IEEE, 2013.

[42] A. B. Hillel, R. Lerner, D. Levi, and G. Raz. Recent progress in road and lane detection: a survey. *Springer Machine Vision and Applications*, (3):727–745, 2014.

[43] Hudify. Wayray website, Jan. 2018.

[44] B. Hummel and K. Tischler. Robust, gps-only map matching: Exploiting vehicle position history, driving restriction information and road network topology in a statistical framework. In *GIS Research UK Conference (GISRUK)*, pages 68–77, 2005.

[45] F. Inc. Foursquare, 2019.

[46] P. Jain, J. Manweiler, and R. Roy Choudhury. Overlay: Practical mobile augmented reality. In *Proceedings of ACM MobiSys*, pages 331–344, 2015.

[47] R. Jennrich and F. Turner. Measurement of non-circular home range. *Journal of Theoretical Biology*, 22(2):227–237, 1969.

[48] Y. Jiang, Z. Li, and J. Wang. Ptrack: Enhancing the applicability of pedestrian tracking with wearables. In *2017 IEEE 37th International Conference on Distributed Computing Systems*, pages 2193–2199. IEEE, 2017.

[49] Y. Jiang, H. Qiu, M. McCartney, G. Sukhatme, M. Gruteser, F. Bai, D. Grimm, and R. Govindan. Carloc: Precise positioning of automobiles. In *Proceedings of ACM SenSys*, pages 253–265, 2015.

[50] R. Jose, G. A. Lee, and M. Billinghurst. A comparative study of simulated augmented reality displays for vehicle navigation. In *Proceedings of ACM OzCHI*, pages 40–48, 2016.

[51] S. Jung, J. Youn, and S. Sull. Efficient lane detection based on spatiotemporal images. *IEEE Transactions on Intelligent Transportation Systems*, (1):289–295, 2016.

[52] S. Karagiorgou and D. Pfoser. On vehicle tracking data-based road network generation. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 89–98. ACM, 2012.

[53] C. Kaufmann, R. Risser, A. Geven, and R. Sefelin. Effects of simultaneous multi-modal warnings and traffic information on driver behaviour. In *Proceedings of European Conference on Human Centred Design for Intelligent Transport Systems*, pages 33–42, 2008.

[54] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of ACM MobiSys*, pages 127–140, 2011.

[55] I. Koutsopoulos. Optimal incentive-driven design of participatory sensing systems. In *2013 Proceedings IEEE INFOCOM*, pages 1402–1410. IEEE, 2013.

[56] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of ACM MobiCom*, pages 447–458, 2014.

[57] O. Kutter, A. Aichert, C. Bichlmeier, J. Traub, S. Heining, B. Ockert, E. Euler, and N. Navab. Real-time volume rendering for high quality visualization in augmented reality. In *International Workshop on Augmented environments for Medical Imaging including Augmented Reality in Computer-aided Surgery (AMI-ARCS 2008), New York, USA*, 2008.

[58] N. D. Lane, Y. Chon, L. Zhou, Y. Zhang, F. Li, D. Kim, G. Ding, F. Zhao, and H. Cha. Piggyback crowdsensing (pcs): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 7. ACM, 2013.

[59] N. Lathia and L. Capra. How smart is your smartcard?: measuring travel behaviours, perceptions, and incentives. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 291–300. ACM, 2011.

[60] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, pages 707–710, 1966.

[61] H. Li, L. Kulik, and K. Ramamohanarao. Automatic generation and validation of road maps from gps trajectory data sets. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1523–1532. ACM, 2016.

[62] Y. Li, Q. Huang, M. Kerber, L. Zhang, and L. Guibas. Large-scale joint map matching of gps traces. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 214–223. ACM, 2013.

[63] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceeding*

of the 11th annual international conference on Mobile systems, applications, and services, pages 69–82. ACM, 2013.

[64] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, and Y. Zhu. Mining large-scale, sparse gps traces for map inference: comparison of approaches. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 669–677. ACM, 2012.

[65] Z. Liu, S. Jiang, P. Zhou, and M. Li. A participatory urban traffic monitoring system: the power of bus riders. *IEEE Transactions on Intelligent Transportation Systems*, 18(10):2851–2864, 2017.

[66] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[67] R. Malaka, K. Schneider, and U. Kretschmer. Stage-based augmented edutainment. In *Springer International Symposium on Smart Graphics*, pages 54–65, 2004.

[68] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343. ACM, 2009.

[69] NSE.SG. National science experiment, Aug. 2017.

[70] W. Y. Ochieng, M. Quddus, and R. B. Noland. Map-matching in complex urban road networks. *Revista Brasileira de Cartografia*, 2(55), 2003.

[71] M. Omachi and S. Omachi. Traffic light detection with color and edge information. In *Proceedings of IEEE ICCSIT*, pages 284–287, 2009.

[72] T. O. Oshin, S. Poslad, and A. Ma. Improving the energy-efficiency of gps based location sensing smartphone applications. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 1698–1705. IEEE, 2012.

[73] OSM. Open street map, Jan. 2018.

[74] O. Palinko, A. L. Kun, Z. Cook, A. Downey, A. Lecomte, M. Swanson, and T. Tomaszewski. Towards augmented reality navigation using affordable technology. In *Proceedings of ACM AutomotiveUI*, pages 238–241, 2013.

[75] A. Pereira, G. A. Lee, E. Almeida, and M. Billinghurst. A study in virtual navigation cues for forklift operators. In *Proceedings of IEEE SVR*, pages 95–99, 2016.

[76] W. Piekarski and B. Thomas. Arquake: the outdoor augmented reality gaming system. *Communications of the ACM*, (1):36–38, 2002.

[77] O. Pink and B. Hummel. A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 862–867. IEEE, 2008.

[78] S. V. Ramani and Y. N. Tank. Indoor navigation on google maps and indoor localization using rss fingerprinting. *arXiv preprint arXiv:1405.5669*, 2014.

[79] C. M. Richard, R. D. Wright, C. Ee, S. L. Prime, Y. Shimizu, and J. Vavrik. Effect of a concurrent auditory task on visual search performance in a driving-related image-flicker task. *SAGE Human Factors: The Journal of the Human Factors and Ergonomics Society*, (1):108–119, 2002.

[80] J. T. Richardson. The use of latin-square designs in educational and psychological research. *Educational Research Review*, pages 84–97, 2018.

[81] M. K. S. Santos. Virtual windshields: merging reality and digital content to improve the driving experience. 2013.

[82] H. Sawano and M. Okada. Real-time video processing by a car-mounted camera and its application to car navigation systems by an augmented reality-based display method. *The Journal of the Society for Art and Science*, (2):57–68, 2006.

[83] D. Schmalstieg, A. L. Fuhrmann, and G. Hesina. Bridging multiple user interface dimensions with augmented reality. In *ISAR*, pages 20–29, 2000.

[84] Y.-W. Seo, C. Urmson, and D. Wettergreen. Exploiting publicly available cartographic resources for aerial image analysis. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 109–118. ACM, 2012.

[85] Z. Shan, H. Wu, W. Sun, and B. Zheng. Cobweb: a robust map update system using gps trajectories. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 927–937. ACM, 2015.

[86] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu. Stpp: Spatial-temporal phase profiling-based method for relative rfid tag localization. *IEEE/ACM Transactions on Networking*, 25(1):596–609, 2017.

[87] I. Skog and P. Handel. In-car positioning and navigation technologies—a survey. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):4–21, 2009.

[88] SKYHOOK. Skyhook, Aug. 2017.

[89] R. Song, W. Lu, W. Sun, Y. Huang, and C. Chen. Quick map matching using multi-core cpus. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 605–608. ACM, 2012.

[90] A. Steiner and A. Leonhardt. Map-generation algorithm using low-frequency vehicle position data. In *Transportation Research Board 90th Annual Meeting*, number 11-0486, 2011.

[91] Sygic. Sygic official website, Jan. 2018.

[92] H. Tae-Hyun, J. In-Hak, and C. Seong-Ik. Detection of traffics for vision-based car navigation system. In *Springer Advances in Image and Video Technology*, pages 682–691. 2006.

[93] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman. Lane marking aided vehicle localization. In *Proceedings of IEEE ITSC*, pages 1509–1515, 2013.

[94] TESLA. Tesla official websit, Jan. 2018.

[95] R. Toledo-Moreo, D. Bétaille, and F. Peyret. Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps. *IEEE Transactions on Intelligent Transportation Systems*, (1):100–112, 2010.

[96] G. User. Source code of walkway discovery from large scale crowdsensing, Feb. 2018.

[97] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang. Ccs-ta: quality-guaranteed online task allocation in compressive crowdsensing. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 683–694. ACM, 2015.

[98] 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*, pages 27–40. ACM, 2013.

[99] WAYRAY. Wayray website, Jan. 2018.

[100] C. Wenk, R. Salas, and D. Pfoser. Addressing the need for map-matching speed: Localizing globalb curve-matching algorithms. In *null*, pages 379–388. IEEE, 2006.

[101] C. E. White, D. Bernstein, and A. L. Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation research part c: emerging technologies*, 8(1):91–108, 2000.

[102] Wikipedia. Crowdsensing, 2018.

[103] E. Wilhelm, Y. Zhou, N. Zhang, J. Kee, G. Loh, and N. Tippenhauer. Sensg: Large-scale deployment of wearable sensors for trip and transport mode logging. In *Transportation Research Board 95th Annual Meeting*, number 16-4353, 2016.

[104] Wireless@SG. Wireless@sg, Aug. 2017.

[105] S. Worrall and E. Nebot. Automated process for generating digitised maps through gps data compression. In *Australasian Conference on Robotics and Automation*, volume 6. Brisbane: ACRA, 2007.

[106] B. Worton. A review of models of home range for animal movement. *Ecological modelling*, 38(3-4):277–298, 1987.

[107] H. Wu, C. Tu, W. Sun, B. Zheng, H. Su, and W. Wang. Glue: a parameter-tuning-free map updating system. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 683–692. ACM, 2015.

[108] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier. Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint. In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 55–62. IEEE, 2015.

[109] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi. Emc 3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint. *IEEE Transactions on Mobile Computing*, 14(7):1355–1368, 2014.

[110] H. Xiong, D. Zhang, L. Wang, J. P. Gibson, and J. Zhu. Eemc: Enabling energy-efficient mobile crowdsensing with anonymous participants. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):39, 2015.

[111] L. Xu, X. Hao, N. D. Lane, X. Liu, and T. Moscibroda. More with less: Lowering user burden in mobile crowdsourcing through compressive sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 659–670. ACM, 2015.

[112] Y. Yamaguchi, T. Nakagawa, K. Akaho, M. Honda, H. Kato, and S. Nishida. Ar-navi: An in-vehicle navigation system using video-based augmented reality technology. In *Springer Symposium on Human Interface and the Management of Information*, pages 1139–1147, 2007.

[113] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 173–184. ACM, 2012.

[114] K.-M. Yang and S.-B. Cho. A non-gps low-power context-aware system using modular bayesian networks. *Proceedings of the MOBILITY*, 2014.

[115] H. Yin and O. Wolfson. A weight-based map matching method in moving objects databases. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 437–438. IEEE, 2004.

[116] C.-W. You, N. D. Lane, F. Chen, R. Wang, Z. Chen, T. J. Bao, M. Montes-de Oca, Y. Cheng, M. Lin, L. Torresani, et al. Carsafe app: Alerting drowsy and distracted

drivers using dual cameras on smartphones. In *Proceedings of ACM MobiSys*, pages 13–26, 2013.

[117] Y. Yuan, D. Zhang, F. Miao, J. A. Stankovic, T. He, G. Pappas, and S. Lin. Dynamic integration of heterogeneous transportation modes under disruptive events. In *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, pages 65–76. IEEE Press, 2018.

[118] D. Zhang, J. Huang, Y. Li, F. Zhang, C. Xu, and T. He. Exploring human mobility with multi-source data at extremely large metropolitan scales. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 201–212. ACM, 2014.

[119] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, and T. He. coride: carpool service with a win-win fare model for large-scale taxicab networks. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 9. ACM, 2013.

[120] D. Zhang, H. Xiong, L. Wang, and G. Chen. Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 703–714. ACM, 2014.

[121] P. Zhou, Y. Zheng, and M. Li. How long to wait?: predicting bus arrival time with mobile phone based participatory sensing. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 379–392. ACM, 2012.

[122] M. V. Zoej and M. Mokhtarzade. Road detection from high resolution satellite images using artificial neural networks. In *Proceedings of ISPRS Congress, Istanbul, Turkey*, volume 35, page 104, 2004.