13 Mapping Parking Spaces Using Crowd-Sourced Trajectories

SUBHRASANKHA DEY, SALIL GOEL, MARTIN TOMKO, AND STEPHAN WINTER

Abstract

Mapping urban parking spaces helps drivers to reduce their search and cruising for parking, thus reducing traffic, reducing emissions, and reducing total travel times. Mapped urban parking spaces can also be monitored for real-time occupancy information. But while many cities in Asia, Africa, and Latin America are experiencing a strong increase of private car use on the roads, they typically lack such reliable information regarding on-street parking spaces. Hence, in this chapter we explore globally applicable mapping methods for on-street parking locations, as a first step towards smart parking (for an alternative approach see Chapter 11).

Keywords

Parking space, parking lot, mapping, trajectory

13.1 Introduction

The necessity to cruise for parking in urban centers leads to extended trip times and causes extra congestion up to 30 percent of total traffic flows (Shoup, 2017; Shoaeb et al., 2016; Hansen, 2018; Bischoff and Nagel, 2017; Chai et al., 2019; Brooke et al., 2018; Bischoff et al., 2019). Cruising for parking is caused by the scarcity of public parking capacity in the urban centers (Shoup, 2017), and further increases due to a lack of information about parking spaces in common navigation systems (Benenson et al., 2008; Bischoff and Nagel, 2017). The first step towards providing this information is mapping the public parking spaces in cities. But map information regarding parking spaces is often missing or incomplete for a whole range of reasons, such as lack of funding or commitment of mapping authorities to capture parking spaces, or lack of common agreement of what constitutes a parking space (see Chapter 8.1 for the variety of their nature: marked or unmarked, on-street or off-street, dedicated or grabbed, legal or illegal). One way of approaching the challenge of globally mapping urban parking spaces is therefore crowd-sourcing (Coric and Gruteser, 2013; Bock et al., 2019; Di Martino et al., 2019).

https://doi.org/10.34727/2021/isbn.978-3-85448-045-7_13

Recently, crowd-sourced trajectory data are becoming popular in intelligent transportation systems (ITS) and science due to vast range of applications in the transport domain. This kind of data is attractive particularly in countries with lacking public infrastructure, such as India. Hence, in this chapter we will discuss whether crowd-sourced trajectory data can be used to extract parking information. We will provide a brief overview about existing research in mapping of parking spaces. Then we will discuss about crowd-sourcing based trajectory data, and the applicability of trajectory data in the context of parking information extraction. We will then provide two case studies to demonstrate the novelty of mapping parking spaces using only crowd-sourced trajectory data. We will conclude by discussing the future potential of existing methodologies in the context of parking information extraction from crowd-source trajectory data.

13.2 Literature Review

Authoritative mapping of on-street parking spots, e.g., by city councils[1] or by mapping technology firms such as Google Maps, are lengthy and costly processes (Coric and Gruteser, 2013), and are typically limited to marked parking spaces, which is only a subset of all parking opportunities in a city. Parking information can be captured using dedicated infrastructures. As an example, the SFpark project[2] where sensors were placed under the pavement beneath the marked parking spaces. Another example is the PARKNET project (Mathur et al., 2010), which captured information on marked parking spaces using ultrasonic sensors and GNSS (Global Navigation Satellite Systems) units on floating vehicles (Mathur et al., 2010).

However there are alternative methods for mapping parking spaces focusing on a crowd-sourcing based globally applicable solutions. These methods utilize moving vehicles equipped with proximity sensors (e.g., ultrasonic sensors, or electromagnetic sensors) to detect parked vehicles. The vehicles are also equipped with units to record the vehicle's coordinates and time-stamps. This spatio-temporal data can be used to identify whether the vehicle was stationary or moving at a time. When the vehicle is stationary, the recorded location contains the information of a parking space. Crowd-sourcing removes the costs of dedicated infrastructure for parking related data collection, and enables to more comprehensively track where people actually park in the city. One such crowd-sourcing approach (Coric and Gruteser, 2013) uses collected data for identification of legal and illegal on-street parking spaces. Rinne et al. (2014) provided a detail discussion on the pros and cons of crowd-sourcing based parking information collection and concluded as a promising approach to help users when dedicated infrastructure based sensors are unavailable. Many smart parking apps

[1] `https://data.melbourne.vic.gov.au/` – City of Melbourne
[2] `https://bit.ly/399SLA6` – San Francisco Municipal Transportation Agency, May 2018

use crowd-sourced GNSS trajectory data, and motivate smartphone users to voluntarily share parking related information from smartphone users (Kopeckỳ and Domingue, 2012). Farkas and Lendák (2015) investigated the effect of crowd-sourcing activities for urban parking scenario and presented as a case study using a multi-agent simulation. The goal of the simulation was to investigate the role of parking occupancy information on assisting drivers. Simulation results reveals that 30 % participation in crowd-sourcing leads to 14 % shorter cruising time.

Coric and Gruteser (2013) utilized crowd-sourcing to identify illegal parking spaces in the on-street parking maps without the assumptions of existing parking map database. Recently, vehicles are many times equipped with parking sensors in order to assist drivers during the end of cruising. Coric and Gruteser (2013) have utilized such vehicles in a roaming condition equipped with parking sensors to detect a parked vehicle along-with the roaming vehicle's locations with time-stamps. The recorded locations and time-stamps of the roaming vehicle thus becomes a GNSS trajectory data. Legality of the parking space is evaluated using a centralized server that needs several sensor measurements from the same location. In order to achieve their goal, Coric and Gruteser (2013) used crowd-sourcing to collect a large dataset of roaming vehicle's trajectory. Such crowd-sourced GNSS trajectory data have been also used earlier by data mining researchers to infer road maps (Biagioni and Eriksson, 2012; Liu et al., 2012), demonstrating that this data source is accurate enough for high-detail urban mapping (Haklay, 2010) and again, to track or predict the occupancy of parking spaces (Zheng et al., 2015). Thus we can be certain about the importance of crowd-sourced GNSS trajectory data, or simply trajectory data. In the upcoming sections, we will discuss about some of the unexplored concepts in order to map parking spaces only from trajectory data. The organization of the chapter is as follows: We will discuss certain characteristics of a trajectory dataset in Section 13.3. In Section 13.4, we will discuss different methods to utilize a trajectory dataset for extracting parking related information. In those methods, we will discuss the independence of dedicated infrastructure. We will then present a case study with a real world trajectory dataset collected using crowd-sourcing in Section 13.5.

13.3 Crowd-sourced Trajectory Datasets

A GNSS trajectory data set contains records of the discrete positions of the mobile sensing device over time. The sensing device – for example a smartphone – is typically carried by a person or a vehicle. Hence, the typical structure of a trajectory is:

$$\begin{aligned}&\lambda_1, \phi_1, t_1\\&\lambda_2, \phi_2, t_2\\&\dots\\&\lambda_n, \phi_n, t_n\end{aligned}$$

with positions recorded in geographic longitude λ_i and latitude ϕ_i, and time t_i recorded in the local time zone (converted from GPS time). This trajectory starts at t_1, when the device is switched on, and stops at t_n, when the device is switched off. An optional parameter in a trajectory data set is the *trip ID*, separating the trajectories of the same sensing device over time into trips. The device separates trips by the turning on or turning off of the sensing. Hence, a trajectory can record a whole *trip* as defined in Chapter 3, subsuming all mobility between two longer stationary activities (Das and Winter, 2016), or parts of a trip, or more than a trip, all labeled by one *trip ID*. For example, a trajectory can be recorded by tuples where j represents the trip ID and k the time instance:

$$\{(\lambda_k, \phi_k, t_k)_j\}.$$

In addition, the trajectory of the mobile sensing device can be shared, and thus, trajectory datasets from multiple devices can be crowd-sourced through platforms. In this case, a second additional identifier i is introduced, characterizing the sensing device that had collected a specific trajectory:

$$\{(\lambda_{j.k}, \phi_{j.k}, t_{j.k})_i\}.$$

Trajectory data can be recorded traveling with any switched-on tracking application, such as a smartphone navigation app, a dedicated vehicle navigation service, or a vehicle's black box (Zheng et al., 2008).

13.3.1 Multi-modality of a Trip

A crowd-sourced trajectory dataset consists of multiple trips. Trip data collected by tracking devices on board vehicles is vehicle-only by nature, i.e., uni-modal. Both the first record of a uni-modal trip (approximating the origin of a trip) and the last record of a the same trip (approximating the destination of a trip) contain valuable information of parking spaces. For example, the presence at the same location between the last record of one trajectory and the first record of the next trajectory implies the possibility of the vehicle having been in a parking location in the time interval between these records. A single trip of a trajectory dataset can also be multi-modal depending upon the user's mobility activities while collecting the data (e.g., walking to the parked car, driving and parking, and walking to the destination). The recording of multi-modal trips is done by a person-bound device (such as a smartphone running a navigation app). It can capture the person's movements while being on board of the vehicle, but also their walking, their travel on other modes, and even their stationary activity locations. Parking locations can be found from the changes of modes in such multi-modal trajectory from drive to walk or from walk to drive. Since data collection depends on the travelers' switching on their navigation (or tracking) service, parts of trips or even full trips may not be covered. Also, because travelers tend to switch off when the service is no longer needed – either because they approach their destination, or because they enter environments they are familiar with – ends of trips may not be fully covered. Many trips seem to stop mid-trip because the smartphone (or the navigation service on the smartphone) has been switched off. Still, there is a correlation between trips and parking spaces that we will explore in collected trajectory data.

13.3.2 Labeled with Transport Mode

Trip data may be labeled with the mode of transport. For a labeled car-only trip data, identification of parking spaces is straight-forward: Start points and end points of trips can be identified by the temporal gaps between trips, and these two points of a trip indicate a parking space within the accuracy of GNSS. On the other-hand, trip data recorded by a person bound device can be labeled with car and walk as mode of transport. Since these trips are in principle multi-modal, one can only assume that where a person enters a car (switching from walking to driving) or gets out of a car (switching from driving to walking) – the *change points* (Dabiri et al., 2019) – the car is in a parking position.

13.3.3 Unlabeled with Transport Mode

Only a few published trajectory datasets are labeled, and this motivates researchers to investigate the utilities of unlabeled trajectory data (Zheng et al., 2008; Cottrill et al., 2013). However, identification of change points from an unlabeled multi-modal trajectory data is not straight forward. Multi-modal trajectory data collected in the field is unlabeled and requires a travel mode detection first. Travel mode detection techniques require classification algorithms that are trained with feature values (e.g., velocity, acceleration, and change of direction) that are either extracted from the trips of a trajectory dataset, or potentially sourced from further sensor readings such as an accelerometer, a compass, or an inertial measurement unit (Jahangiri and Rakha, 2015; Etemad et al., 2018; Dabiri et al., 2019; Zheng et al., 2008). The travel mode detection algorithm thus identify the change points after estimating travel modes in unlabeled trips. Travel mode detection is done after:

1. extracting salient feature values (e.g., velocity, acceleration, and change of

direction) from a trajectory data, and

2. training a classification algorithm using the extracted feature values (Etemad et al., 2018; Dabiri et al., 2019)).

If parking related transportation modes (i.e., driving a car, walking), and thus, change points, be detected from unlabeled trajectory data reliably, these identified change points can be also used to map parking spaces.

Let us assume there are a total of I trips in a crowd-sourced trajectory dataset, $I \geq 1$. A trip can contain a number J of data points, $J \geq 2$. Let the trip i have J^i number of data points, then the j^{th} data point of trip i contains the tuple $< x^i_j, y^i_j, t^i_j, tr^i_j, m^i_j >$, where x^i_j is (usually) the longitude, y^i_j is (usually) the latitude, t^i_j is the timestamp of the location recording, tr^i_j is the trip id (i in this case), and m_j is the mode of transport used when arriving at j^{th} data point, $1 \leq j \leq J^i$. Hence, if a mode change happens at the j^{th} data point, then $m^i_j \neq m^i_{j+1}$, and j becomes a *change point*. A trip i may contain multiple change points, collected in C^i, the set of all change points of trip i. C^i contains the tuples longitude and latitude of change points. Thus, $C = C^1 \cup \ldots \cup C^i \cup \ldots C^I$ for all change points in I.

13.3.4 Salient Features of a Trip

In this section we will discuss about the salient features of a trajectory dataset. Features are extracted to build a training dataset from a trajectory dataset with a number I of trips in order to classify unlabeled data points. According to the previous research, three such salient features are velocity (v_j), acceleration (a_j), and change of direction (dr_j). At each data point, these derived features are defined as follows for trip i:

$$v^i_j = \frac{\sqrt{(x^i_j - x^i_{j-1})^2 + (y^i_j - y^i_{j-1})^2}}{t^i_j - t^i_{j-1}} \tag{13.1}$$

$$a^i_j = \frac{v^i_j - v^i_{j-1}}{t^i_j - t^i_{j-1}} \tag{13.2}$$

$$dr^i_j = tan^{-1}\frac{y^i_j - y^i_{j-1}}{x^i_j - x^i_{j-1}} - tan^{-1}\frac{y^i_{j+1} - y^i_j}{x^i_{j+1} - x^i_j} \tag{13.3}$$

where $2 \leq j \leq J^i$ for v^i_j, $2 \leq j \leq J^i$ for dr^i_j, and $3 \leq j \leq J^i$ for a^i_j.

13.4 Mapping of Parking Spaces Using Multi-modal Trips

We have already discussed how to extract parking spaces from a car-only trip data. In this section, we will focus on multi-modal trips labeled with transport modes (either recorded manually, or predicted using mode detection techniques). Theoretically, for multi-modal trips, we can divide the set of change points (C) into two categories: the walk-to-car change points at the start of a car-part of the trip, and car-to-walk change points as the end of a car-part of the trip. A multi-modal trip can contain a single or multiple change points.

13.4.1 Single Change Point Trips

The trips with single change point (SCP) have only one sub-trip with travel mode labeled with *car*, and only one sub-trip with travel mode labeled with *walk*. SCP trips can have either a walk-to-car (C_S) or a car-to-walk (C_E) change points where $\{C_E \cup C_S\} \subset \mathbf{C}$.

Sub-trips and change points of two SCP trips i and k ($k \neq i$) are illustrated in Figure 13.1. A parking space is a region with a finite area. The area of a parking space can vary depending on its category from a few square meters (street side marked parking space) to a few hundred square meters (parking lot). There are multiple incoming and outgoing trips from a parking space. Hence, if a region with finite area contains multiple change points from different SCP trips, we can say that a valid parking space has been identified. Thus, a valid parking space with finite area $\mathbf{P}$ contains at least one C_E^i and one C_S^k:

$$\{C_E^i \cup C_S^k\} \subset \mathbf{P}. \tag{13.4}$$

Figure 13.1: Sub-trips and change points of two SCP trips i and k.

13.4.2 Multiple Change Point Trips

Change points are separate different modes of travel in a multi-modal trip. Multiple change points can be found in a multi-modal trip if there are multiple sub-trips of different travel modes. These trips are defined as multiple change point trips (or MCP trips). Let C_{WC}^l and C_{CW}^l be change points for car-to-walk and walk-to-car respectively in an MCP trip l. Clearly, $C_{WC}^l \subset \mathbf{P}$ and $C_{CW}^l \subset \mathbf{P}$ if the person walking returns to the same vehicle, and thus, the vehicle was surely parked all the time at the location characterized by the two observations C_{WC}^l and C_{CW}^l. Sub-trips and change points of such a trip l are shown in Figure 13.2. In the absence of measurement errors:

$$C_{CW}^l = C_{WC}^l. \tag{13.5}$$

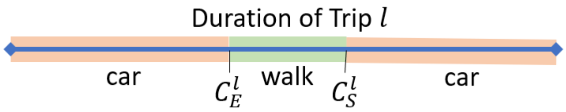

Figure 13.2: Sub-trips and change points of a multiple change point trip l.

13.4.3 Effect of Measurement Error

A trajectory records the continuous movement of a mobile object in discrete time (Ranacher et al., 2016b). Hence there is a possible source of error in the collected data caused by noise or systematic effects of the positioning observations, such as GNSS (Ranacher et al., 2016a). Due to this error, two recorded point locations of a same geographic point location are not guaranteed to have equal coordinates. This measurement error is often modeled as the Gaussian noise in the measurement $N(0, \Sigma)$, where Σ is the variance-covariance matrix of the measurement error:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}.$$

Let us understand this with an example, and let us assume $\sigma_x = \sigma_y = \sigma$ for simplicity. Let O_1 and O_2 be two positioning observations from the same location, recorded at different times and/or using different devices. As the measurement error is normally distributed, in the absence of systematic errors the true location of the device(s) can be expected to be with a likelihood of 99.7 % within a circle of a radius of 3 σ. Thus, Figure 13.3 illustrates possible situations for two recorded locations O_1 and O_2 that have actually the same true location:

- In the worst case, $\|O_1 - O_2\| = 6\sigma$, with the circles meeting in one point.
- In the average case, $\|O_1 - O_2\| < 6\sigma$, indicating some overlap of the circles less than the area of the either circle.
- In the best case (but with a likelihood of 0), $O_1 = O_2$, indicating coinciding O_1 and O_2 as in Equation 13.5.

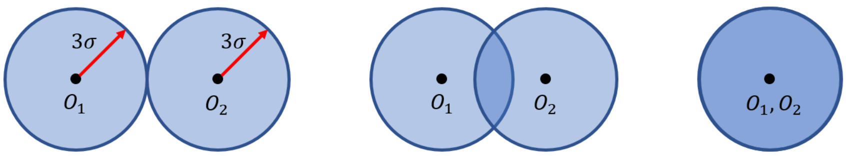

Figure 13.3: Three possible case scenarios for two recorded points with same true latitude and longitude.

13.5 Case Study with GeoLife Data

We have prepared a case study with the labeled GNSS trajectories collected in the GeoLife project (Zheng et al., 2011). This dataset has 75 trips with 130,973 data points that are labeled with both parking related transportation modes: *walk* and *car*. We have used Google Maps and two APIs from the Google cloud platform[3]: Places API, and Street View Static API for the case study. These APIs are used to extract the locations of valid parking point locations P with additional information (e.g., the type of parking space). The value of the measurement error's standard deviation σ is not provided in the GeoLife dataset, hence, let us assume this σ as 10 meters (Merry and Bettinger, 2019; Ranacher et al., 2016a). In this section, we will discuss about different techniques to identify parking spaces and the type of the parking space (e.g., off-street or on-street) by querying an existing map-database (e.g., Google Maps).

13.5.1 Single Change Point Trips

A valid parking location is observed by above methods by a point P, and is stored in a map database by a point location. Practically however, a parking space is an area **P** bigger than a point, and **P** can contain any number of different observations P, or map references P. For each parking point location in the Google Maps database, we have estimated a parking space **P** by constructing the smallest convex hull that contains P in a map. Nodes of this smallest convex hull are typically on the nearest road segment. We will investigate whether change points of SCP trips (C_E^i and C_S^k) are likely to fall inside $\mathbf{P}$ as described in Equation 13.4 such that:

$$\begin{aligned} \mathbf{P} \cap \mathbf{C}(C_E^i, 3\sigma) \neq \varnothing \\ \mathbf{P} \cap \mathbf{C}(C_S^k, 3\sigma) \neq \varnothing \end{aligned} \tag{13.6}$$

where $\mathbf{C}(C, 3\sigma)$ represents a circle with center C and radius 3σ.

Figure 13.4 is representing one valid parking space extracted from Google Maps as an example where C_S and C_E are observations (with their uncertainty areas) of multiple SCP trips, satisfying Equation 13.6.

13.5.2 Multiple Change Point Trips

In the absence of measurement errors, C_{CW}^l and C_{WC}^l are the same geographic locations in Equation 13.5. Let σ^2 be the measurement error variance for trip l. Hence, σ is the radius of circles that indicate the possible region of the true location, centered at C_{CW}^l and C_{WC}^l. Thus two recorded points with same true

[3]https://developers.google.com/maps/documentation/api-picker

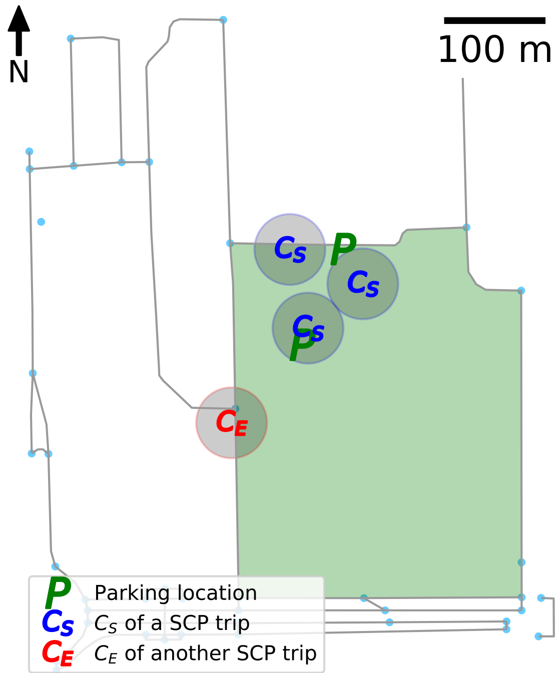

Figure 13.4: Mapped parking spaces using SCP trips of GeoLife trajectory data.

location can be found in the overlapping region of the two circles. In the worst case scenario (Figure 13.3), the recorded points are away from each other by a distance of 6 times of σ with $\approx$ 98 % probability. Thus we get the constraints for validating the same parking spaces recorded twice:

$$C^l_{WC} - C^l_{CW} \leq 6\sigma. \tag{13.7}$$

Equation 13.5 satisfies if Equation 13.7 holds true for the trip l. Thus we can conclude that a parking space $\mathbf{P}$ is found using change points C^l_{CW} and C^l_{WC} of trip l.

A map is presented in Figure 13.5 to show the true locations of parking spaces as extracted from Google Maps[4] as well as the locations of change points of MCP trips (C_{CW} and C_{WC}) extracted from the GeoLife dataset. The radius of the blue circle is chosen to be smaller to illustrate that these two change points are coinciding in the map, indicating that the vehicle was stationary at that point. Thus it should be a parking space further supported by Figure 13.5 where change points are found inside a valid parking space P. Thus change points of crowd-sourced MCP trips can be used to identify unmapped parking spaces.

[4] `https://developers.google.com/maps/documentation/api-picker`

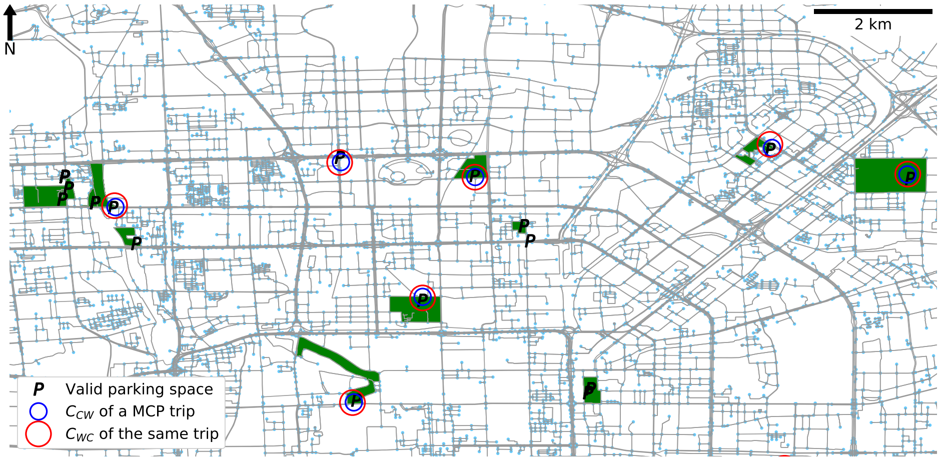

Figure 13.5: Mapped parking spaces using MCP trips of GeoLife trajectory data.

13.6 Reflection on Indian Traffic

Many cities in India have incomplete information on parking spaces due to lack of dedicated parking infrastructure. Hence, the applicability of crowd-sourced trajectory data for such cities in India can be beneficial for users and less costly than authoritative infrastructure. For illustration purposes, the model presented above has been applied to volunteered trajectory data collected from the social activities in the city of Kolkata, the former capital of India (Figure 13.6). The green P are the parking locations collected again from some trajectories in the Google Maps database. Change points are often overlapping with the mapped parking spaces indicating the validity of the model. This pattern can be seen in many social activity places, e.g., the Avani Riverside Mall (a shopping mall), Park Street (a tourist place), and Howrah station (one of the largest railway stations in east India). Many of these parking spaces are not mapped even in the Google Maps database, concluding the future potential of the model.

However, availability of crowd-sourced trajectory data is subject to question in the Indian context. Willingness of the crowd towards participation in crowd-sourcing to produce maps, as well as then to utilize parking information also requires further investigation. Effectiveness of the method also depends on other aspects, e.g., the enforcement regimes on traffic regulations. For example, if sharing of trajectory data means that also illegal parking is captured, and that parking fines for this illegal parking may be levied, people may abstain from sharing their own trajectory information. However, the potential of crowd-sourcing remains unquestionable irrespective of the crowd's willingness.

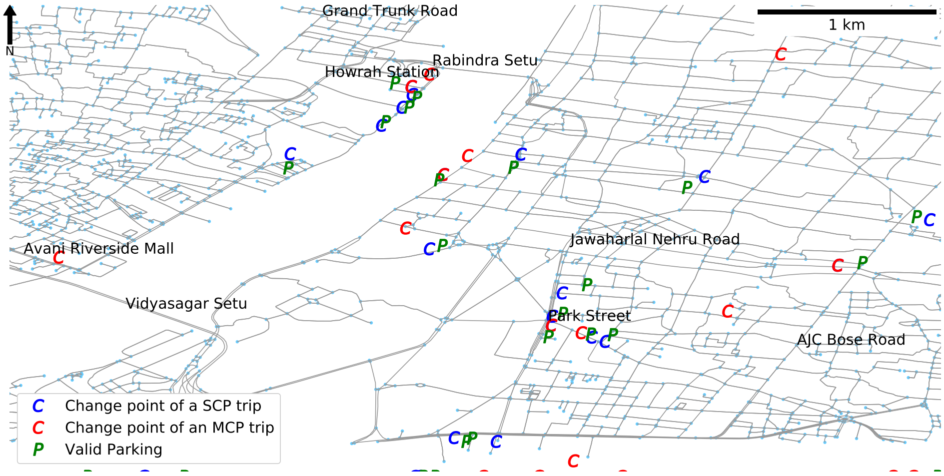

Figure 13.6: Reflection of the model in Kolkata, India.

13.7 Conclusions

In this chapter, we have discussed about different methodologies for parking space identification from crowd-sourced trajectory data. We have shown that these trajectory data are capable of mapping parking spaces and addressed the underlying challenges while doing so. Trajectory data those are labeled with transport mode can be used to identify *change points*. We have discussed about how these change points in a trip can be further used for mapping parking spaces. These change points can be used in a map database to extract details about parking spaces.

In future work, it will be required to predict the category of parking without using a map database. Future work should investigate the robustness of existing methodologies in the context of mapping parking spaces from crowd-sourced data. In future, there is a scope of important research on calculating the parking time of a car inside a parking space without interfering with the privacy of a user.

Bibliography

Benenson, I., Martens, K., and Birfir, S. (2008). PARKAGENT: An agent-based model of parking in the city. *Computers, Environment and Urban Systems*, 32(6):431–439.

Biagioni, J. and Eriksson, J. (2012). Inferring road maps from gps traces: Survey and comparative evaluation. In *91st Annual Transportation Research Board*.

Bischoff, J., Maciejewski, M., Schlenther, T., and Nagel, K. (2019). Autonomous vehicles and their impact on parking search. *IEEE Intelligent Transportation Systems Magazine*, 11(4):19–27.

Bischoff, J. and Nagel, K. (2017). Integrating explicit parking search into a transport simulation. *Procedia Computer Science*, 109:881–886.

Bock, F., Di Martino, S., and Sester, M. (2019). What is the impact of on-street parking information for drivers? In *International Symposium on Web and Wireless Geographical Information Systems*, pages 75–84. Springer.

Brooke, S., Ison, S., and Quddus, M. (2018). Analysing parking search ('cruising') time using generalised multilevel structural equation modelling. *Journal of Transport Economics and Policy (JTEP)*, 52(3):202–220.

Chai, H., Ma, R., and Zhang, H. M. (2019). Search for parking: a dynamic parking and route guidance system for efficient parking and traffic management. *Journal of Intelligent Transportation Systems*, 23(6):541–556.

Coric, V. and Gruteser, M. (2013). Crowdsensing maps of on-street parking spaces. In *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 115–122. IEEE.

Cottrill, C., Pereira, F., Zhao, F., Dias, I., Lim, H., Ben-Akiva, M., and Zegras, P. C. (2013). Future mobility survey. *Transportation Research Record*, 2354:59–67.

Dabiri, S., Lu, C.-T., Heaslip, K., and Reddy, C. K. (2019). Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data. *IEEE Transactions on Knowledge and Data Engineering*, 32(5):1010–1023.

Das, R. D. and Winter, S. (2016). A context-sensitive conceptual framework for activity modelling. *Journal of Spatial Information Science*, 2016(12):45–85.

Di Martino, S., Vitale, V. N., and Bock, F. (2019). Comparing different on-street parking information for parking guidance and information systems. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1093–1098. IEEE.

Etemad, M., Júnior, A. S., and Matwin, S. (2018). Predicting transportation modes of gps trajectories using feature engineering and noise removal. In *Canadian Conference on Artificial Intelligence*, pages 259–264. Springer.

Farkas, K. and Lendák, I. (2015). Simulation environment for investigating crowdsensing based urban parking. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 320–327. IEEE.

Haklay, M. (2010). How good is volunteered geographic information? a comparative study of openstreetmap and ordnance survey datasets. *Environment and Planning B*, 37(4):682–703.

Hansen, I. (2018). Determination and evaluation of traffic congestion costs. *European Journal of Transport and Infrastructure Research*, 1(1):61–72.

Jahangiri, A. and Rakha, H. A. (2015). Applying machine learning techniques to transportation mode recognition using mobile phone sensor data. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2406–2417.

Kopeckỳ, J. and Domingue, J. (2012). ParkJam: Crowdsourcing parking availability information with linked data. In *Extended Semantic Web Conference*, pages 381–386. Springer.

Liu, X., Biagioni, J., Eriksson, J., Wang, Y., Forman, G., and Zhu, Y. (2012). Mining large-scale, sparse gps traces for map inference: comparison of approaches. In *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 669–677.

Mathur, S., Jin, T., Kasturirangan, N., Chandrasekaran, J., Xue, W., Gruteser, M., and Trappe, W. (2010). Parknet: drive-by sensing of road-side parking statistics. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pages 123–136.

Merry, K. and Bettinger, P. (2019). Smartphone gps accuracy study in an urban environment. *PloS one*, 14(7):e0219890.

Ranacher, P., Brunauer, R., Trutschnig, W., Van der Spek, S., and Reich, S. (2016a). Why GPS makes distances bigger than they are. *International Journal of Geographical Information Science*, 30(2):316–333.

Ranacher, P., Brunauer, R., Van der Spek, S., and Reich, S. (2016b). What is an appropriate temporal sampling rate to record floating car data with a GPS? *ISPRS International Journal of Geo-Information*, 5(1):1.

Rinne, M., Törmä, S., and Kratinov, D. (2014). Mobile crowdsensing of parking space using geofencing and activity recognition. In *10th ITS European Congress, Helsinki, Finland*, pages 16–19.

Shoaeb, A., El-Badawy, S., and Gabr, A. (2016). Direct costs of urban traffic congestion case study: Main corridors in Mansoura City. *International Journal of Computing, Communication and Instrumentation Engineering*, 3(2):1–8.

Shoup, D. (2017). *The High Cost of Free Parking*. Routledge.

Zheng, Y., Fu, H., Xie, X., Ma, W.-Y., and Li, Q. (2011). *Geolife GPS trajectory dataset - User Guide*, Geolife GPS trajectories 1.1 edition. Geolife GPS trajectories 1.1.

Zheng, Y., Liu, L., Wang, L., and Xie, X. (2008). Learning transportation mode from raw gps data for geographic applications on the web. In *Proceedings of the 17th International Conference on World Wide Web*, pages 247–256.

Zheng, Y., Rajasegarar, S., and Leckie, C. (2015). Parking availability prediction for sensor-enabled car parks in smart cities. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6. IEEE.