

Stephan Winter / Salil Goel (Eds.)

# SMART PARKING IN FAST-GROWING CITIES

Challenges and Solutions



Academic Press

Stephan Winter / Salil Goel (Eds.)  
SMART PARKING IN FAST-GROWING CITIES



Stephan Winter / Salil Goel (Eds.)

# **SMART PARKING IN FAST-GROWING CITIES**

Challenges and Solutions



**Academic Press**

Cite as:

Winter, S., & Goel, S. (Eds.). (2021). *Smart Parking in Fast-Growing Cities: Challenges and Solutions*. TU Wien Academic Press. <https://doi.org/10.34727/2021/isbn.978-3-85448-045-7>

**TU Wien Academic Press, 2021**

c/o TU Wien Bibliothek  
TU Wien  
Resselgasse 4, 1040 Wien  
[academicpress@tuwien.ac.at](mailto:academicpress@tuwien.ac.at)  
[www.tuwien.at/academicpress](http://www.tuwien.at/academicpress)



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). <https://creativecommons.org/licenses/by-sa/4.0/>

ISBN (Online): 978-3-85448-045-7

Available online: <https://doi.org/10.34727/2021/isbn.978-3-85448-045-7>

Media proprietor: TU Wien, Karlsplatz 13, 1040 Wien

Publisher: TU Wien Academic Press

Editors (responsible for the content): Stephan Winter and Salil Goel

## **Dedication**

This book – like any other book – has taken a toll on the time the authors could have spent with their nearest. And yet it was these people who encouraged us and supported the entire writing process. This book is dedicated to their generosity and love.

## Preface

Parking in cities is a challenge in every country, and the focus of a wide range of research and development fields, from sensing and communication technology, to artificial intelligence and operations research. Parking cannot be looked at in isolation, but rather as a critical element in the complex fabric of urban mobility. It is bound by the dynamics between parking demand, parking supply, and human behavior. Demand can be influenced by urban planning, transport planning (especially the provision of competitive alternative traveling modes), and market mechanisms such as pricing or gamifying. In many ways this book stops short at this point, because it only looks at supporting the more efficient use of existing supply. This book focuses on data-driven means of a smart city, deployed for a more efficient use of existing parking infrastructure.

Despite being a challenge felt in cities across the globe, parking pressure is experienced and addressed differently depending on the wealth of the country concerned. This book focuses on cities in low- and middle-income countries: cities that are experiencing rapid urbanization and motorization, where free and non-managed on-street parking meets haphazard parking behavior, and where the capacity to invest in parking infrastructure is limited. These factors are in stark contrast to those in cities located in high-income countries where not only is the technical infrastructure advanced, but the use of private vehicles has also been saturated for a while, and for these, the first indicators of peak car use are discussed.

Institutions like the World Bank are phasing out the still prevalent, but ill-defined terms of *developed* and *developing countries*. We call them ill-defined because, in principle, every country is developing further from its current state. The World Bank is replacing these terms by classes of gross national income per capita. We subscribe to this approach since it directly relates to the capacity to invest into infrastructure, including information and communication technology.

This book has been made open-access because of this focus on low- and middle-income countries, going back, at its core, to a project funded by the Indian Government, *Advanced Parking Information and Management for Indian Traffic*. The present work is designed as a resource book. It should immediately help city authorities, engineering firms, and transport engineers world-wide to

---

develop solutions for their specific context. The book collects and reviews published research and technology; it does not contain original research. The rich citations of the scientific literature, however, might help the reader to go deeper into the concerned subject matter. Hence, this book is equally suitable as a reference text in seminars on intelligent transport, especially for IT and engineering students who also want to learn about the foundations of the spatial information used in data-driven approaches.

These intentions are reflected in the structure of the book. The book is divided into two parts. The first part lays out the context of geospatial technologies for urban mobility in smart cities. The second part focuses on parking information and management using these technologies, considered in the context of low and middle income countries.

The authors of this book may be contacted – they would only be too happy to engage.

Stephan Winter and Salil Goel (Eds.)

Melbourne (Australia) and Kanpur (India)

June 29, 2021



## **Authors**

### **Debaditya Acharya**

Department of Manufacturing, Materials and Mechatronics, RMIT University, Melbourne, Victoria 3000, Australia

deb.acharya@rmit.edu.au

### **Ali Aliedani**

Computer Engineering Department, Basrah University, Basrah, Iraq

ali.nabeel@uobasrah.edu.iq

### **Subhrasankha Dey**

Department of Infrastructure Engineering, The University of Melbourne, Parkville, Victoria 3010, Australia

deys@student.unimelb.edu.au

### **Salil Goel**

Department of Civil Engineering, Indian Institute of Technology Kanpur, India

sgoel@iitk.ac.in

### **Kourosh Khoshelham**

Department of Infrastructure Engineering, The University of Melbourne, Parkville, Victoria 3010, Australia

k.khoshelham@unimelb.edu.au

---

**Seng W. Loke**

Department of Computer Science, Deakin University, Burwood, Victoria 3125,  
Australia

seng.loke@deakin.edu.au

**Balasubramanian Nagarajan**

Department of Civil Engineering, Indian Institute of Technology Kanpur, India

nagaraj@iitk.ac.in

**Martin Tomko**

Department of Infrastructure Engineering, The University of Melbourne, Parkville,  
Victoria 3010, Australia

tomkom@unimelb.edu.au

**Yaoli Wang**

Institute of Remote Sensing and Geographic Information Systems, Peking Uni-  
versity, China

wangyaoli@pku.edu.cn

**Stephan Winter**

Department of Infrastructure Engineering, The University of Melbourne, Parkville,  
Victoria 3010, Australia

winter@unimelb.edu.au

## Acknowledgements

This book is the outcome of a collaboration between Indian and Australian researchers, and guest authors who join us. This collaboration was made possible by a grant from SPARC, the Scheme for Promotion of Academic and Research Collaboration of the Government of India that aims at “improving the research ecosystem of India’s higher educational institutions by facilitating academic and research collaborations between Indian institutions and the best institutions in the world from 28 selected nations to jointly solve problems of national and international relevance in the first phase”. The SPARC project *Advanced Parking Information and Management for Indian Traffic* has been a collaborative effort between IIT Kanpur and the University of Melbourne, 2020-21.

Further support has been received by the Australian Government (especially by the grants ARC DP170100109 and ARC DP170100153), by the Melbourne India Partnership Academy, and by the University of Melbourne postgraduate scholarships program.

Finally, our thanks go to TU Wien Academic Press, the open access publisher of the Technical University Vienna, for their generous support. This collaboration has been made possible by the affiliation of one of the editors with this institution.

## Table of Contents

<b>Dedication</b>	<b>v</b>
<b>Preface</b>	<b>vi</b>
<b>Authors</b>	<b>viii</b>
<b>Acknowledgements</b>	<b>x</b>
<b>List of Abbreviations</b>	<b>xv</b>
<b>Geospatial Technologies for Urban Mobility</b>	<b>1</b>
<b>1 Geospatial Technologies for Urban Mobility: Introduction</b> <i>Stephan Winter and Salil Goel</i>	<b>2</b>
<b>2 Reference Systems for Urban Mobility</b> <i>Balasubramanian Nagarajan</i>	<b>7</b>
2.1 Reference Frames and Map Projections . . . . .	7
2.2 Map Projections . . . . .	12
2.3 The Basics of Global Navigation Satellite Systems . . . . .	16
2.4 Conclusions . . . . .	20
<b>3 Tracking Urban Mobility</b> <i>Stephan Winter</i>	<b>22</b>
3.1 Introduction . . . . .	22
3.2 Internet of Things . . . . .	24
3.3 Tracking by Sensors . . . . .	28

---

3.4	Tracking Data Reference Frames . . . . .	30
3.5	Properties of Data . . . . .	31
3.6	Privacy Implications of Transport Data . . . . .	33
3.7	Trips, Segmentation, and Map Matching . . . . .	34
3.8	Conclusion . . . . .	37
<b>4</b>	<b>Navigation in Urban Environments</b>	
	<i>Salil Goel</i>	<b>42</b>
4.1	Introduction . . . . .	42
4.2	Navigation Technologies: An Overview and Comparison . . . . .	45
4.3	Sensor Fusion for Navigation . . . . .	55
4.4	Upcoming Trends in Localization and Navigation . . . . .	57
4.5	Summary and Conclusions . . . . .	57
4.6	Further Reading . . . . .	58
<b>5</b>	<b>Computer Vision Techniques for Urban Mobility</b>	
	<i>Kourosh Khoshelham</i>	<b>61</b>
5.1	Introduction . . . . .	61
5.2	Common Computer Vision Tasks for Mobility Applications . . . . .	62
5.3	Computer Vision with Stationary Sensors . . . . .	64
5.4	Computer Vision with Mobile Sensors . . . . .	67
5.5	Concluding Remarks . . . . .	70
<b>6</b>	<b>Urban Mobility and Parking Demand</b>	
	<i>Stephan Winter</i>	<b>77</b>
6.1	Introduction . . . . .	77
6.2	Active Mobility . . . . .	80
6.3	Motorized Mobility . . . . .	85
6.4	Disrupting Urban Mobility . . . . .	88
6.5	Interacting with Urban Mobility . . . . .	89
6.6	Conclusions . . . . .	91

---

<b>Parking as a Challenge for Urban Mobility</b>	<b>98</b>
<b>7 Parking as a Challenge for Urban Mobility: Introduction</b> <i>Stephan Winter and Salil Goel</i>	<b>99</b>
<b>8 The Nature of Urban Parking</b> <i>Stephan Winter</i>	<b>104</b>
8.1 Demand for Parking . . . . .	104
8.2 Supply for Parking . . . . .	108
8.3 Parking Space in the City . . . . .	110
<b>9 A Review of Smart Car Parking as IoT Systems</b> <i>Ali Aliedani and Seng W. Loke</i>	<b>116</b>
9.1 Introduction . . . . .	116
9.2 Car Parking Overview . . . . .	117
9.3 Cloud and Agent-Based Architectures . . . . .	120
9.4 Discussion and Open Issues on Interoperability . . . . .	124
9.5 Conclusion . . . . .	126
<b>10 Sensors for Parking Occupancy Detection</b> <i>Kourosh Khoshelham</i>	<b>132</b>
10.1 Introduction . . . . .	132
10.2 Smart Parking Concept . . . . .	133
10.3 Parking Occupancy Detection Sensors . . . . .	133
10.4 Comparison, Considerations and Open Challenges . . . . .	141
<b>11 Parking Occupancy Detection and Slot Delineation Using Deep Learning: A Tutorial</b> <i>Debaditya Acharya and Kourosh Khoshelham</i>	<b>143</b>
11.1 Introduction . . . . .	143
11.2 Prerequisites . . . . .	148
11.3 Tutorial 1: Parking Occupancy Detection by Fine-tuning a Pre-trained CNN . . . . .	153
11.4 Tutorial 2: Automatic Parking Slot Delineation Using Deep Learning	160
11.5 Conclusions . . . . .	170

---

<b>12 Reducing Parking Pressure by Sharing Resources</b>	<b>174</b>
<i>Yaoli Wang</i>	
12.1 Current Situation and Issues of Urban Parking . . . . .	174
12.2 Exploration for Parking Solutions . . . . .	175
12.3 Vision of a New Mode: Park-and-Ridesharing . . . . .	177
12.4 Looking Forward to the Future . . . . .	181
<b>13 Mapping Parking Spaces Using Crowd-Sourced Trajectories</b>	<b>184</b>
<i>Subhrasankha Dey, Salil Goel, Martin Tomko, and Stephan Winter</i>	
13.1 Introduction . . . . .	184
13.2 Literature Review . . . . .	185
13.3 Crowd-sourced Trajectory Datasets . . . . .	186
13.4 Mapping of Parking Spaces Using Multi-modal Trips . . . . .	189
13.5 Case Study with GeoLife Data . . . . .	192
13.6 Reflection on Indian Traffic . . . . .	194
13.7 Conclusions . . . . .	195

## List of Abbreviations

**5G** Fifth generation technology standard for broadband cellular networks. 26, 27, 29, 46, 53, 57, 123

**CAM** Cooperative Awareness Message. 123

**CAV** Connected Autonomous Vehicles. 25, 27, 28

**CCTV** Closed Circuit Television. 22, 29, 30, 118, 143, 144, 161

**CNN** Convolutional Neural Network. 62, 145–147, 149–151, 153–157, 159, 160

**CPU** Central Processing Unit. 145–147, 157, 159, 160, 170

**DR** Dead Reckoning. 42, 43, 46, 47, 57

**DSRC** Dedicated Short Range Communication. 26, 27, 57, 123, 125

**GNSS** Global Navigation Satellite Systems. 7, 8, 10, 14, 16–21, 29, 30, 32, 34–36, 42–46, 48–57, 70, 185, 186, 188, 191, 192

**GPS** Global Positioning System. 16, 18–20

**GPU** Graphics Processing Units. 145–147, 154–157, 159, 160, 162, 170

**HD** High-definition (maps). 68–70

**ICT** Information and Communication Technologies. 3

**IMU** Inertial Measurement Unit. 46, 56

**IoT** Internet of Things. 3, 22, 24, 100, 116, 117, 119, 121, 123, 124, 126

**IoU** Intersection over Union (score). 69

**ITS** Intelligent Transportation System. 23, 25, 123, 185

**KF** Kalman Filter. 44, 45, 55, 56



---

**LiDAR** Light Detection and Ranging. 44, 51, 52, 55, 56, 61–65, 67–71

**LORAN** Long Range Navigation. 43, 46, 47, 54

**MCP** Multiple Change Point (trip). 190, 193

**MSL** Mean Sea Level. 9

**OGC** Open Geospatial Consortium. 24

**P&R** Park-and-ride. 176, 177

**P&S** Park-and-ridesharing. 177, 178, 180

**PGI** Parking Guidance and Information (system). 119

**RCNN** Region Based Convolutional Neural Network. 63, 67, 143, 146, 147, 151, 153, 160–163, 166, 169–171

**RFID** Radio-frequency Identification. 23, 28, 30, 118, 121, 132, 137, 138, 141, 142

**RTT** RTT (Return Trip Time. 50, 57

**SAE** Society of Automotive Engineers. 123

**SBAS** Satellite Based Augmentation Systems. 18–20

**SCP** Single Change Point (trip). 190, 192

**SLAM** Simultaneous Localization and Mapping. 52, 56, 57

**SoOP** Signals of Opportunity. 49, 54

**SST** Sea Surface Topography. 9

**SVM** Support Vector Machines. 66, 146, 147

**ToA** Time of Arrival. 49, 50

**UAV** Unmanned Aerial Vehicles. 57

**UTM** Universal Transverse Mercator. 7, 14–16

**UWB** Ultra-Wide Band. 29, 49, 50, 53, 54, 57

**V2I** Vehicle-to-infrastructure (communication). 27, 28

**V2V** vehicle-to-vehicle (communication). 26, 27, 87, 116, 117, 123, 125

---

**V2X** Vehicle-to-anything (communication). 87, 117, 123

**VIO** Visual Inertial Odometry. 52

**Wi-Fi** no further meaning; wireless network protocols based on IEEE 802.11.  
29, 46, 49, 57, 87, 124

# **Geospatial Technologies for Urban Mobility**

# 1 Geospatial Technologies for Urban Mobility: Introduction

STEPHAN WINTER AND SALIL GOEL

## Abstract

This chapter gives a broad introduction to the topics in the first part of this book. This first part considers a range of geospatial technologies for smart cities and urban mobility, and demonstrates their potential to shape the future of urban mobility. In this way, the first part prepares the ground, or the framework, for the second part that focuses on technologies for smart parking and specifically on smart parking challenges in the context of cities where private motorization is not yet saturated. Readers who are already familiar with geospatial technologies and are only interested in the parking challenges, can jump ahead to the second part after this introduction.

## Keywords

Geospatial technologies, positioning, tracking, navigation, mobility

“It is hard to make predictions, especially about the future.” This proverb has already been used in many contexts by poets, humorists, politicians and scientists. It is cited here again because we live in a time where it is particularly hard to predict the future of urban mobility – hard enough so that other publications talk of *disruptions* (Meyer and Shaheen, 2017; Riggs, 2018), and disruptions, or even the risk thereof, mean that any extrapolation from the current state must lead to failure. But we need to understand the future in order to make the right decisions today. The decisions of today co-determine the future.

In other complex situations it is common to refer to trends. This reference to trends is what scientists do, and also, what consultants do. Trends consider the past and the present states of a phenomenon and extrapolate those states to the future. Underlying this approach are three assumptions:

1. that we fully capture and understand the essence of the past states,
2. that we fully capture and understand the essence of the present state, and
3. that we can safely assume a continuity, described by some mathematical function, that estimates the next states sufficiently closely.

Each of these assumptions are rarely valid, even with the best efforts to avoid bias in the selection or presentation of data (Huff, 1954). The data that we have about the past is simply the data that has been collected at a previous point of time, and we can no longer ask for additional observations. The data about the present often comes from selected samples, but here the selection of these samples is critical. In both cases the quality of the data (its representativeness and accuracy) plays a role, and, thus data collection methods come under scrutiny. One promise of “big” data for “smart” cities (Li et al., 2016) is that we might overcome selection bias and thus improve representativeness and accuracy by either taking significantly larger samples than ever before or even by observing a system comprehensively. The third assumption, however, is one that data scientists hardly have control over. We will discuss a range of challenges caused by this assumption in the context of urban mobility, pointing out, on the one hand, that urban mobility is a complex system where interference quickly leads to interdependent and often unintended consequences, and, on the other, that we are experiencing a time of major shifts in technology that may be disrupting to urban mobility as mentioned above (Meyer and Shaheen, 2017).

Thus, while we abstain from making predictions, we will discuss, in this part of the book, the current evidence:

- a review of the present state of urban mobility, in order to recognize the challenges that need to be addressed; and
- the potential of current developments in science and technology, in order to inform the reader about the tools, both these at hand and those that are emerging.

Thus, this book is about change, and more precisely, about enabling agency.

It is worth stating that this book looks at urban mobility, and parking in particular, from the perspective of increasingly “smart” urban environments (we will discuss the term “smart” later in detail). That is to say, we look at the use of historical and current data on space and time in order to become more efficient with our resources: resources such as parking or road space, cruising time, the environmental impact (greenhouse gases), and the impact on human health (particles). The tools at our disposal are:

- Information and Communication Technologies (ICT) providing real-time connectivity of all involved agents,
- Sensor fusion and the Internet of Things (IoT) providing comprehensive localized sensing in both spatial and temporal domains, and
- mapping technologies (geospatial IT) for real-time geospatial data analytics, visualization, and interaction.

Accordingly, this book – and here is our disclaimer – does not consider a whole range of approaches to advance urban mobility and parking that are considered by other scientific disciplines, such as economic approaches (using market mechanisms), transport planning approaches (on transport infrastructure investments), optimization approaches (using control theory for flow improvements), or urban planning and the future of work.

Also, as a second disclaimer, this book is bound to the common good. Focusing on the greater common good, this book deliberately leaves out other aspects such as maximizing particular individuals' or commercial interests. The agency it wants to enable is inspired primarily by the United Nations Sustainable Development Goals, the “blueprint to achieve a better and more sustainable future for all”<sup>1</sup>. The Sustainable Development Goals, stressing the phrase, “for all”, point to the global challenges that humankind faces and struggles to cope with. These truly global challenges – including poverty, inequality, climate change, environmental degradation, peace and justice – will not be overcome by technology alone, but by a change of political will and commitment. But emerging technology can only be used by those willing to use it. It provides opportunities to introduce change in this case to urban mobility, of which, parking is a significant factor. All of the Sustainable Development Goals are interrelated, but the most obvious ones for our focus are (from the same document):

- Goal 11: *Make cities and human settlements inclusive, safe, resilient and sustainable*. The goal is explicitly linked not only to existing cities but to urbanization<sup>2</sup>: “Since 2007, more than half the world’s population has been living in cities, and that share is projected to rise to 60 per cent by 2030”. This goal reminds us that mobility is only one component in another complex system: that of people organizing and optimizing their social, economic, and health needs. And since the city offers more opportunity than rural communities, the global trend to urbanization will continue. In 2018, 4.2 billion people lived in cities, and this number is expected to grow by 2.5 billion by 2050 (United Nations, 2018), i.e., within only one generation. 90 % of this growth is expected to happen in Asia and Africa. This growth, which is linked to the economic opportunities in cities, challenges sustainable development in various ways. It increases over-proportionally the use of resources – cities “account for about 70 % of global carbon emissions and over 60 % of resource use”<sup>3</sup> – and some of the causes lie obviously in urban mobility. Rapid urbanization is also linked to urban sprawl with its negative impact on urban mobility and equitable access in the city. Unintended consequences are, on one hand, a growing number of people of low economic status that feel to be cut off from access, and, on the other

---

<sup>1</sup><https://www.un.org/sustainabledevelopment/sustainable-development-goals/>

<sup>2</sup><https://www.un.org/sustainabledevelopment/cities/>

<sup>3</sup><https://www.un.org/sustainabledevelopment/cities/>

hand, of inadequate and overburdened infrastructure and services, including roads and transport.

- Goal 13: *Take urgent action to combat climate change and its impacts.* The United Nations observes<sup>4</sup>: “Given current concentrations and on-going emissions of greenhouse gases, it is likely that by the end of this century, the increase in global temperature will exceed 1.5 °C compared to 1850 to 1900 for all but one scenario. [...] Global emissions of carbon dioxide (CO<sub>2</sub>) have increased by almost 50 per cent since 1990. Emissions grew more quickly between 2000 and 2010 than in each of the three previous decades.” And yet, with decisive climate action, it “is still possible, using a wide array of technological measures and changes in behavior, to limit the increase in global mean temperature to two degrees Celsius above pre industrial levels”.

As the UN highlighted in its UN World Urbanization Prospects 2018, “urbanization is a complex socio-economic process that transforms the built environment, converting formerly rural into urban settlements, while also shifting the spatial distribution of a population from rural to urban areas. It includes changes in dominant occupations, lifestyle, culture, and behavior, and thus alters the demographic and social structure of both urban and rural areas. A major consequence of urbanization is a rise in the number, land area, and population size of urban settlements, and in the number and share of urban residents compared to rural dwellers.” And it continues: “An increasing share of economic activity and innovation becomes concentrated in cities, and cities develop as hubs for the flow of transport, trade and information. Cities also become places where public and private services of the highest quality are available and where basic services are often more accessible than in rural areas.” The global trend towards urbanization will just add pressure on mobility, even if the individual demand for a private car and parking might diminish.

## Bibliography

Huff, D. (1954). *How To Lie With Statistics*. W. W. Norton & Company.

Li, S., Dragicevic, S., Anton, F., Sester, M., Winter, S., Coltekin, A., Pettit, C., Jiang, B., Haworth, J., Stein, A., and Cheng, T. (2016). Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115(May):119–133.

Meyer, G. and Shaheen, S. A. (2017). *Disrupting Mobility – Impacts of Sharing Economy and Innovative Transportation on Cities*. Springer International, Cham, Switzerland.

---

<sup>4</sup><https://www.un.org/sustainabledevelopment/climate-change/>

Riggs, W. (2018). *Disruptive Transport: Driverless Cars, Transport Innovation and the Sustainable City of Tomorrow*. Equity, Justice and the Sustainable City. Routledge, London.

United Nations (2018). World urbanization prospects: The 2018 revision. Report, Department of Economic and Social Affairs, Population Division.



## 2 Reference Systems for Urban Mobility

BALASUBRAMANIAN NAGARAJAN

### Abstract

In the study of urban mobility, understanding the contribution of geospatial data and the relevant geospatial technologies associated with the collection, storage, and manipulation of geo-referenced data is essential. Correct understanding of the reference frames used for the collection of geospatial data ensures that the integrity and accuracy of the data collected is maintained, keeping in mind the surveying principle, 'whole to part'. It also ensures that the errors involved in the collection of data are well within the accuracy range expected for the particular scale of mapping, which may then be used for computations of distance or area. Geospatial data is collected in a three-dimensional space and is converted to a two-dimensional space for many practical applications. A large range of map projections are available for this conversion which, again, maintain different cartographic properties for any specific application. The Universal Transverse Mercator (UTM) is one such projection, which has properties that come in handy for our problem of mapping for urban mobility. Global Navigation Satellite Systems (GNSS) and their integration with cellular network infrastructure have caught the imagination of people and served as the inspiration for a wide spread of applications such as automatic vehicle location, tracking systems, navigation, pedestrian navigation systems, intelligent transportation Systems, and precise positioning of emergency callers, all using a location in some reference system.

### Keywords

Reference frames, geoid, geoid undulation, ellipsoid, orthometric and ellipsoidal heights, map projections, UTM, GNSS

## 2.1 Reference Frames and Map Projections

### 2.1.1 Introduction

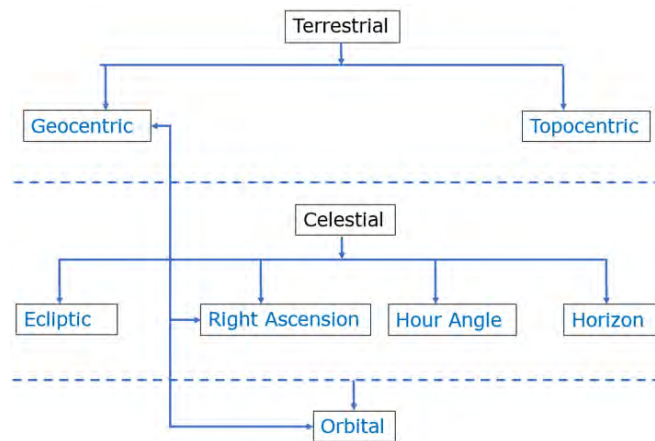
Spatially distributed data, when geo-referenced, is called geospatial data. The locations of these geospatial data points on the surface of the Earth are identified in terms of coordinates in some reference system. The reference system (or coordinate system) is chosen by pragmatic considerations of the application: locations can be identified in planar coordinate systems (2D) approximating the local surface of the Earth, or in three-dimensional (3D) Cartesian, or curvilinear coordinate systems. In the planar coordinate system, the horizontal coordinates

[https://doi.org/10.34727/2021/isbn.978-3-85448-045-7\\_2](https://doi.org/10.34727/2021/isbn.978-3-85448-045-7_2)

This chapter is licensed under a Creative Commons Attribution-ShareAlike 4.0 International licence.

are represented, typically, as  $(X, Y)$  or also as  $(Easting, Northing)$ . In the three-dimensional Cartesian coordinate system coordinates are typically called  $(X, Y, Z)$ , and in the geodetic coordinate system as  $(Latitude \phi, Longitude \lambda, H)$ , which can refer either to a sphere or to an ellipsoid. Whatever the reference frame used for defining the position of a point on the surface of the Earth, one should understand that the coordinates can be transformed from one system to another by mathematical relations.

Requirements for defining the coordinate systems, whether in 2D or in 3D, are specifications of the location of the origin, the orientation of the axes, and the parameters that define the mapping, or mathematical transformation, from a Cartesian system to the chosen coordinate system. Depending upon whether we consider a static Earth or a rotating Earth, and also, whether we want to consider only the diurnal motion or the earth's revolution around the sun in the solar system, the coordinate systems are further identified as *terrestrial*, *celestial*, and *orbital* coordinate systems. Orbital coordinate systems refer to the measurements that use artificial satellites around the Earth. Though we use satellite data such as Global Navigation Satellite Systems (GNSS) or remote sensing imagery in urban mobility, the discussion in this chapter will be only on terrestrial coordinate systems. Depending on the chosen location of the origin of the coordinate system, terrestrial coordinate systems are classified as *geocentric* and *topocentric* coordinate systems. Figure 2.1 classifies the various coordinate systems used in geodesy.



**Figure 2.1:** Various types of coordinate systems used in geodesy.

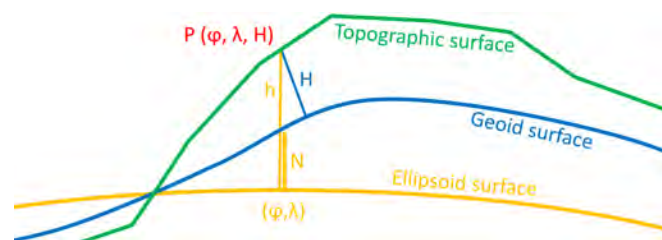
### 2.1.2 Horizontal Datum

When we look at the Earth from a distance, say from the International Space Station, the shape of the Earth seems to be perfectly spherical. But in reality, it

has an undulating surface, making a spherical coordinate system unsuitable for referring to a point on the Earth's surface, or for computing distances or areas. In order to come closer to the shape of the Earth the spheroid is often replaced by a reference ellipsoid. A reference ellipsoid is a mathematical surface described by a few mathematical parameters. The center of the ellipsoid is chosen to coincide with the center of mass of the Earth, and hence, the ellipsoid also is referred to as *geocentric*, or globally best fitting ellipsoid.

Since the surface of the actual Earth varies considerably, with high mountains, valleys, plains, deserts, forests, and coastal lines, another approximation, the *geoid*, has been introduced to represent the natural Earth surface even more closely. A geoid is defined as an equipotential surface: a constant geopotential on the surface, such that gravity is perpendicular at every point. Practically, the geoid surface is considered to coincide with the oceans' surface at rest. Since the oceans' surface is never at rest, this model is realized through *mean sea level* (MSL) measurements. In most practical applications, mean sea level is used synonymously with the geoid surface, although the MSL and the Geoid are still separated by anomalies expressed by *sea surface topography* (SST) of a magnitude of  $\pm 1$  to 2 meters.

The reference ellipsoid and the geoid surfaces are separated by *geoid undulations*  $N$ . Figure 2.2 illustrates the various surfaces of interest: the geoid, the reference ellipsoid, and the topographical surface. As one can observe, for the point  $P$  on the surface of the Earth (the topographic surface), its horizontal coordinate  $(\phi, \lambda)$  is referred to on the ellipsoid ( $Q$ ) and its height  $H$  is measured above the geoid ( $R$ ).



**Figure 2.2:** Relationships between the Earth's topographic surface, the geoid, and the reference ellipsoid.

### 2.1.3 Vertical Datum

Any point on the surface of the Earth can be described by a point on a reference ellipsoid using the two coordinates,  $\phi$  and  $\lambda$ , and a vertical coordinate or height. Height, however, is typically not measured above the ellipsoid for a pragmatic reason: the water flow criterion. We expect that wherever water flows, it does from a *higher* point to a *lower* point (or conversely, from lower potential to higher

potential). Thus, points at the same height should have no potential difference, and this is the reason why the geoid, as the surface of constant geopotential, is chosen to measure height. For example, when we say that the height of Mt. Everest is 8,848 meters, we mean that its height above the Geoid surface (realized by the MSL or otherwise) is 8,848 meters.

The vertical datum for any country has been determined, for a long time, by the long-term mean sea level observations carried out at tide gauges established at the (nearest) coastline of the country. Hence, the mean sea level is bound to a particular tide gauge, for an epoch. It then has been transferred to the *tide gauge benchmarks* established on the land, which have then been used to densify the vertical control work in the country using the method of *leveling*. Heights provided by leveling procedures along with gravity observations are called Helmert's orthometric heights in India. The curvilinear coordinates,  $\phi$  and  $\lambda$ , are computed on the ellipsoid as the horizontal datum, and the orthometric height,  $H$  is computed above the geoid as the vertical datum (Figure 2.2).

#### 2.1.4 Height Systems

Independent of the pragmatic choice of the vertical datum in geodetic coordinate systems, the height of any point on the surface of the Earth can also be measured above the reference ellipsoid. If the height of the point is measured above the reference ellipsoid along the ellipsoidal normal, it is called the ellipsoidal height,  $h$  ( $H$  is called orthometric height). Figure 2.2 shows the relationship between the ellipsoidal height,  $h$  and orthometric height,  $H$ .

While the orthometric height has a physical meaning concerned with water flow, the ellipsoidal height is only a geometrical height of mathematical interest. If the parameters of the ellipsoid are changed the ellipsoidal heights will also change. Since GNSS receivers only provide ellipsoidal heights, we need to convert these ellipsoidal heights to orthometric heights using the geoid undulation values at that point to make them practically useful.

The geoid undulation  $N$  (Figure 2.2) is defined as the separation between the point on an ellipsoid and the corresponding point on the geoid measured along the ellipsoidal normal. Since the geoid surface can be either above the ellipsoidal surface or below, the geoid undulation values can be positive (when the geoid is above the ellipsoid) or negative. Geoid undulation values at any location can be computed with respect to the level ellipsoid using the gravity anomaly values derived from the Stokes' integral (Heiskanen and Moritz, 1967). *Earth gravity models* like EGM96, EGM08 or EGM20, can be used to obtain these geoid undulation values. The conversion of ellipsoidal heights obtained with GNSS observations to practically useful orthometric heights is then simply  $H = h - N$ .

### 2.1.5 Reference Systems Used in Geodesy

As a summary, we mentioned the following geocentric coordinate systems used for referring to a point on the surface of the Earth:

1. Cartesian geocentric coordinate system  $(X, Y, Z)$ .
2. Ellipsoidal coordinate system  $(\phi, \lambda, h)$  where  $\phi, \lambda$  are the geodetic latitude and geodetic longitude respectively, and  $h$  is the ellipsoidal height.
3. Geodetic coordinate system  $(\phi, \lambda, H)$  where  $\phi, \lambda$  are the geodetic latitude and geodetic longitude respectively, and  $H$  is the orthometric height.
4. Astronomical coordinate system  $(\Phi, \Lambda, H)$  where  $\Phi, \Lambda$  are the astronomical latitude and astronomic longitude respectively, and  $H$  is the orthometric height.

These coordinate systems can be transformed from one system to another using mathematical relations. The formulae used for transformations are given below without any derivation (see one of Bomford, 1980; Ahmed, 2006; Heiskanen and Moritz, 1967; Jekeli, 2016; Krakiwsky, 1973; Richardus and Adler, 1972; Snyder, 1983; Vanicek and Krakiwsky, 1987, for more details).

### 2.1.6 Transformation from Cartesian to Ellipsoidal Coordinates

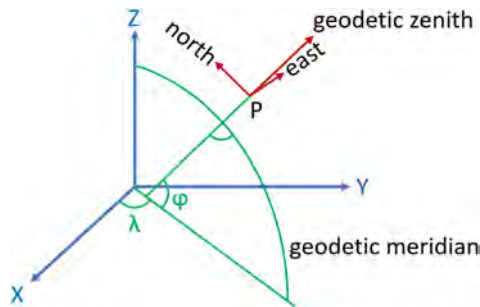
When the coordinates of a point  $P$  are known in a Cartesian coordinate system  $P = (X, Y, Z)$ , its coordinates in an ellipsoidal coordinate system can be computed using the following standard relations. Here,  $N$  is the radius of curvature in the prime vertical section and  $e^2$  relates to the parameters of the reference ellipsoid chosen:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} (N + h) \cos \phi \cos \lambda \\ (N + h) \cos \phi \sin \lambda \\ (N(1 - e^2) + h) \sin \phi \end{pmatrix}. \quad (2.1)$$

For urban mobility applications, however, we may require the data in a topocentric coordinate system.

In Figure 2.3  $X, Y, Z$  represents the geocentric coordinate system, and  $\lambda$  and  $\phi$  describe  $P$ , the origin of a topocentric coordinate system, by ellipsoidal coordinates. In this topocentric coordinate system other points can be described by their coordinates  $(E, N, u)$  with  $E$  being the distance from  $P$  along the East axis,  $N$  being the distance from  $P$  along the North axis, and  $u$  (for 'up') the distance along the axis to the geodetic zenith. The transformation from the geocentric to the topocentric system is shown by the following relation:

$$\begin{bmatrix} E \\ N \\ u \end{bmatrix} = R_2(-90 - \phi_p) R_3(-180 - \lambda_p) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.2)$$

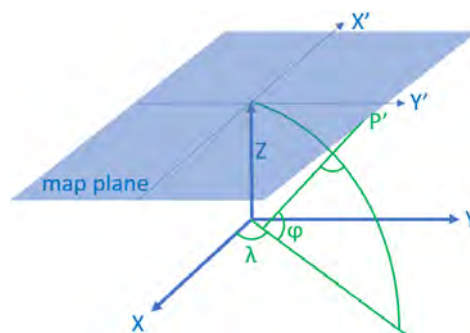


**Figure 2.3:** Local geodetic coordinate system with origin at  $P$ .

where  $\phi_p$  and  $\lambda_p$  are the geodetic latitude and longitude of the point  $P$  in the  $X, Y, Z$  geocentric ellipsoidal coordinate system, and  $R$  is the rotation matrix, given with their respective rotation angles, denoting rotations about the  $Y$  and  $Z$  axis respectively.

## 2.2 Map Projections

A *map projection* is a technique to represent the Earth's curved surface on a planar map. To represent any parts of the surface of the Earth on a flat paper map or on a computer screen, the curved reference surface must be mapped onto the 2D mapping plane. The reference surface for large-scale mapping is usually an ellipsoid. Mapping onto a 2D mapping plane means transforming each point on the reference surface with geographic coordinates  $(\phi, \lambda)$  to a set of 2D Cartesian coordinates  $(x, y)$  representing positions on the map's plane (Figure 2.4).



**Figure 2.4:** A map projection of a point  $P(\phi, \lambda)$  on an ellipsoid to a point  $P'(x, y)$  in a 2D Cartesian coordinate system.

### 2.2.1 Scale Distortions on a Map

Unfortunately, any map projection is associated with scale distortions. There is simply no way to flatten out a piece of ellipsoidal or spherical surface without stretching some parts of the surface more than others. The amount and kind of distortions a map will have depends largely – next to the size of the area being mapped – on the type of map projection that has been selected. For example, in Figure 2.4 the distance of  $P'$  from the tangential point (and origin) of the 2D coordinate system is different from the distance of  $P$  to this tangential point.

Map distortions also vary from location to location, and are often described by local *ellipses of distortion*. The ellipse of distortion, also known as *Tissot's Indicatrix*, shows the shape of an infinitely small circle with a fixed scale on the Earth as it appears when plotted on the map. Every circle is plotted as either a circle or an ellipse, or, in extreme cases, as a straight line. The size and shape of the ellipse shows how much the scale has changed and in what direction.

### 2.2.2 Classification of Map Projections

Map projections can be described in terms of their:

- projection surface (cylindrical, conical, or azimuthal),
- point of secancy to the ellipsoid/sphere (tangent, secant, or polysuperficial),
- aspect with respect to the axis of revolution (normal, transverse, or oblique), and
- distortion property (conformal, equivalent, or equidistant).

The distortion properties of map projections are typically classified according to what is not distorted on the map:

- In a *conformal* (orthomorphic) map projection the distortions of angles between lines in the projected map are identical to the angles between the original lines on the curved reference surface. This means that angles (with short sides) and shapes (of small areas) are shown correctly on the map.
- In an *equal-area* (equivalent) map projection the areas in the map are identical to the areas on the curved reference surface (taking into account the map scale), which means that areas are represented without distortions.
- In an *equidistant* map projection the length of particular lines in the map are the same as the length of the original lines on the curved reference surface (taking into account the map scale).

A particular map projection can have any one of these three properties. No map projection can be both conformal and equal-area. A projection can only be equidistant (true to scale) at certain places or in certain directions.

### 2.2.3 Choosing a Map Projection

Every map production must begin with the choice of a map projection and its parameters. The cartographer's task is to ensure that the right type of projection is used for any particular map. A well chosen map projection takes care that scale distortions remain within certain limits and that map properties match the purpose of the map.

In theory, the selection of a map projection for a particular area can be made on the basis of:

- the *shape* of the area,
- the *location* (and orientation) of the area, and
- the *purpose* of the map. The equidistant cylindrical projection (also called Plate Carrée projection) is a cylindrical map projection with an equidistant property. The map is equidistant (true to scale) along the meridians (in the North-South direction). Both shape and area are reasonably well preserved with the exception of the polar regions.

In summary, the *ideal map projection* for any country would either be an azimuthal, cylindrical, or conical projection, depending on the shape of the area, with a secant projection plane located along the main axis of the country or the area of interest. The selected distortion depends largely on the purpose of the map.

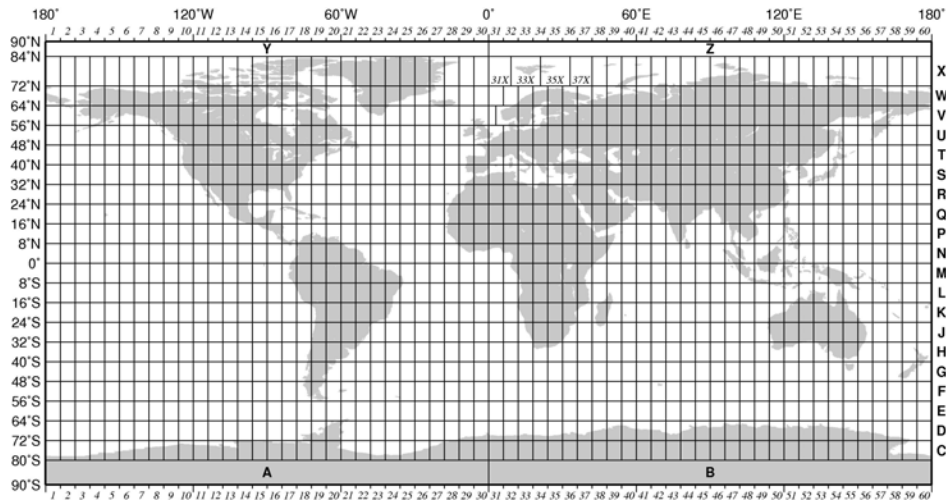
### 2.2.4 Universal Transverse Mercator Projection

In our problem of urban mobility, maps with a scale of 1:10,000 and larger are commonly used, or have been the basis for digital spatial databases. The maps generated with a UTM projection will cause negligible distortions in navigation applications and are hence preferred. Also, webmapping applications often show UTM coordinates of points (mouse positions). GNSS software also provides subroutines for conversion from GNSS coordinates to UTM coordinates directly.

The UTM projection uses a transverse cylinder, secant to the reference surface. The UTM projection divides the world into 60 narrow longitudinal zones of 6 degrees, numbered from 1 to 60. The narrow zones of 6 degrees (and the secant map surface) make the distortions so small that they can be ignored when constructing a map for a scale of 1:10,000 or smaller.



The UTM projection is designed to cover the world, excluding the Arctic and Antarctic regions. The areas not included in the UTM system, i.e., the regions north of 84° North and south of 80° South, are mapped with the *Universal Polar Stereographic* (UPS) projection. Figure 2.5 shows the UTM zone numbering system.

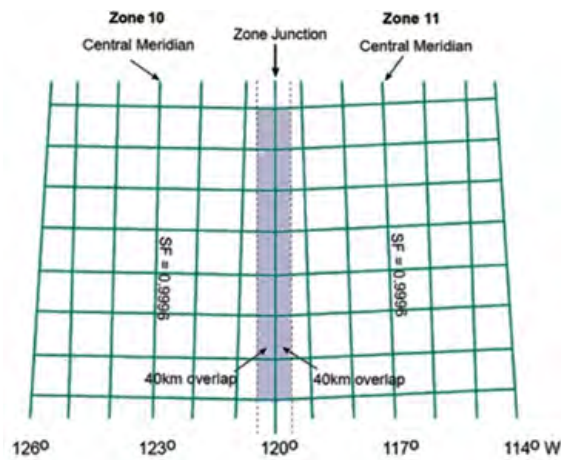


**Figure 2.5:** UTM with defined longitudinal zones. Source: <https://bit.ly/3scguqx> – © Paul Wessel, 2008.

A scale factor of 0.99960 is given to the central meridian of a UTM zone. To avoid negative coordinates for positions located west of the central meridian, the central meridian is given a (false) Easting value of 500,000 m. The equator has been given a Northing value of 0 m for positions north of the equator, and a (false) Northing value of 10,000,000 m for positions south of the equator. Each zone has its own central meridian. For example, Zone 44 extends from 78° East to 84° East. Therefore the central meridian has a longitude value of 81° East.

If a map series covers more than one UTM zone it becomes inconvenient to have the Eastings changing suddenly at a zone junction. For this reason a 40 kilometer overlap into an adjacent zone is allowed (Figure 2.6). Mapping beyond this area will result in larger distortions at the edges of a UTM zone which may not be acceptable for the larger map scales.

If a map series covers more than one UTM zone it becomes inconvenient to have the Eastings changing suddenly at a zone junction. For this reason a 40 kilometer overlap into an adjacent zone is allowed (Figure 2.6). Mapping beyond this area will result in larger distortions at the edges of a UTM zone which may not be acceptable for the larger map scales.



**Figure 2.6:** Two adjacent UTM zones of 6 degrees longitude with a 40 km overlap into the adjacent zone. Source: <https://bit.ly/3cbf3mX> – © Richard Knippers, 2009.

## 2.3 The Basics of Global Navigation Satellite Systems

GNSS and their integration with cellular network infrastructure have inspired a wide range of applications such as automatic vehicle location, tracking systems, navigation, pedestrian navigation systems, intelligent transportation systems, and precise positioning of emergency callers, all using a location in some reference system.

### 2.3.1 What is GNSS?

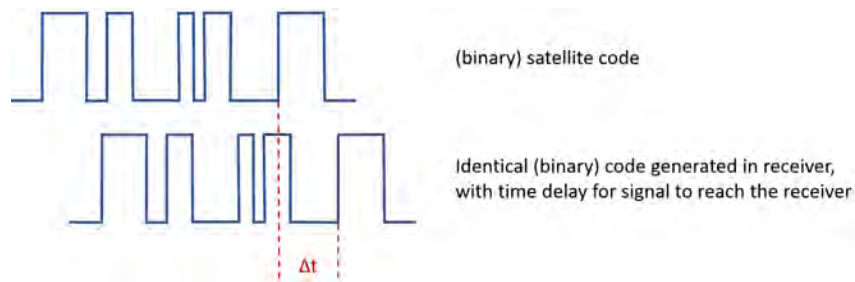
GNSS (see also Chapter 4) is a constellation of satellites providing signals from space that transmit positioning and timing data to GNSS receivers. These receivers then use this data to determine their location. Presently GNSS contains four different constellations having global coverage such as GPS (USA), GLONASS (Russia), Galileo (European Union), BeiDou (China), and two having regional coverage, IRNSS (India) and QZSS (Japan). Satellite based augmentation systems such as WAAS, EGNOS, MSAS and GAGAN also form part of GNSS.

### 2.3.2 Basic Observables of GNSS

Using GNSS, we can have the following observables: a) pseudo range measurements, and b) carrier phase measurements.

### 2.3.2.1 Pseudo Range Measurements

Pseudo range measurements between the GNSS pseudo ranges is a measure of the range or distance between the receiver and the satellite. The range is measured by multiplying the time delay  $\delta t$  for the signal to arrive at the receiver from the satellite and the velocity of light in vacuum  $c$ . Since the GNSS signal travels through the ionosphere and the troposphere before reaching the receiver, and there is also a clock error in computing the time delay, the range measured is called a pseudorange and not a range. Figure 2.7 explains the pseudorange measurements.



**Figure 2.7:** Pseudorange measurement using the formula  $R = c\delta t$ , where  $c$  is the velocity of light in vacuum and  $\delta t$  time delay for the signal to reach the receiver.

For an epoch, the pseudoranges,  $R_i$ , from  $n$  visible satellites are measured. Since the coordinates of the visible satellites are given (they are known from the broadcast ephemeris received from the relevant satellites) they need at least four visible satellites to solve the system of  $i$  equations for the variables  $(x, y, z)$  (the unknown observer position), and  $b$  (the receiver clock bias):

$$R_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - b \quad i = 1, \dots, n. \quad (2.3)$$

Since the equation for estimating the location of the GNSS receiver is quadratic, the equations have to be linearized and the unknown parameters are estimated using the least squares adjustment and an iterative technique.

### 2.3.2.2 Accuracy of Navigated Position using PseudoRanges

The accuracy,  $P_a$ , of the estimated observer position using pseudorange measurements are given by the relation:

$$P_a = UERE \times DOP \quad (2.4)$$

where  $UERE$  (User Equivalent Range Error) refers to the accuracy to which the pseudoranges can be measured (empirically, a value of  $1/100^{th}$  of the wavelength of the signal used is assumed), and  $DOP$  (Dilution of Precision) refers to

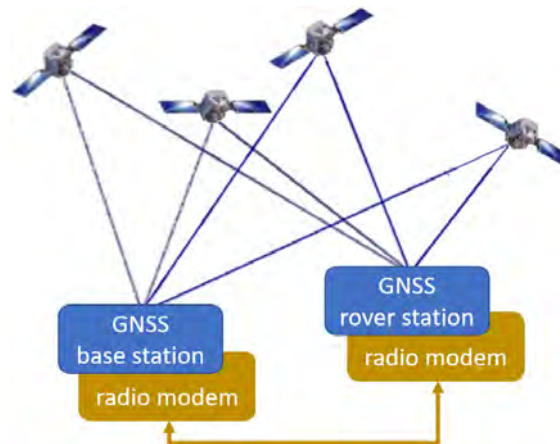
the geometry of the visible satellites used for estimating the position (usually a maximum of GDOP = 5 is used). In such a scenario, if we use the P or Precision code of the GPS, we may get a positional accuracy of 1-2 meters, and if we use the C/A or Coarse Acquisition Code, we will get an accuracy of 10 meters. Since the US government has introduced an anti-spoofing technique for intentionally degrading the accuracy of the GPS, normal GPS users can only use C/A code for measuring the pseudoranges, and hence can get the positional accuracy in the range of 10 meters only in real-time applications such as navigation. However, since GNSS provides access to more satellites than only GPS, the DOP value can be reduced to get an accuracy of 6 to 8 meters for real-time position estimation. The errors involved in the GNSS measurements include satellite orbital errors, satellite clock errors, ionospheric errors, tropospheric errors, receiver clock errors, antenna offset errors, and multipath errors. Though most of these errors can be modeled or minimized, we still get a positional accuracy in the order of meters.

### **2.3.2.3 Differential GNSS Technique**

*Differential GNSS* is a technique that has been introduced to completely remove/minimize the above errors so that positional accuracy in navigation can be improved to 1 to 2 meters. In this technique, a reference receiver is placed at a known station whose coordinates are accurately known. The rover (user) receiver is moved to different locations where the positional coordinates are to be estimated. The reference receiver computes the difference between its known coordinates and the position computed from GNSS observations at a set epoch interval, which we call differential corrections. The epoch time and the differential corrections are transmitted from the reference receiver location using UHF/VHF frequency, so that the rover, which computes its position from the available GNSS satellites to an accuracy of 6 to 8 meters, now receives the differential corrections from the reference receiver and corrects its position to an accuracy of 1 to 2 meters. Figure 2.8 illustrates the differential GNSS technique. If differential corrections transmitting facilities are not available, the corrections can be provided off-line for post-processing.

### **2.3.2.4 Satellite Based Augmentation Systems**

When only a few stations need to receive differential correction for improving their positional accuracy, DGNSS (or Ground Based Augmentation Systems (GBAS)) are deployed. However, if these differential corrections have to be transmitted to a larger area, of the size of a country or continent, then WAAS for the North American countries, EGNOS for the European Union, GAGAN for India, MSAS for Japan, are a few of the Satellite Based Augmentation Systems (SBAS) which are presently operational and provide DGNSS corrections more widely. Commer-



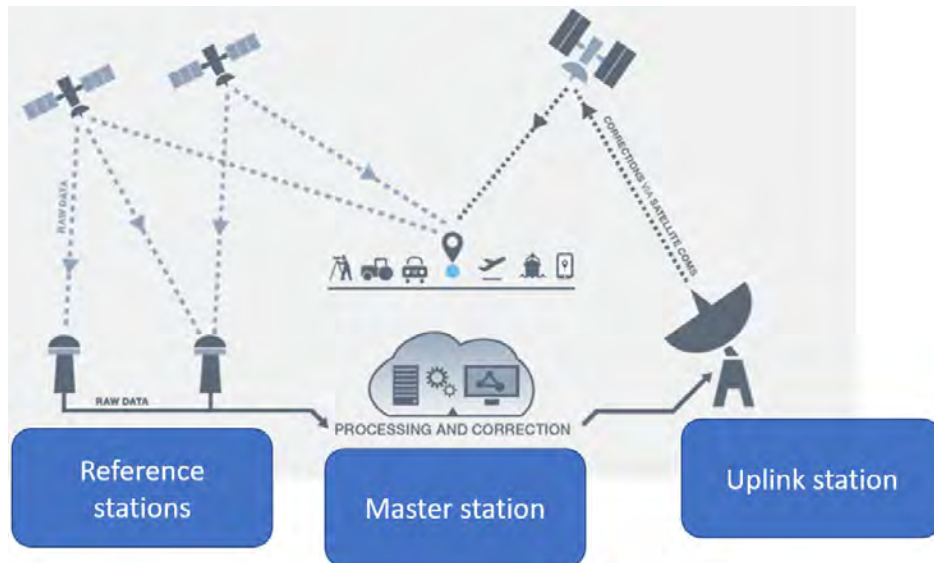
**Figure 2.8:** Differential GNSS technique.

cial vendors also provide differential corrections on payment basis for improving the navigational position accuracy. Figure 2.9 illustrates the working of an SBAS in a region.

### 2.3.2.5 Carrier Phase Measurements

Another way of measuring the ranges to the satellites can be obtained through the carrier phases. The range would simply be the sum of the total number of full carrier cycles plus fractional cycles at the receiver and the satellite, multiplied by the carrier wavelength (see Figure 2.10). The ranges determined with the carriers are far more accurate than those obtained with the codes (i.e., the pseudoranges). This is due to the fact that the wavelength (or resolution) of the carrier phase, 19 cm in the case of GPS L1 frequency, is much smaller than those of the codes. There is, however, one problem. The carriers are pure sinusoidal waves, which means that all cycles look the same. Therefore, a GPS receiver has no means to differentiate one cycle from another. In other words, the receiver, when it is switched on, cannot determine the total number of complete cycles between the satellite and the receiver. It can only measure a fraction of a cycle very accurately (less than 2 mm), while the initial number of complete cycles remains unknown, or is ambiguous.

This method is, therefore, commonly known as the initial cycle ambiguity, or the ambiguity bias. Fortunately, the receiver has the capability to keep track of the phase changes after being switched on. This means that the initial cycle ambiguity remains unchanged over time, as long as no signal loss (or cycle slips) occurs. Once the initial cycle ambiguity parameters are resolved, accurate range measurements can be obtained, which lead to accurate position determination. This high accuracy positioning can be achieved through the so-called relative

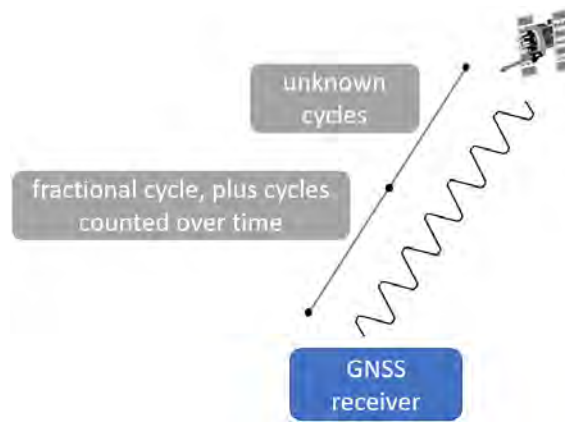


**Figure 2.9:** An SBAS system providing differential GNSS corrections for a large area (base graphic provided by Geoscience Australia; modified).

positioning techniques, either in real time or in post-processing. Unlike in navigational positioning using pseudoranges, where one receiver is enough to get the position, this method requires two GPS receivers simultaneously tracking the same satellites in view for determination of baseline length and subsequently the coordinates of the rover station. The routine accuracy one can get in measuring baseline length is in the order of 1 ppm, i.e., 1 millimeter accuracy in a baseline length of 1 kilometer. Compared to the navigational receivers, survey grade receivers capable of carrier phase measurements are expensive, where the complicated post-processing software itself takes a greater share in the cost.

## 2.4 Conclusions

Geospatial data analysis is an excellent tool in the planning and execution of urban mobility related projects, especially in cities in low and middle income countries which experience shortage of space, undisciplined traffic movements, and unorganized parking of vehicles. Geospatial technologies such as geographic information systems with map data, or global navigation satellite systems (and others that we have not discussed here) provide ample assistance and tools for scientific analysis of available resources and optimal planning in urban mobility. City planners understand the trade off between accuracy required and the cost of the technique. With so many options available one needs to understand the requirements correctly and apply the right tools to get the right results.



**Figure 2.10:** GNSS positioning using carrier phase.

## Bibliography

- Ahmed, E.-R. (2006). *Introduction to GPS: The Global Positioning System*. Artech, second edition edition.
- Bomford, G. (1980). *Geodesy*. Oxford University Press, Oxford.
- Heiskanen, W. A. and Moritz, H. (1967). *Physical Geodesy*. W.H. Freeman and Company, San Francisco.
- Jekeli, C. (2016). *Geometric Reference Systems in Geodesy*. Ohio State University.
- Krakiwsky, E. J. (1973). *Conformal Map Projections in Geodesy*. University of New Brunswick, Fredericton, New Brunswick.
- Richardus, P. and Adler, R. K. (1972). *Map projections for geodesists, cartographers and geographers*. North-Holland Pub. Co, Amsterdam, NL.
- Snyder, J. P. (1983). *Map projections used by the U.S. Geological Survey*. Geological Survey bulletin. U. S. Government Printing Office, Washington, DC.
- Vanicek, P. and Krakiwsky, E. J. (1987). *Geodesy: The concept*. North-Holland, Amsterdam.

## 3 Tracking Urban Mobility

STEPHAN WINTER

### Abstract

Within the coordinate reference systems discussed in the previous chapter, location can be described. Location data is increasingly becoming available from sensors integrated in urban mobility: sensors that are attached to travelers or vehicles, or even to fix locations registering travelers or vehicles passing by. This chapter will introduce some tracking technologies and their properties, and then define the notion of a trajectory, with its critical properties of spatial and temporal granularity (precision and sampling rate), and accuracy (linked to map matching). In addition, the chapter introduces the two complementary frames of references for tracking urban mobility, the Lagrangian and the Eulerian, and how to convert between them.

### Keywords

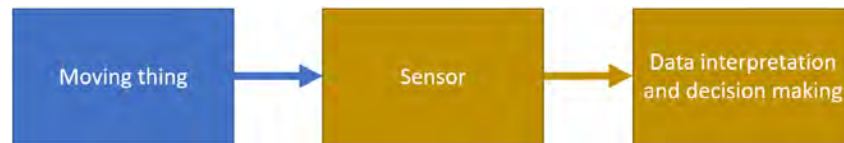
Internet of things, IoT, location, tracking

### 3.1 Introduction

Tracking urban mobility is key to any smart interaction with mobility, inclusive of parking. Tracking relies on three components that must interact in some forms: something that moves, a sensor recognizing and characterizing this movement, and a connection of the sensor to some computing device, typically on board or through the Internet (Figure 3.1). The *moving object* can be, for example, a person in a market hall, tracked for his/her movement through a CCTV camera (the *sensor*), and their trajectory being analyzed for their shopping behavior in the marketplace (the *computing of information out of data*). As another example, the moving object can be a bus, equipped with a satellite positioning tracker, and the trajectory being sent to the operator for providing real-time arrival estimates on displays at bus stops. The same goal can be achieved by a parking sensor that senses whether a vehicle has moved into a parking slot and reports this information to a parking guidance system.

Of the three elements in Figure 3.1, the two yellow ones are usually subsumed as the *Internet of Things*. The Internet of Things is characterized by sensing and computing devices that are embedded in everyday objects, including mobile





**Figure 3.1:** Something that moves, a sensor recognizing and characterizing this movement, and a connection of the sensor to some computing device, typically through the Internet.

ones, and connected via the Internet for the transfer of observed data. While these devices carry unique identifiers, in many applications their location is also critically important for the interpretation of their collected data. For example, if a sensor measures air quality, then, for a proper understanding of its reading, it is necessary to know where this reading has been taken. For a sensor set up at a fixed location in the city this location might be recorded once, at the time of set-up. But sensors embedded in mobile objects must synchronize their readings with the current location of the sensor platform. Therefore, sensors embedded in mobile objects typically are also observing the mobile object's location.

A location is always a location with respect to something else. So this location can be described only in a reference system (Chapter 2). For example, a moving vehicle has embedded sensors that track its distances from the vehicle in front or from the road markings, in order to describe its location in relationship to these other objects. That same vehicle can also carry a sensor to receive satellite signals for triangulating its position within an abstract coordinate system (or you might say, relative to the current locations of the satellites, but then the definition of a location becomes circular). And again, this vehicle can carry an RFID chip that communicates with toll bridges, capturing its location as it passes them. These three examples respectively highlight (1) a dynamic relative reference frame among other moving objects, (2) an absolute reference frame with respect to the Earth, and (3) a static relative reference frame with respect to an anchor location.

This chapter will relate to such reference frames and lay the foundations for observations in these reference frames. At the end of this chapter you should be able to distinguish categories of capturing and describing location over time, which means especially of mobile objects or individuals. You should be aware of errors that adhere to all measurement data, and sensitivities about location data. We will conclude by discussing two major applications of tracking data: travel mode detection and map matching. The framework for collecting, characterizing, and analyzing data discussed in this and the following chapters lays the foundation for *intelligent transportation systems*, or ITS. Accordingly, *intelligent transportation systems subsume all efforts to use data to improve the capacity, safety, and environmental impact of existing transport infrastructure*.

## 3.2 Internet of Things

The Internet of Things (IoT), defined above as sensing and computing devices that are embedded in everyday objects, including mobile ones, and connected via the internet for the transfer of observed data, relies on unique identifiers for the devices and standard protocols to communicate between these devices (ITU, 2014). The devices not only sense, but are also often coupled with mechanical and digital machines to control the physical environment. Two typical, and overlapping application domains for the Internet of Things are *smart cities* and *transport*. For these domains, the International Telecommunication Union's Study Group 20 is fully dedicated to promote interoperability in the Internet of Things.<sup>1</sup> Similarly, the Open Geospatial Consortium (OGC) is working on an "open, geospatial-enabled and unified way to interconnect the Internet of Things' devices, data, and applications over the Web" in what they call the SensorThings API.<sup>2</sup> The emerging OGC standard provides already "a standard way to manage and retrieve observations and metadata from heterogeneous IoT sensor systems".

In the context of urban mobility – the overlap between smart cities and transportation, and a prime example of a heterogeneous system – the Internet of Things provides the observations to connect:

- people to needs, including vehicles;
- vehicles to vehicles;
- vehicles to infrastructure.

In each of these categories a range of relevant applications relies increasingly on the Internet of Things technology.

### 3.2.1 People to Needs

Computing devices in the hands of people – their smartphones, tablets, laptops, wearables, and other connected computing devices – are typically equipped with a range of sensors, and they are connected to the Internet. These devices enable a range of applications connecting people to their needs. Applications relevant for mobility can facilitate more efficient mobility services or even reduce the demand for mobility.

A prominent category of applications that reduce the demand for mobility are those supporting work or study online. An employer working from home does not need to commute, and a school kid in an online teaching program does not

---

<sup>1</sup><https://www.itu.int/en/ITU-T/studygroups/2017-2020/20/Pages/default.aspx>

<sup>2</sup><https://www.ogc.org/standards/sensorthings> – OGC, 2015/2017

require a parent (or a bus) to bring them to school. They achieve their needs without the need to move. Behind these applications there are usually cloud-based platforms that track the identity (and for this purpose also often the location) of their users. Another category of applications is concerned with the coordination between people. Colleagues that commute together (“ride-sharing”) reduce the demand for vehicles on the road; the basis for this coordination is a cloud-based platform that matches requests based on their locations and travel times. Demand-responsive public transport solutions fall into this category as well. They are based on tracking the location, occupancy, and actual travel commitments of vehicles and the location and travel demand of people in order to provide real-time matching. Even mass transport can be improved by Internet of Things technologies, for example, by tracking the occupancy of vehicles on the road and estimating people’s current travel demand, or by tracking the location of vehicles on the road and estimating their arrival times (Verma et al., 2020).

### 3.2.2 Vehicles and Vehicles

Nowadays, vehicles carry an increasing number of sensors, most of which ensure road safety and enable autonomous driving. While autonomy, in the first instance, relies fully on decisions made by computational devices on board the a vehicle, autonomous driving can become even smarter by connecting this vehicle with other vehicles or the infrastructure. Connected autonomous vehicles (CAV) allow for greater situational awareness of the individual vehicle, supporting advanced driver assistance / advanced autonomous driving systems, for example, speed adaptation or early braking fed by greater foresight or sight around street corners. CAV also allows for cooperation (cooperative intelligent transportation systems, C-ITS), such as vehicle platooning, or cooperative parking systems. Cooperative ITS have two additional challenges compared to autonomous decision making:

- One is the *certainty* that the collaborator behaves as expected. If heterogeneous agents cooperate, communication is a basic requirement, but is susceptible to communication failures (e.g., one agent reporting velocity in *mph*, the other taking it for *km/h*). Thus, certainty is a matter of agreeing on standardized protocols and exchange languages (Zheng et al., 2015).
- The other challenge is the *trust* that the collaborator is acting in the mutual interest of all. Any system of heterogeneous agents, in principle, is susceptible to deception (e.g., by agents seeking their own advantage in traffic, or by agents with malicious intent). Trust is therefore a matter of cybersecurity and more difficult to address. Among the approaches to establish trust between agents are: looking for redundancy and assessing all evidence (Shafer, 1976; Truelove et al., 2017), or, vice versa, demanding proof of location (Amoretti et al., 2018).

At the time of writing this book, two vehicle-to-vehicle communication systems (V2V) are competing for the same frequency spectrum: the Wi-Fi based dedicated short-range communication (DSRC), and the cellular based 5G. DSRC is an open standard with a range of about 300 m, while 5G is licensed (with fees to operators) and has a slightly larger range (500 m) and shorter latency, but also a shorter shelf life and a significant reliance on infrastructure. While the development of V2V communication systems has been stalled by these two mutually incompatible systems competing worldwide, we put this debate aside and work with the principles of V2V communication.

In principle, applications using V2V communication pursue *decentralized* solutions, or solutions found between and agreed by peers (Duckham, 2013), compared to *centralized* solutions that are decreed by a central authority. An example of a decentralized application is a speed adaptation system: If vehicles share information with vehicles following them that they are about to brake, then the vehicles that are behind can use this information to adapt their speed early, and to coordinate their slowing down times with each other to achieve a smoother traffic flow. No central authority is required – as any central authority would be overwhelmed by the task of optimizing traffic flow at this level. Centralized applications rely on the sensors on board vehicles as well, which presumes other forms of vehicle communications (vehicle to infrastructure, see 3.2.3). An example for such a centralized application – one where a centralized authority provides benefits from its global overview of traffic – is an internet-enabled car navigation system that is able to reroute vehicles in case of road congestion or closures ahead. In this case, no V2V communication is required. Thus, applications using V2V communication are typically operating locally instead of globally. As a consequence they are achieving locally optimal solutions, but not globally optimal solutions: the vehicles (or drivers) operate with limited knowledge.

Since V2V communication is supporting mostly local applications, the communication can be managed to become spatially and temporally sensitive: broadcasting can be limited to certain ranges and time frames. Several communication strategies have been suggested in the literature. Flooding is the most simple one: every vehicle that receives a message instantaneously re-broadcasts the message. To limit this to local applications, flooding can be limited to a certain range or area: only vehicles that are within this range or area would re-broadcast. Other strategies have been designed to reduce the large redundancy of the flooding strategy but still cover all the vehicles in the intended area, among them, a probabilistic strategy (only a certain percentage of vehicles re-broadcast) and a distance-based strategy (only distant vehicles in the communication range re-broadcast). Also, to maintain a message over certain time frames, and inform vehicles that enter the range or area late, these strategies can be applied periodically.

In the context of this book, we are mostly interested in V2V applications that have an impact on parking pressure. These are applications that improve transport capacity such that there are fewer vehicles on the road, and applications that support cooperative parking behavior. Other applications, such as for safer or for smoother driving will be neglected.

An obvious example of such applications is cooperative parking in a large car park, where some vehicles are leaving their parking spot and others are searching for a parking spot. If a vehicle leaving the car park broadcasts the freed parking spot to vehicles nearby then the searching vehicles receiving these messages can take this information into consideration in their own search strategy. Simulations have shown that cooperative parking in such an opportunistic manner, i.e., with no booking mechanisms through a central authority, reduces the search time for vehicles (Aliedani et al., 2016; Aliedani and Loke, 2019). Since this cooperation relies on the vehicles' abilities to locate themselves (in relation to the free(d) parking spots), and these abilities are degraded in indoor environments such as parking garages, others have worked on localization in parking garages to improve cooperative parking (Balzano and Vitale, 2017).

### **3.2.3 Vehicles and Infrastructure**

Communication technologies such as the Wi-Fi based DSRC and the cellular based 5G can also establish communication channels to base stations installed at fixed locations in the infrastructure. Strictly speaking, cellular-based 5G is bound to such base stations anyway, although, in the narrow sense of vehicle-to-infrastructure (V2I) communication the infrastructure partner is one that interacts with traffic or vehicles directly.

An example of such an interaction is a traffic light that communicates its signal phase and timing information to approaching vehicles. In response, the approaching vehicles can optimize their fuel consumption by adapting their speeds early (Rakha and Kamalanathsharma, 2011). A more advanced example considers bi-directional communication at traffic lights. This way, a smart controller in the traffic light can optimize the traffic flow based on the number of approaching vehicles from various incoming directions (Bento et al., 2012). Plenty of similar applications can be thought of. They all concern coordination bound to a location or neighborhood, such as informing about roadside or surface conditions, ephemeral events on the immediate road network neighborhood, or condition of the supporting infrastructure. One example of this is the dynamic reallocation of lanes depending on the current traffic demand: the lane direction can be quickly switched in response to a controller in the roadside infrastructure that observes traffic sensors and optimizes road use (Hausknecht et al., 2011): these instantaneous switches either need dynamic signage (for human drivers) or V2I communication (for CAV).

Parking is another example where knowledge bound to a location and an interaction via V2I is beneficial. Parking lots and parking garages are prime cases for a smart infrastructure that guides vehicles to empty (or allocated) parking spots. The occupancy of parking spaces can be observed by sensors (Chapter 10). Then, a controller in the infrastructure can take these observations and the requests from vehicles searching for a parking space and optimize the allocation of spaces (Geng and Cassandras, 2012) (Chapter 9). In this way, automated valet parking becomes feasible Löper et al. (2013); Banzhaf et al. (2017): a CAV, arriving at the valet bay of the parking garage, lets the passengers disembark, and is then guided by V2I to an allocated parking space inside the garage. When the passengers later return to the valet bay, they “call” their vehicle through an app. In order to succeed, both the vehicle and the infrastructure are not only communicating with each other (as are the passengers, through their app), but are also sensor-rich platforms: the vehicle, for its autonomous driving, and the garage, for observing the occupancy of its parking spaces. A first trial has been demonstrated successfully in a parking garage in Stuttgart in 2019.

### 3.3 Tracking by Sensors

A large variety of sensors are applied to track what moves in urban spaces. One dimensional barcodes and two-dimensional QR codes are applied mostly to track parcels and goods in urban logistics. Both of these are identifiers. Their location is typically determined by stationary scanners. Using this method, parcels can be checked in at certain stages of their journey. But localization also works in reverse: If the QR code is mounted at a fixed (known) location, then a mobile scanner’s location can be determined. For example, security personnel on their inspection rounds can check in at fixed locations, documenting their presence at particular times.

Radio-frequency identification (RFID) technology applies a similar philosophy but since it is radio-driven it does not need line-of-sight with a reader. RFID identification tags use electromagnetic fields to automatically identify and thus, track objects. RFID technology is used in contactless credit cards (where the location of the pay station is recorded) or in contactless smart public transport cards (where the location of the reader is recorded, and the fare is typically determined). Similarly, electronic toll collection works with active RFID readers: A vehicle equipped with an RFID tag passing the stationary reader is registered with its location and time.

Social media have become a prominent source of tracking users. Although Twitter had turned off (optional) precise georeferencing of messages in June 2019, it still allows usage of references to coarse and nearby places. Other social media, such as Foursquare, offers their users to check-in to places and share this location within their network. The distinction between ‘location’ and ‘place’

is important though. We use *location* so far as a representation of a position in a spatial reference frame (in the form of coordinates), typically derived by some measurements, such as satellite positioning. And we use *place* here for common language references, typically names of places, names of businesses at places, or postal addresses, which would need to be translated into coordinates in a spatial reference frame by a process called georeferencing.

Vision sensors, prominently among them, CCTV cameras, are also applied in urban tracking. Some applications settle for counting moving objects (pedestrians, cars) at particular locations. An example is the City of Melbourne's (Australia) pedestrian counting system, which provides open data<sup>3</sup>. Other applications track moving objects in scenes in order to determine flow or density parameters (Wang et al., 2014). Yet other applications aim at identifying individuals. Identification of vehicles is often done through number plate recognition, and identification of pedestrians through face recognition (Parkhi et al., 2015).

Wi-Fi networks can be used to track connected devices in two ways: by *passive* and by *active* tracking. In passive tracking (or device positioning) the smartphone listens, on each channel, for Wi-Fi access points around, including their individual signal strength, and triangulates between these access points. In active tracking (or network positioning), the smartphones' regular probe requests, which include their MAC address, are registered by the Wi-Fi access points. In this way, the network can track a device, which can then be used for movement analysis (Ruiz-Ruiz et al., 2014). Other radio-based positioning technologies work similarly in principle, such as Bluetooth, Ultra-Wide Band (UWB), and GSM (3G, 4G, 5G).

Another radio-based tracking method, however, uses only one-directional communication: the Global Navigation Satellite Systems, or GNSS (see Chapter 2). These systems rely on triangulation methods based on satellite signals. Receivers for satellite signals come in many shapes; prominent to urban mobility are trackers (GNSS tracking device and SIM card, attached to a moving object) and smartphones. The receivers used in urban mobility are relatively cheaper and have inaccurate antennas, such that, for example, autonomously driven cars cannot rely only on GNSS for their localization. But for the purposes of tracking movements, traffic, and route planning this accuracy is sufficient. GNSS belongs to a category of passive tracking methods since the positioning happens on board the mobile sensor platform (e.g., smartphone). Navigation systems on board vehicles, working with off-line maps, can access those GNSS localizations directly. But many other uses of the tracking data, including online navigation systems, require the integration of localizations of many moving agents (people or vehicles) in real-time, and in these cases, the locally produced tracking data has to be shared with a platform via a mobile internet. Other uses for tracking data are fleet management, car insurance, electronic logbooks, live alerts (speeding, servicing, area violations), or automatic emergency calls.

---

<sup>3</sup><http://www.pedestrian.melbourne.vic.gov.au/>

### 3.4 Tracking Data Reference Frames

Moving objects can be observed theoretically in two ways: from a stationary viewpoint as the objects pass by, or from an accompanying viewpoint (Laube, 2014; Both et al., 2012). This categorization is borrowed from fluid dynamics, where the two viewpoints have been labelled the Lagrangian and the Eulerian frames of reference (Hirt et al., 1974; Bennett, 2006). The sensors discussed above fall in one or the other category, and thus, these two concepts will help us to categorize and better understand the observations in urban traffic management, including parking.

#### 3.4.1 Lagrangian Frame of Reference

In the Lagrangian frame of reference, the observer follows an individual particle (in a fluid) as it moves through space and time. Consider now the particle being an object in urban traffic. Then the Lagrangian observer would follow this object and record its locations over time. The result is a *trajectory*.

Lagrangian observations are typically discrete, i.e., taken at certain points in time, but those observations should be frequent enough to reconstruct the continuous movement  $(x, y, t)$  for any  $t$ . If this frequency is lacking, the reconstruction becomes ambiguous with regard to  $(x, y)$ . This discussion on frequency and ambiguity is relevant in the context of map matching, which is the reconstruction of a movement along a transport network on a map. In order to avoid ambiguity, observations are often made at regular intervals (e.g., GNSS recordings of a smartphone every 5 seconds) or in adaptive sampling rates (e.g., GNSS records only if the smartphone has been moved).

#### 3.4.2 Eulerian Frame of Reference

In the Eulerian frame of reference, the observer focuses on specific locations in space through which a particle (in a fluid) passes. Consider again, the particle being an object in urban traffic. Then the Eulerian observer will register the passing of this object at a particular location. Tracking the number of vehicles crossing an intersection, automatic toll collection, or a pedestrian counting system are examples of Eulerian observations.

Eulerian observations, since they are taken at fixed checkpoints, are typically made by sensors installed in the environment, such as beam counters, smart card terminals, RFID readers, CCTV cameras, or the vehicle counting sensors in traffic control systems (e.g., the induction loops of SCATS, <https://www.scats.com.au/>). These sensors observe continuously.



### 3.4.3 Eulerian-Lagrangian Transformations

Often observations exist in one reference frame, but interpretations are sought in another reference frame. In this situation transformations between the two reference frames are required (Hirt et al., 1974; Wang et al., 2016). Chapter 2 has already introduced coordinate transformations between two coordinate reference systems but here we extend this conversion to transformations of observations in Eulerian and Lagrangian frames of reference.

Eulerian to Lagrangian transformations require recombining trajectories from traffic counts or flow data. Since elementary information about identity has been lost in the counts in a Eulerian reference frame, this transformation can only come up with an estimation of likely or representative trajectories. For example, a shopping mall that wishes to guide movement by popular routes. Tracking individuals may be infeasible due to privacy concerns (or privacy legislation), but the shopping mall operator can count the flows from one sub-area to another, for example, by installing beam counters. The density of traffic at particular beam counters can be reconstructed to identify popular routes.

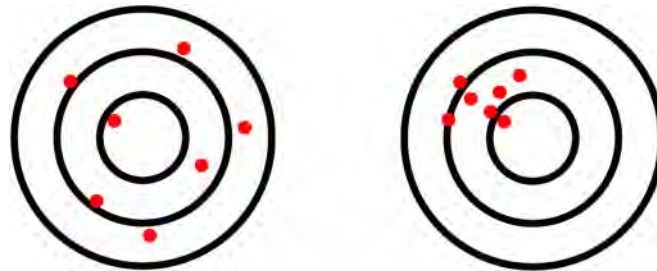
Vice versa, Lagrangian to Eulerian transformations require transforming trajectories into count data. For example, a road authority may have access to household travel survey data: data where a representative sample of the population provides information of their daily travel routines. This data is Lagrangian in nature. But the road authority might be interested in investigating the traffic load at specific intersections, and therefore, converts the Lagrangian data into Eulerian data. A routine way of doing this is using a traffic simulator.

## 3.5 Properties of Data

The tracking data collected by any technology shows a range of properties that have been specified as data quality components (Veregin, 2005). Here we discuss uncertainty, currency, and frequency.

### 3.5.1 Uncertainty

Measurements can never be exact. This is the reason why measurements are usually repeated. Repeated measurements allow subsequent statistical post-processing to balance out random errors (Figure 3.2 left), and also to identify and filter out outliers. However, uncalibrated instruments can also produce biased measurements where the mean is no longer a good estimate of the true value (Figure 3.2 right). Measurements with only random errors are called accurate, while measurements with low variance are called precise. Note that, in line with Figure 3.2 on the right, precise measurements can be inaccurate.



**Figure 3.2:** Measurements can never be exact. On the left, more accurate but less precise measurements, on the right less accurate but more precise measurements.

Take, for example, the capacity of a smartphone to position itself under the open sky, using the signals from the GNSS. A smartphone's GNSS antenna takes a large number of uncertain observations (after all, smartphones use cheap chips) and averages out the random errors. How accurate the result is, i.e., how close the result is to the true location of the smartphone, depends on the impact of systematic errors, such as currently weaker configurations of satellite positions, or multipath effects in urban canyons. Systematic errors cannot be detected from observations alone, and thus, statistical measures such as the standard deviation of repeated observations only describe the precision of a measurement, but not its accuracy. Accordingly, avoiding or controlling systematic errors is critical for measurements, because only then is the most prominent statistical measure, the *mean*, close to the true value. Your favorite mapping app shows a blue circle, centered on the mean, and of a size describing the precision of the measurement. Since mapping apps are commercial services, they are not too transparent about the meaning of these circles, but it is likely that they are linked to some confidence interval, which would be computed from the standard deviation.

### 3.5.2 Data Currency

Many applications of intelligent transportation systems, such as vehicle control, depend on real-time data. However, since communication channels are involved between the observation itself (the sensor), the analysis of the observation (e.g., a cloud-based service), and the use of the derived information (e.g., by a vehicle, or by a driver), real-time operations can only be realized with some latency. In the above example of a mapping app, if the sight to satellites is lost for a while – for example, because a pedestrian walks under dense tree foliage, or a car drives through a tunnel – the smartphone can only show the last known position, which over time gets more and more out-of-date. Hence, data currency has to be actively tracked and considered in the design of applications.

### 3.5.3 Frequency

Many observations in intelligent transportation systems are not made continuously, but with regular or irregular frequency, or the sampling rate. For example, smartphone navigation apps sample the position of a smartphone every couple of seconds – the exact intervals may adapt to current speed and travel mode. A public transport smartcard tracks a person’s movement only with check-in and check-out. Frequency has an impact on the interpretation of observations. If, for example, a car’s location is known every ten minutes, the route it has traveled can be reconstructed only with some ambiguity. Accordingly, frequency is considered as another data quality component.

## 3.6 Privacy Implications of Transport Data

Politico titled a story in 2018: “Google is building a City of the Future in Toronto. Would Anyone Want to Live There?” The story refers to Sidewalk Labs’ (Alphabet’s smart city arm) of Toronto’s eastern waterfront redevelopment, which is based on free and fast Wi-Fi in order to track everything that moves and their activities. From a service provisioning perspective, developments like the one above, or LinkNYC’s Hudson Yard redevelopment, open new opportunities for intelligent mobility. Resistance in the population, on the other hand, comes from the potential secondary use of the data, and hence lack of trust. The jury is out on whether these developments move towards smart cities or surveillance cities.

In principle, societies – always trailing behind technological developments – have to develop legal licensing frameworks and social licensing frameworks for the use of data that impacts privacy. Legal frameworks protect the fundamental values and ethical norms of a society, and are, thus, required to protect the weaker party, or the vulnerable members of the community. It does not help that data has become global while legal frameworks are still formed at national levels. Legal frameworks can demand of making data anonymous before reuse. Anonymous data, however, has also been shown to be susceptible to re-identification (Culnane et al., 2019), such that stronger regulations are needed. Social licenses, in contrast, are licenses given by individuals on the use of their data (Carter et al., 2015). Social licenses imply, first, that the individual is – legally and technically – the owner of their own data, i.e., can control its use. Then the individual might give selective permissions (consent) for uses of their data proposed by a data custodian. This consent of the owner for producing intelligent (transport) services has also been called co-creation.

### 3.7 Trips, Segmentation, and Map Matching

Motorized forms of mobility in the city always require some multi-modality. A person has to at least walk to a (private or public) vehicle, and from the vehicle to their destination. Further mixes are possible, for example, a person taking public transport may transfer between vehicles or modes. In urban transportation we call a movement between two stationary activities a *trip* (Das and Winter, 2016). Hence, a trip consists of a sequence of movements in specific travel modes that are taken with no intended interruption for a planned activity, i.e., including wait times. Correspondingly, a day can be partitioned into *trips* and *activities* between these trips.

Trips and activities between these trips are also the common units of household integrated travel and activity surveys (Roddis et al., 2019; Stopher et al., 2007). The survey data is rather abstract – a typical entry would be, for a member of a household: “7:00 a.m.–7:30 a.m. travel from home to work”. Trajectory data from intelligent transportation systems can provide more details about this trip (Carrion et al., 2014), by splitting the trip into individual segments, for example, by travel modes, and breaks, including wait times. This process of travel mode detection produces segments (of single modes) by some common properties in the trajectory. As it is based on actual trajectory data, it typically provides more accurate data than the surveys themselves (Bricka and Bhat, 2006; Zhao et al., 2015). For example, the trip above could be captured by high-frequency GNSS and inertial sensor observations on a participant’s smartphone and then interpreted by segments of “7:02 a.m.–7:11 a.m. walking; 7:11 a.m.–7:13 a.m. stationary near/at a bus stop (waiting); 7:13 a.m.–7:24 a.m. on a bus on Line 76; 7:24 a.m.–7:26 a.m. walking”. An integration of this data with public transport real-time tracking data could additionally reveal the specific bus or vehicle, which, according to smartcard data interpretation, was crowded at that time. Further refinements are possible, although not often needed in intelligent transportation systems. For example, the waiting time may have involved some wandering around to cope with the cold weather. The segmentation process is also, in principle, a hierarchical one, since each segment can be split into further segments in more detail. For example, embarking a bus – switching travel mode from walking to riding on a bus – does not happen in an instant, but could be split into queuing, embarking, ticketing, and walking to a seat (Das and Winter, 2016).

#### 3.7.1 Mode Detection

Partitioning and labelling the segments of a trip by travel mode are often the first steps of semantic trajectory analysis, which in general aims to attach meaning to connected sequences of  $(x, y, t)$  triplets forming a trajectory (Parent et al., 2013). To identify a segment of a particular travel mode commonly high-frequency GNSS

observations on a traveler's smartphone are available (sequences of  $(x, y, t)$ ). Additional inertial sensor observations or compass measurements enrich the interpretation process and reduce its uncertainty.

Mode detection can be divided into the steps of detecting discontinuities in the trajectory (indicating modal change), and then labelling the segments by travel modes or activities. According to this order of processing, real-time mode detection is more challenging because of the real-time detection of a discontinuity.

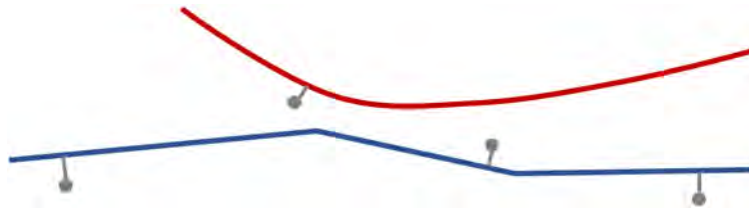
The labelling of a segment can be formulated analytically. For example, a fuzzy set classification method uses rules such as "if the movement along a segment between two stops is never going faster than 5 kilometers per hour, and generally located on sidewalks, then this is walking" (Das and Winter, 2018). Other mode detection methods leave it to (deep) machine learning to detect travel modes from trajectory characteristics (Soares et al., 2019; Nikolic and Bierlaire, 2017). Explicit characteristics are usually as below:

- the speed of traveling, although speed alone is ambiguous in urban mobility;
- stop locations, e.g., at traffic lights or at bus stops;
- acceleration patterns, distinguishing, for example, a heavy vehicle from a light vehicle;
- noise patterns, e.g., of a private car, a bicycle, or a bus;
- space traveled through, e.g., road space, sidewalks, pedestrian zones, or light rail tracks;
- public transport schedules, and ideally real-time tracking data of public transport vehicles to compensate for deviations from the schedule;
- and land use at the start and end of the segment in order to estimate activities.

Mode detection algorithms are challenged by measurement uncertainties. Positioning uncertainty alone has an impact on (a) detecting stationary activities (while the observations are showing random movement), (b) first and second derivatives from positions  $(x, y, t)$ , i.e., speed and acceleration, which are more sensitive to positional uncertainty when the GNSS observation frequency is higher, and (c) the spatial separation between networks (e.g., where tracks go next to road lanes).

### 3.7.2 Map Matching

Closely related to mode detection is the challenge of map matching. Map matching addresses the uncertainty in the trajectory data not for estimating the travel

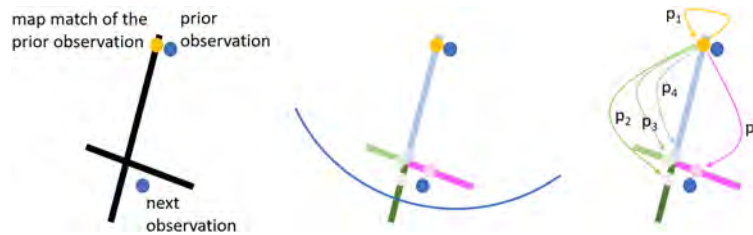


**Figure 3.3:** Map matching of the uncertain GNSS positions along a travel route often goes wrong when you are simply looking for the next road center line.

mode, but for estimating the most likely position of a tracked moving object on the mode's network. The conundrum is that map matching seems to require mode detection being solved first, in order to pick the most appropriate modal travel network. But mode detection itself is based on estimating the space (or network) traveled through already. This conundrum is usually circumvented in the literature by assuming that the travel mode is known. For example, a car navigation system's trajectory is a trajectory of a private vehicle, and thus a trajectory of a movement along a public road network. For these more trivial cases, a large number of map matching algorithms has been proposed, among them (Newson and Krumm, 2009; Quddus et al., 2009; White et al., 2000; Brakatsoulas et al., 2005). If the travel mode is known to change along the trajectory, only combined approaches lead to meaningful results. But if the travel mode is not known, both travel mode and map matching have to be estimated at the same time.

Map matching matches the measured positions, which are afflicted with uncertainty, with their most likely positions on the modal network, in order to infer the tracked object's actual path. In principle, if the object is a road vehicle, and the modal network the road network, one might want to match an observation  $(x, y)$  to the nearest road's or lane's center line. But just matching position to the nearest center line is prone to errors, as the sequence of the identified center lines may not lead to a realistic path (Figure 3.3). Hence, map matching requires methods that consider also the likelihood of the identified center lines within the logic of a travel journey.

One method containing this strategy is the hidden Markov model, HMM, a statistical model of a Markov process (Newson and Krumm, 2009). A Markov process is a sequence of events where the probability of each event depends only on the state attained in the previous event. For a trajectory in a map matching process, we can focus on the last "map matched" (visited) location and the next observed location to estimate the next visited location (Figure 3.4 left), which is sufficient to construct a realistic path for a trip segment. For the application of a Hidden Markov Model, first, all of the candidates for next locations are computed, which are all the nearest map matches for an observation within a reasonable range (Figure 3.4 center). Finally, the transition probabilities from the last visited location to all the possible next locations are computed, such that the sum of



**Figure 3.4:** The Hidden Markov Model applied to map matching.

all these probabilities equals 1. This transition probability is, in its simplest case, just a function of the difference between the observed distance (between the blue points in Figure 3.4) and the traveled path distance (between the yellow point in Figure 3.4 and the potential next locations). But this function can be made more complex in order to consider more contextual information in what is a reasonable path. For example, it can consider inertial sensor or compass observations in addition to location, the actual speed of the vehicle, and the road speed limits, as well as the travel patterns of the driver.

Map matching relies on reasonable sampling rates. When sampling rates are too low, many travel options exist prior to the next observed location, and the matching process becomes indeterminate. When sampling rates are too high, the distances measured become very different from the distance traveled, due to uncertainty in the measurement.

### 3.8 Conclusion

Location data is increasingly becoming available from sensors integrated in urban mobility. This chapter has introduced some tracking technologies and their properties, and then defined the notion of a trajectory with its critical properties of frequency (sampling rates) and accuracy (linked to map matching). Intelligent transportation systems rely on tracking data from everything that moves (Zhu et al., 2019; Guerrero-Ibáñez et al., 2018).

### Bibliography

- Aliedani, A. and Loke, S. W. (2019). Cooperative car parking using vehicle-to-vehicle communication: An agent-based analysis. *Computers, Environment and Urban Systems*, 77:Article 101256.
- Aliedani, A., Loke, S. W., Desai, A., and Desai, P. (2016). Investigating vehicle-to-vehicle communication for cooperative car parking: The copark approach. In *IEEE International Smart Cities Conference (ISC2)*, pages 1–8.

- Amoretti, M., Brambilla, G., Medioli, F., and Zanichelli, F. (2018). Blockchain-based proof of location. In *IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 146–153.
- Balzano, W. and Vitale, F. (2017). Dig-park: A smart parking availability searching method using v2v/v2i and dgp-class problem. In *31st International Conference on Advanced Information Networking and Applications Workshops*, pages 698–703.
- Banzhaf, H., Nienhüser, D., Knoop, S., and Zöllner, J. M. (2017). The future of parking: A survey on automated valet parking with an outlook on high density parking. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1827–1834. IEEE.
- Bennett, A. (2006). *Lagrangian Fluid Dynamics*. Cambridge University Press, Cambridge, MA.
- Bento, L. C., Parafita, R., and Nunes, U. (2012). Intelligent traffic management at intersections supported by v2v and v2i communications. In *5th International IEEE Conference on Intelligent Transportation Systems*, pages 1495–1502.
- Both, A., Duckham, M., Laube, P., Wark, T., and Yeoman, J. (2012). Decentralized monitoring of moving objects in a transportation network augmented with checkpoints. *The Computer Journal*, 56(12):1432–1449.
- Brakatsoulas, S., Pfooser, D., Salas, R., and Wenk, C. (2005). On map-matching vehicle tracking data. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 853–864. ACM Press.
- Bricka, S. and Bhat, C. (2006). Comparative analysis of global positioning system-based and travel survey-based data. *Transportation Research Record: Journal of the Transportation Research Board*, 1972:9–20.
- Carrion, C., Pereira, F., Ball, R., Zhao, F., Kim, Y., Nawarathne, K., Zheng, N., Zengras, P., and Ben-Akiva, M. (2014). Evaluating fms: A preliminary comparison with a traditional travel survey. In *Transportation Research Board 93rd Annual Meeting*. Transportation Research Board.
- Carter, P., Laurie, G. T., and Dixon-Woods, M. (2015). The social licence for research: why care.data ran into trouble. *Journal of Medical Ethics*, 41(5):404–409.
- Culnane, C., Rubinstein, B., and Teague, V. (2019). Two data points enough to spot you in open transport records. *Pursuit*.
- Das, R. D. and Winter, S. (2016). A context-sensitive conceptual framework for activity modelling. *Journal of Spatial Information Science*, 2016(12):45–85.



- Das, R. D. and Winter, S. (2018). A fuzzy logic based transport mode detection framework in urban environment. *Journal of Intelligent Transport Systems*, 22(6):478–489.
- Duckham, M. (2013). *Decentralized Spatial Computing*. Springer, Heidelberg.
- Geng, Y. and Cassandras, C. G. (2012). A new “smart parking” system infrastructure and implementation. *Procedia - Social and Behavioral Sciences*, 54:1278–1287.
- Guerrero-Ibáñez, J., Zeadally, S., and Contreras-Castillo, J. (2018). Sensor technologies for intelligent transportation systems. *Sensors*, 18(4):Paper 1212.
- Hausknecht, M., Au, T., Stone, P., Fajardo, D., and Waller, T. (2011). Dynamic lane reversal in traffic management. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1929–1934.
- Hirt, C. W., Amsden, A. A., and Cook, J. L. (1974). An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253.
- ITU (2014). Common requirements of the Internet of Things. Report ITU-T Y.2066, International Telecommunication Union, Geneva, Switzerland.
- Laube, P. (2014). *Computational movement analysis*. SpringerBriefs in Computer Science. Springer, Cham.
- Löper, C., Brunken, C., Thomaidis, G., Lapoehn, S., Fouopi, P. P., Mosebach, H., and Köster, F. (2013). Automated valet parking as part of an integrated travel assistance. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2341–2348. IEEE.
- Newson, P. and Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 336–343. ACM Press.
- Nikolic, M. and Bierlaire, M. (2017). Review of transportation mode detection approaches based on smartphone data. In *17th Swiss Transport Research Conference*.
- Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., and Yan, Z. (2013). Semantic trajectories modeling and analysis. *ACM Computing Surveys*, 45(4):Article 42.
- Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *British Machine Vision Association Conference*, pages 1–12. British Machine Vision Association.

- Quddus, M. A., Noland, R. B., and Ochieng, W. Y. (2009). The effects of navigation sensors and spatial road network data quality on the performance of map matching algorithms. *GeoInformatica*, 13(1):85–108.
- Rakha, H. and Kamalanathsharma, R. K. (2011). Eco-driving at signalized intersections using v2i communication. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 341–346.
- Roddis, S., Winter, S., Zhao, F., and Kutadinata, R. (2019). Respondent preferences in travel survey design: an initial comparison of narrative, structured and technology-based travel survey instruments. *Travel Behaviour and Society*, 16(7):1–12.
- Ruiz-Ruiz, A. J., Blunck, H., Prentow, T. S., Stisen, A., and Kjærgaard, M. B. (2014). Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 130–138. IEEE.
- Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.
- Soares, E. F. d. S., Campos, C. A. V., and Lucena, S. C. d. (2019). Online travel mode detection method using automated machine learning and feature engineering. *Future Generation Computer Systems*, 101:1201–1212.
- Stopher, P., FitzGerald, C., and Xu, M. (2007). Assessing the accuracy of the sydney household travel survey with gps. *Transportation*, 34(6):723–741.
- Truelove, M., Vasardani, M., and Winter, S. (2017). Testing the event witnessing status of micro-bloggers from evidence in their microblogs. *PLOS ONE*, 12(12):Paper e0189378.
- Veregin, H. (2005). *Data Quality Parameters*, pages 177–189. Wiley, Hoboken, NJ.
- Verma, R., Shrivastava, A., De, K., Mitra, B., Saha, S., Ganguly, N., Nandi, S., and Chakraborty, S. (2020). A smartphone-based passenger assistant for public bus commute in developing countries. *IEEE Transactions on Computational Social Systems*, 7(2):465–476.
- Wang, J., Duckham, M., and Worboys, M. (2016). A framework for models of movement in geographic space. *International Journal of Geographical Information Science*, 30(5):970–992.
- Wang, X., Wang, M., and Li, W. (2014). Scene-specific pedestrian detection for static video surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):361–374.

- White, C. E., Bernstein, D., and Kornhauser, A. (2000). Some map matching algorithms for personal navigation assistants. *Transportation Research C*, 8(1-6):91–108.
- Zhao, F., Pereira, F. C., Ball, R., Kim, Y., Han, Y., Zegras, C., and Ben-Akiva, M. (2015). Exploratory analysis of a smartphone-based travel survey in singapore. *Transportation Research Record: Journal of the Transportation Research Board*, 2494:45–56.
- Zheng, K., Zheng, Q., Chatzimisios, P., Xiang, W., and Zhou, Y. (2015). Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions. *IEEE Communications Surveys & Tutorials*, 17(4):2377–2396.
- Zhu, L., Yu, F. R., Wang, Y., Ning, B., and Tang, T. (2019). Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):383–398.

## 4 Navigation in Urban Environments

SALIL GOEL

### Abstract

This chapter provides an overview of technologies and methodologies for navigation in urban environments. It covers a range of technologies including wireless sensors, inertial and feature based that can be used either alone or within an integration. This chapter also discusses the brief principles of multi-sensor integration and outlines commonly used methodologies and approaches.

### Keywords

Navigation, positioning, statistical estimation techniques, sensor fusion, Inertial Measurement Unit

### 4.1 Introduction

Navigation is defined as the process of planning an object's position or trajectory using geometry, radio signals, etc. Therefore, navigation as a process, may involve estimating the object's position on earth, and/or guiding it through the course so that the object reaches the target destination. The importance and relevance of navigation and associated technologies has grown manifold in the last few years, so much so that not only almost all of the modern day cars are equipped with navigation systems, even the cheapest mobile phones in the market now offer navigation technologies at minuscule costs. The availability of affordable hardware and associated navigation technologies has made it possible for an average consumer to benefit from these technologies. This has been made possible primarily by the advent of Global Navigation Satellite Systems (GNSS). Today, these navigation systems are being used for day-to-day activities such as finding driving directions or vehicle tracking, as well as in advanced and complex applications such as driverless cars and robotic platforms.

The early navigators relied on ground landmarks and/or celestial observations for locating themselves and finding their way at sea. Another technology that was commonly used in the early navigation was *dead reckoning* (DR). The DR technique estimates the current position relative to the previous known position by keeping track of the distance traveled (or velocity measurement) and direc-

tion of movement. Inertial sensors, first conceived in the early 19<sup>th</sup> century were commonly used for DR. The earliest inertial sensors were mechanical in nature, but were later transformed to strapdown systems with advancements in micro-processor technology in the 20<sup>th</sup> century. As will be explained later, the position estimated using DR techniques diverges from the true position over time due to an accumulation of errors. It is due to the accumulation of errors that inertial sensors cannot provide a navigation solution on their own for an extended duration.

The development of radio technology paved the way for the development of terrestrial radio-navigation systems during the mid-20<sup>th</sup> century. LORAN (Long Range Navigation) and Omega were the first radio-navigation systems to be developed, with Omega being the first worldwide radio-navigation system to become operational in the 1970s. Although Omega was decommissioned in 1997, LORAN-C continued operations in the US until it was turned off in 2010. The Russian counterparts of LORAN and Omega called Chayka and Alpha RSDN-20, respectively, were also developed around the same time as the ones in the US. Although Chayka was operational at least until 2014, some reports suggest that RSDN-20 continues to be operational to this day.

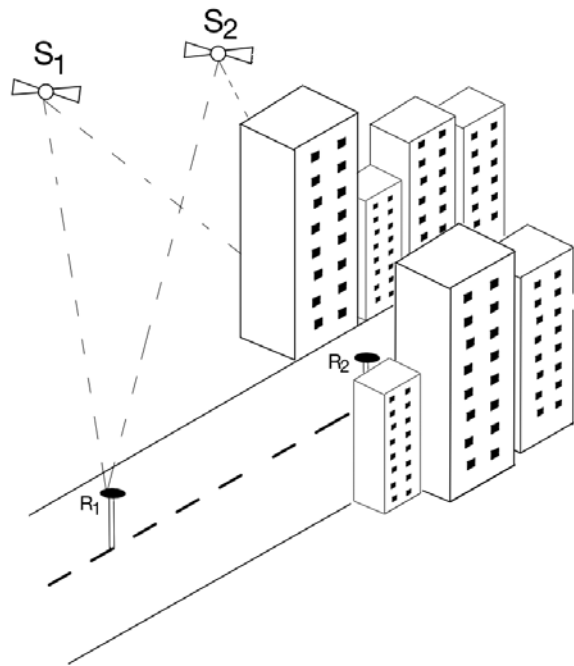
The interest in terrestrial radio-navigation systems saw a sharp decline with the arrival of satellite radio-navigation in the early 2000s. Recently, there has been a renewed interest in terrestrial radio-navigation systems fueled primarily by the vulnerabilities of satellite radio-navigation (i.e. GNSS), leading to the development of e-LORAN (Enhanced LORAN). The US and South Korea have already initiated efforts to deploy e-LORAN to complement the GNSS.

The modern day navigation systems are primarily powered by satellite based radio-navigation, collectively called GNSS. GNSS based navigation uses radio signals that are transmitted by the GNSS satellites, and received by the receivers (installed on cars, mobile phones, etc.) on the earth's surface. Using the information from these radio signals and satellite orbits, a receiver can estimate its position (and therefore, the position of the platform on which it is installed) almost in real-time. The US-based Global Positioning System (GPS) was the first GNSS system worldwide. The first GPS satellite was launched in 1978 and the system became operational in 1995. Following the success of the GPS and to reduce reliance on the US based GPS, other countries followed suit and started developing their own navigation systems. As of today, the GNSS satellite constellation includes the Global Positioning System (GPS) satellites by the United States, the GLONASS (Global Navigation Satellite System) by Russia, Galileo by the European Union (EU), Beidou by China, and other regional systems such as the QZSS (Quasi-Zenith Satellite System) by Japan, and the most recent one being the IRNSS (Indian Regional Navigation Satellite System) by India. Combined together, there are a total of about 132 GNSS satellites in operation as of date which includes 31 GPS (as on February 20, 2020), 23 GLONASS, 22 Galileo,

44 Beidou, 4 QZSS and 8 IRNSS satellites. A typical modern day navigation system (including recent mobile phones) can make use of all or some of these constellations to provide ubiquitous navigation solutions anywhere on the earth.

While GNSS remains the default and the most common navigation technology being used today on almost all platforms including aerial, terrestrial and marine, it is prone to various vulnerabilities such as spoofing and jamming. Even when there is no threat of spoofing or jamming, GNSS requires a clear line of sight between the receiver and satellites, and therefore, fails in indoor and other occluded environments such as dense urban regions or under a tree canopy. Furthermore, the signals received by a GNSS receiver in an urban environment are often corrupted by multipath, leading to significant reduction in navigational accuracy. Consider an example shown in Figure 4.1 where two GNSS receivers  $R_1$  and  $R_2$  are installed. The receiver  $R_1$  is in a relatively open environment, while  $R_2$  is installed in a typical urban environment consisting of urban canyons. The signals from two of the satellites,  $S_1$  and  $S_2$ , can reach  $R_1$  directly, while the signals from the same satellites to  $R_2$  are obstructed by buildings. Some of these signals may reach  $R_2$  but after undergoing multiple reflections from various surfaces, they cause multipath errors. The GNSS signals cannot penetrate buildings and, hence, a user at  $R_2$  may be rendered incapable of navigation using GNSS only. It is because of these major reasons that developing a robust navigation system remains one of the most important and challenging problems for urban mobility to this day.

To mitigate some of the limitations of GNSS, it is often integrated with complementary technologies and/or sensors, some of which include inertial sensors, vision sensors such as cameras, and even LiDAR (Light Detection and Ranging), to name a few. Modern day inertial sensors include a triaxial gyroscope, triaxial accelerometer, triaxial magnetometer, and other optional temperature and pressure sensors. The DR principle may be used to derive the navigation solution using inertial sensors, that provide the accelerations and rotation rates around the body axis. The vision sensors including cameras and LiDAR can provide information about the location of landmarks, which can then be used to estimate one's position and provide the navigation solution. The expectation from such an integration is that the complementary sensor/technology will provide the navigation solution in the partial (or extended) absence of the GNSS. At the heart of this integration of one or more sensors with the GNSS, lies an estimation framework that fuses the observations from multiple sensors including the GNSS, to yield a navigation solution. This estimation framework may utilize knowledge about the characteristics of the observations from each sensor, platform behavior, and the operating environment to yield the navigation solution. Kalman Filter (KF) (and its variants) have been the popular choice of the estimation framework since they were first proposed in the 1960s. The KF became popular after its application in trajectory estimation for the Apollo program and was ultimately incorporated



**Figure 4.1:** GNSS navigation in urban environments.

in the Apollo navigation computer. Even today, different variants of KF are being used in many commercial navigation systems. To overcome the assumptions of KFs (discussed in the later part of this chapter), many new filters and estimation frameworks have been proposed. While it is not possible to cover all estimation frameworks within this chapter, the overall philosophy and broad concepts of these frameworks are discussed. This chapter will also touch upon some recent and upcoming trends and navigation technologies (in Section 4.4) and discuss how these technologies are expected to help mitigate some of the major challenges of the navigation community. A summary of this chapter and conclusions are given in Section 4.5.

## 4.2 Navigation Technologies: An Overview and Comparison

Modern day navigation technologies can be classified into three broad categories. The first class of navigation technology makes use of proprioceptive sensor observations to perform navigation. Some examples of such types of sensors include odometers, accelerometers, gyroscopes, compass, magnetometers, barometers and more. Such systems rely on internal observations such as turning rate (gyroscope), velocity/acceleration (accelerometers), magnetic variation (compass/magnetometer), pressure variation (barometer), wheel rotation

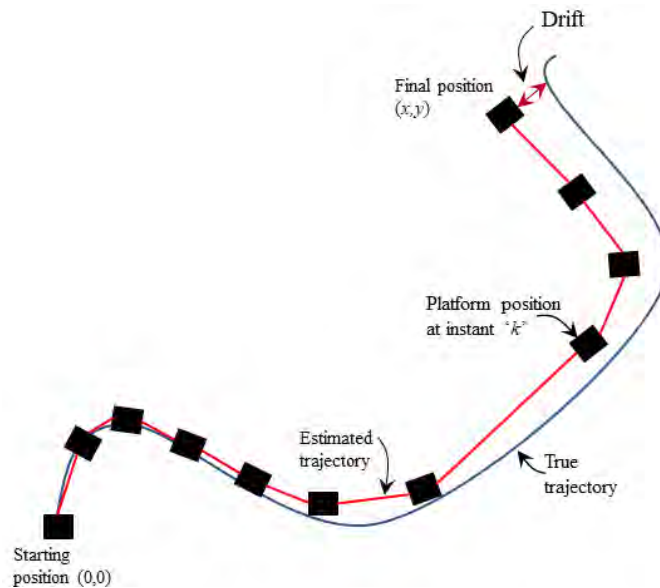
(odometer) etc. to perform navigation. Essentially, this form of navigation comes under the purview of DR. The second class of technology uses specially designed radio signals for navigation. This includes satellite based navigation (i.e. GNSS), terrestrial radio-navigation such as LORAN and modern terrestrial system Locata, and even signals that were not originally intended for navigation, including Wi-Fi, 3G/4G telecommunication signals, and even the upcoming 5G signals. The third class of navigation technology relies on observing and detecting distinct features in the operating environment (such as lines, edges, or corners) from multiple locations of the observer and then using these 'observations' to assist the user in navigation. These three classes of navigation technologies are discussed in the following sections and a qualitative comparison of the same is presented.

#### 4.2.1 Proprioceptive Sensor Observations

Proprioceptive sensors, by definition, record observations that are 'internal' to a system. A navigation system that relies on proprioceptive sensors is oblivious to the external features or environment around it and uses only the internal observations for navigation. These internal observations may include acceleration, turn-rate, wheel rotation rate etc. The observations are sent to an estimation framework that derives the navigation solution. Some of the commonly used sensors include accelerometers, gyroscopes, and odometers. An accelerometer measures the acceleration of a body, a gyroscope measures the rate of rotation, while an odometer measures the distance traveled by a wheeled vehicle. An integration of appropriate combination of these sensors, combined with a suitable estimation framework can yield the navigation solution of a moving vehicle, with respect to a local origin, from where the vehicle started moving. At each instant of time,  $k$ , the vehicle estimates the distance vector from the instant  $k - 1$  to  $k$  using the sensor observations. The resulting position at any instant,  $k$ , can be computed from the position at  $k - 1$  and the displacement vector. This is demonstrated in Figure 4.2, where the red line denotes the estimated displacement vector.

It is obvious from the demonstration in Figure 4.2 that the 'quality' of the estimated trajectory or vehicle position is dependent on the sampling rate of the sensors, as well as, the maneuvers undertaken by the vehicle. Using a sensor with a relatively low sampling rate on a highly maneuverable vehicle would lead to an inaccurate representation of the trajectory undertaken by the vehicle. An Inertial Measurement Unit (IMU) is one of the most commonly used sensors that integrates triaxial accelerometers and triaxial gyroscopes in a single unit. While an IMU yields only the raw sensor observations, AHRS (Attitude Heading Reference System) includes a filter, in addition to an IMU, that processes the raw observations to yield the platform position, orientation, or velocity. While in the early days inertial sensors were quite large and mechanical in nature, modern





**Figure 4.2:** Navigation using proprioceptive sensors.

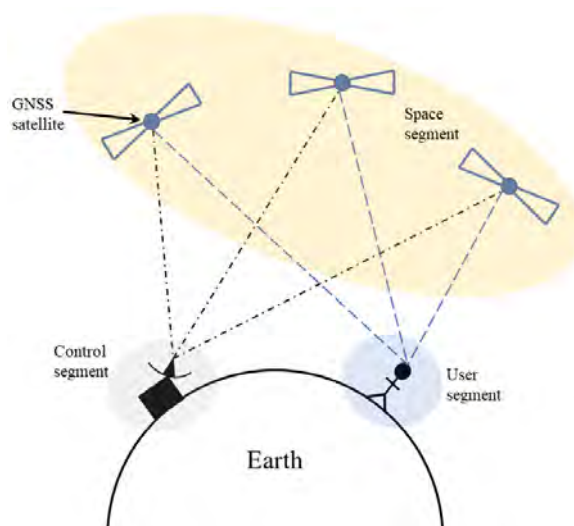
day sensors can be of the size of a few microns, such as the ones used in modern day smartphones and hand-held devices. In terms of cost and performance, modern day inertial sensors can range from costing a few dollars that may drift by hundreds of meters in a few minutes, to costing over a million dollars that drift less than 1–2 km in one day.

Given the initial vehicle position, accelerations and vehicle turn rate (in three directions), the displacement vector, and the final vehicle position can be estimated using Newtonian kinematic equations. This principle of navigation is commonly referred to as DR and was the earliest form of navigation adopted by sailors. Although the principle is relatively simple in theory, it suffers from various practical limitations. The major limitation is the corruption of sensor observations with errors, which get accumulated over time as they get integrated through the kinematic equations, causing the estimated vehicle position to drift from its 'true' position. It is primarily because of this reason that DR cannot be used for an extended period, and is often combined with complementary sensors that can help contain the drifts caused by the errors in inertial sensors.

#### 4.2.2 Using External Signals

The use of external signals for navigation dates back to the early 20<sup>th</sup> century when radio technology was just being adopted across the world. The early 'external signals' based navigation systems made use of terrestrial signals, such as LORAN and Omega. While very few terrestrial navigation systems remain

operational today, satellite based radio navigation has become quite popular recently. As shown in Figure 4.3, a satellite based navigation system comprises three major components: the Control segment, the Space segment, and the User segment.



**Figure 4.3:** Three broad segments of GNSS.

The Space segment comprises a satellite constellation which transmits radio signals to the users on the ground. The control segment is responsible for proper operation of the space segment and includes a network of monitor and control stations. These stations maintain the satellite orbits, track satellites, upload navigational data and maintain the satellite status. The user segment consists of the GNSS receivers that make use of the information received from the satellites to estimate their navigation solution. A vehicle/platform on Earth uses specially designed signals transmitted by four or more satellites, to compute its navigation solution using multilateration. In general, GNSS signals consist of a carrier, a ranging code, and navigation data. Accordingly, a user can either use the carrier phase or pseudorange observations to estimate the user's position. The navigation data provides the satellite ephemeris, clock bias parameters, satellite health status, and other information that is used in position estimation.

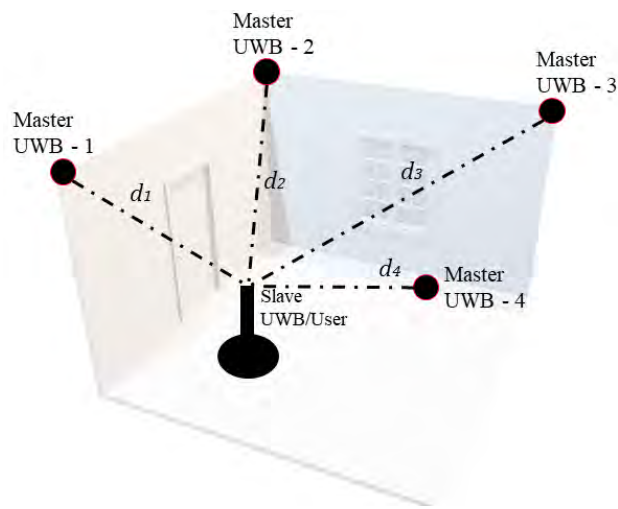
The first step in position estimation using GNSS is to derive the satellite position in an earth fixed coordinate system using the satellite ephemeris information. This is followed by a pseudorange or carrier phase model that makes use of either the pseudorange or the carrier phase (derived from either the ranging code or the carrier signal) and satellite position to estimate the user position. The quality of the GNSS solution is dependent on the receiver's antenna, choice of signals used (code based vs. carrier phase observations) and processing methodology adopted. In general, low-cost GNSS receivers can achieve accu-

racy in the order of  $\sim 5$  m, while high end GNSS receivers that use multiple frequencies and different corrections can achieve accuracies of the order of a few centimeters. Although the GNSS has become ubiquitous and navigation using this technology has become quite easy, it suffers from two major limitations. Firstly, GNSS is prone to spoofing and jamming and many instances have been reported across the world where the GNSS was intentionally or unintentionally jammed or spoofed. This limitation poses a risk to the user, in the sense that the user may be intentionally denied access to the GNSS by jamming the GNSS signals, or the user may be 'mis-directed' by spoofing the signals. The second limitation of this system is that GNSS signals require a direct line of sight between the satellite and receiver antennas. This condition cannot always be fulfilled due to obstruction caused by trees or buildings in urban environments. This limitation is so severe that it is becoming increasingly difficult to use GNSS in urban environments, due to the expansion of urban canyons, tunnels, underground/covered spaces (e.g. parking). Hence, GNSS alone may not be capable of meeting the navigational requirements of a majority of the users in a modern day environment. GNSS is therefore integrated with other complementary sensors or technologies, and the commonly used sensors for this purpose are inertial sensors.

GNSS is more suitable for long-term solutions, while inertial sensors can provide short-term solutions. An integration of these two technologies helps to overcome the limitations of each of them. For example, a vehicle passing through a tunnel may be denied GNSS due to unavailability of direct line of sight, whereas inertial sensors can provide a short-term navigation solution. When the vehicle emerges from the tunnel, GNSS signals can be re-acquired and the navigation solution can be maintained. For navigation in indoor and other GNSS-denied environments, various other types of signals are also being used these days, including but not limited to Wi-Fi, cellular signals, AM/FM signals, Ultra-Wide Band (UWB), and more. Signals such as Wi-Fi, cellular and AM/FM, collectively referred to as Signals of Opportunity (SoOP), were not originally intended for navigational purposes, but are now being applied in navigation. The common disadvantage of using SoOP is the low precision of the navigation solution (typically  $\sim 50$ – $100$  m) offered by them that limits their usability in demanding navigational applications.

UWB technology is gaining prominence as an alternative or complement to the GNSS for navigation in partially GNSS denied or indoor environments. UWB is a radio technology designed for short range and high bandwidth applications and operates in the 3.1 to 10.6 GHz frequency range. Use of UWB for localization or navigation requires a master and a slave combination. The master UWB node may be installed at a known location and the slave UWB node is carried by the vehicle/user. This master-slave UWB combination can be used to estimate the range between these nodes, using either Time of Arrival (ToA) or Return Trip

Time (RTT) observations. ToA methods measure the time of arrival at the slave node, while RTT methods measure the total time taken for the pulse to travel from the master to the slave and back to the master. Unlike RTT methods, ToA based methods require the master and receiver clocks to be perfectly synchronized and, therefore, RTT based methods are a popular choice for range estimation. As demonstrated in Figure 4.4, the user position (equipped with slave node) can be estimated once the range from the slave to three or more master nodes is known. Typical UWB sensors can cost  $\sim 50\text{--}100$  USD and offer an accuracy range of  $\sim 2$  cm, therefore, allow the user position to be estimated with an accuracy of better than  $\sim 10$  cm.



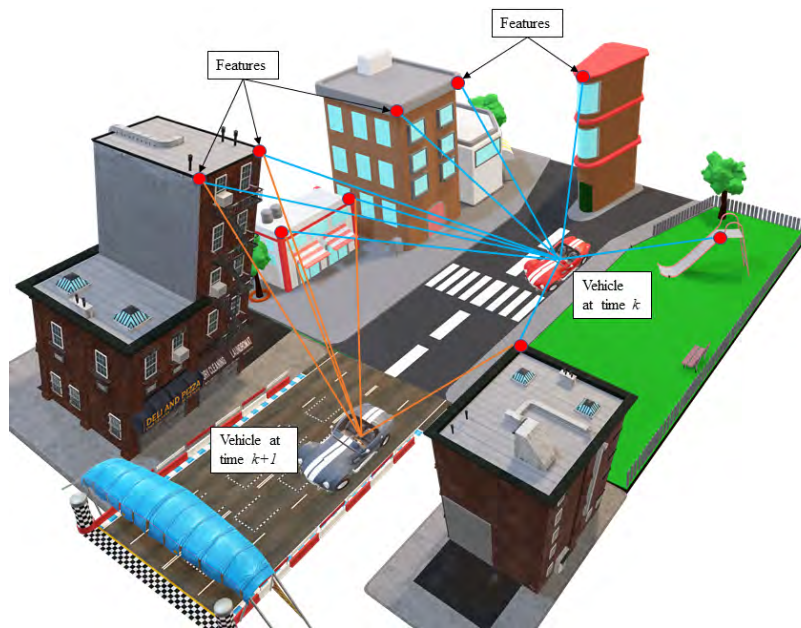
**Figure 4.4:** Navigation in GNSS-denied environment using UWB.

A UWB system requires all master UWBs to possess unique identifiers that identify it. The accuracy of the user is also dependent on the geometry of the network formed by these sensors, and hence, care must be taken to design 'optimal' networks that yield the best possible accuracy to the user. The use of UWBs has been successfully demonstrated in indoor environments, even in indoor/underground parking spaces where GNSS is absent. There have been efforts to integrate UWBs with GNSS to allow a seamless transition to a user when transiting from indoor to outdoor environments. Although UWB technology has proven to yield sufficiently high accuracies, it is limited by the short range of the sensors, investment and efforts required in setting up a sufficiently large infrastructure and poor penetration of UWB devices in the mass consumer market. Furthermore, the presence of a large number of master and/or slave nodes within the same environment is shown to cause network congestion, which causes significant drops in the communication range and hence severely affects the user's navigation. Despite these limitations, UWB technology can be used in limited indoor environments such as underground/indoor parking spaces, or the other specific

areas where stringent navigational accuracy requirements are required to be met in the absence of GNSS.

### 4.2.3 Using Environmental Features

A unique characteristic of the urban environments is the presence of distinct artificial features, using which a user can locate himself/herself. This principle of navigation using environmental features is quite similar to the navigation process that humans and other animals use on a day-to-day basis. It is, then obvious, that a navigation system that relies on environmental features needs at least one sensor that can capture information about the surrounding environment, extract *useful* features, and sequentially track those features to locate itself as it is navigating through a feature rich environment (for example, urban cities). Such a sensor could be a camera that can capture highly detailed semantic information or a LiDAR sensor that can record detailed geometric information. Figure 4.5 demonstrates an example of navigating using environmental features.



**Figure 4.5:** Urban navigation using environmental features.

Different types of distinct features are available in a typical urban environment, such as the ones demonstrated in figure 4.5. Some of the features that may be useful in navigation are marked in red. A vehicle senses the surrounding environment, extracts useful features, and tracks these features as it moves along. Given an initial starting point, the location of these features is first estimated in a user defined coordinate system. As the vehicle moves to the next location,

these features are tracked and used to estimate the new vehicle position. Additionally, new features that may be visible to the vehicle from the new position are also added to the estimation process. This process continues, and the location of the features and vehicle position may be estimated simultaneously. To ensure that the features are georeferenced, and the vehicle position is estimated in a global coordinate system (such as the World Geodetic System 1984), the initial vehicle position and orientation must be initialized with respect to this system. This approach of navigation using environmental features comes under the purview of simultaneous localization and mapping (SLAM) or odometry, depending on whether a map of the environment is simultaneously constructed or not. SLAM approaches attempt to construct a map of the environment, while simultaneously estimating the vehicle position. On the other hand, odometry using cameras (called Visual Odometry) or camera and inertial sensors (called Visual Inertial Odometry (VIO)), or even LiDAR odometry, focuses only on estimating the vehicle trajectory and does not consider simultaneously mapping the environment. Map matching is another technique that has been quite popular for localization in indoor/outdoor environments. Map matching methods require the user to have access to a precise map of the environment in advance. The user derived position is 'matched' to one of the features on the map, thereby assisting the user in improving its own position on the map.

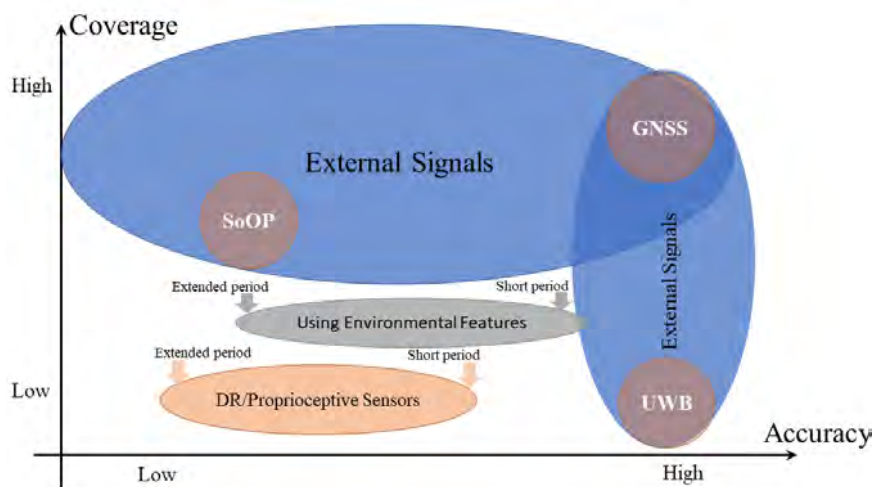
Navigation systems that make use of environmental features have found applications in mapping and/or navigating indoor environments and exploration missions. A key advantage of such approaches is that they do not require any prior infrastructure setup, except sensors on the vehicle platform. However, such an approach is successful only if there are sufficiently *distinct* features available in an environment. For example, SLAM or VIO may not work successfully in long hallways, that may be devoid of any distinct features. Similar to DR approaches, this method also suffers from drift due to accumulation of errors in the sensor observations. Hence, various techniques are deployed to constrain these drifts, which includes loop closures followed by an optimization (e.g. bundle adjustment) and/or inclusion of absolute positioning systems such as GNSS. Unlike the earlier navigation technologies, systems using features for navigation can be computationally expensive, due to the computational complexity involved in feature detection, feature tracking and optimization.

#### **4.2.4 Qualitative Comparative Analysis**

The last few years have witnessed a proliferation of different sensors that are being used in navigation technologies. This is primarily due to the increasing demand for navigation solutions and the increasing role of navigation (and mapping) technologies in possibly all sectors and aspects of life including but not limited to transportation, construction, mining, and exploration, urban management, disaster management, etc. As of today, a significantly large number of people

are using one or other navigation technologies. While most everyday users are satisfied with low-cost GNSS sensors (that may be installed in smartphones), various advanced applications require the use of complementary sensors to develop robust solutions. It is therefore, important for a user to understand the limitations of each of these technologies, and possibly, have an understanding of the performance offered by these technologies before they are integrated into a system.

To assess and compare the strengths and weaknesses of each of the navigation approaches, five different parameters have been chosen: accuracy, coverage, stability of the navigation solution, dependence on infrastructure, and computational complexity. Accuracy here refers to the closeness of the estimated solution to the 'true' value, while coverage refers to the area over which the navigation solution can be estimated. Stability of the solution refers to the ability of the navigation technology to maintain the estimated solution over a period of time. Dependence on infrastructure attempts to qualitatively assess the investment required for making the solution work and the computation complexity attempts to compare the computational infrastructure needed for achieving the desired solution. A broad comparison of various navigation technologies in terms of coverage and accuracy is presented in Figure 4.6.



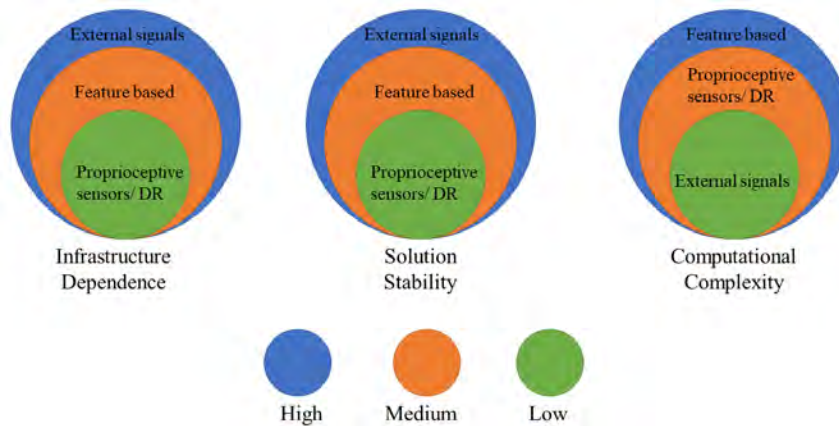
**Figure 4.6:** Accuracy versus coverage: Comparison of navigation solutions.

The technologies that make use of external signals for navigation cover the broadest spectrum on the coverage-accuracy plot. GNSS is capable of providing global coverage to the extent that it can be used ubiquitously, while also providing high navigational accuracies. This makes them a good candidate for use in a wide variety of applications. On the other hand, UWB technology can provide high localization accuracy comparable with the GNSS but is limited to local areas only. Other signals such as 4G/5G, AM/FM etc. that fall under the broad category



of SoOP provide a broader coverage as compared to UWB, but have significantly poorer accuracy. Terrestrial signals, such as LORAN, provide higher coverage and better accuracy compared to the SoOP, but also require higher investment in terms of infrastructure and maintenance. In contrast, technologies based on proprioceptive sensors and use of environmental features provide lower coverage, and accuracies ranging from low to high, depending on the optimization techniques used and time period over which the solution is used. Over an extended period of usage, the accuracy of both feature based and proprioceptive sensor based technologies tends to degrade due to accumulation of errors, causing the solution to drift.

As can be seen in Figure 4.7, a significant investment is needed in terms of infrastructure development and maintenance to be able to use external signals (such as GNSS or UWB) for navigation. In contrast, proprioceptive sensor based solutions require the least amount of infrastructure, but they also offer poor solution stability over extended periods of time. Extraction and tracking of useful features from the surrounding environment can be quite computationally expensive and therefore, a significant investment in terms of computational resources is needed for feature based navigation. Another implication of the higher computational complexity is that the navigation solution may not be available to a user in real-time, which may be a critical requirement in certain applications. Although each of the navigation solutions have their own advantages and limitations, many of the limitations, at least in terms of accuracy, coverage and solution stability can be overcome by a suitable integration of complementary technologies at cost of increased complexity and higher financial investment.

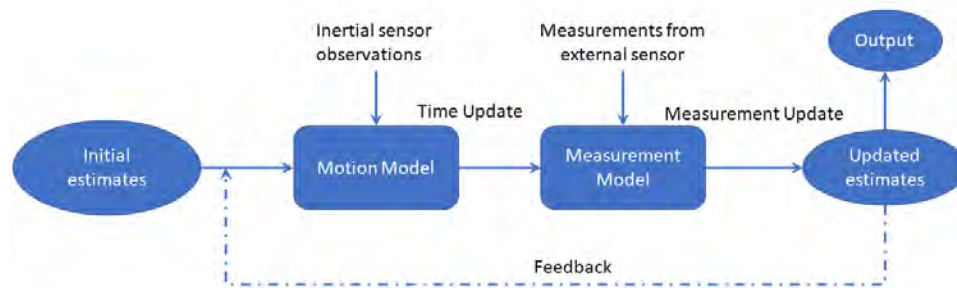


**Figure 4.7:** Qualitative comparison of navigation systems.



### 4.3 Sensor Fusion for Navigation

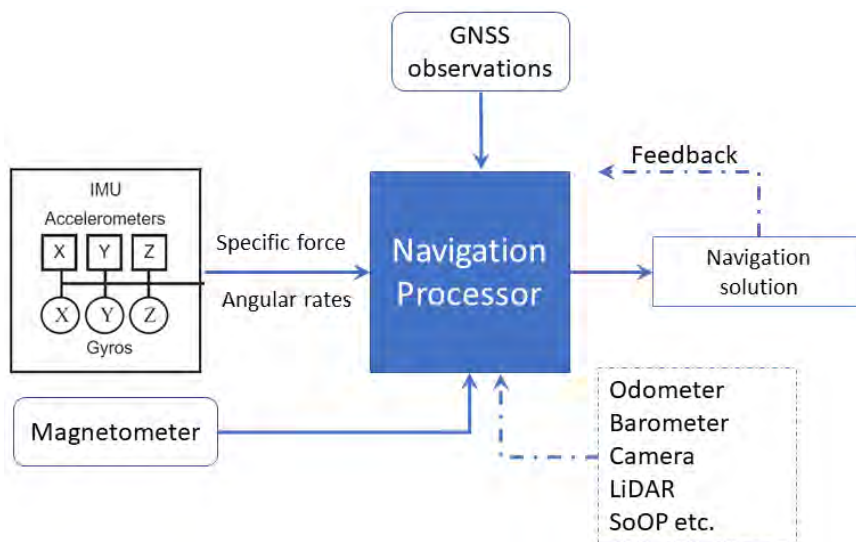
At the heart of a navigation system is a navigation processor (e.g. a filter) that takes the raw observations from one or more sensors as input and provides an estimate of the vehicle state, including position and velocity. Over the years that many different types of processing filters and architectures have been proposed, each with their own advantages and limitations. The Kalman Filter, originally proposed in the 1960s, has enjoyed quite a significant amount of popularity and remains one of the popular choices for estimation even today. The KF employs a predictor-corrector architecture to sequentially process the sensor observations and generate the state estimates of a moving platform (e.g., car, pedestrian). The predictor component of the KF generally uses a kinematic motion model to predict the state vector, given an initial state estimate and inertial sensor observations. This motion model should represent the platform characteristics and maneuvering capabilities. The predicted state estimate is then passed to the corrector component of the KF that makes use of the predicted state, sensor observations (e.g., GNSS, LiDAR, camera), and a measurement model to update the predicted state and generate the corrected state estimate, along with the sensor biases. The employed measurement model represents the relationship between the platform state that needs to be estimated, and the sensor observations. The sensor biases, thus estimated, are generally fed back to the KF to correct the sensor observations, and the whole process repeats itself. A simplified graphical representation of this method is represented in Figure 4.8.



**Figure 4.8:** A generic predictor-corrector framework employed in navigation processors.

The traditional KF made various simplifying assumptions such as linear motion and measurement models, uncorrelated measurement and process noise, Gaussian nature of the noise etc. Therefore, various filters have been developed over the years, that employ a similar predictor-corrector framework but attempt to overcome the limitations of the conventional KF. Extended Kalman Filter (EKF) can be suitably employed when the non-linearity in motion and/or measurement models is not very high. Unscented Kalman Filter (UKF) performs better than EKF and KF in case of highly non-linear motion and/or measurement models,

while assuming the noise to be Gaussian in nature. Particle filters are one of the most generic forms of KF that do not make any assumption about noise distribution and do not assume the models to be linear. There are multiple other variants of each of these filters that tackle other assumptions such as colored noise or correlated errors. The architecture of the employed filter is dependent on the types of sensors integrated in a multi-sensor platform, and therefore, multiple variants of these filters exist. For example, EKF is generally suitable for a GNSS/IMU integration, while SLAM or odometry based filters are needed for integration of camera and/or LiDAR sensors. These days graph based methods (e.g., Belief Propagation) are gaining popularity for vehicle navigation and/or tracking applications. Nevertheless, the basic architecture of a multi-sensor platform for navigation is depicted in Figure 4.9.



**Figure 4.9:** Generic multi-sensor fusion architecture for navigation.

An important area of research in developing such multi-sensor platforms is developing a robust navigation processor that is capable of handling sensor observations from multiple sensors such as GNSS, IMU, Odometer, Camera, or LiDAR. The complexity of the processor is dependent on the chosen set of sensors, the environment where the platform is expected to operate, deliverables of the processor (trajectory and/or map, etc.) and other application requirements such as expected accuracy, real-time versus post-processed solution, etc. Therefore, designing a robust navigation processor is a non-trivial and complex process that requires an ingenious combination of science and art.

## 4.4 Upcoming Trends in Localization and Navigation

The growing realization of the benefits of the navigation technologies has allowed them to be used in a wide variety of applications, to the extent that many modern day consumer electronics and some daily use items are now equipped with one or the other navigation technology. For example, GNSS has become synonymous with smartphones and smartwatches, and cars are now equipped with GNSS sensors. A whole new industry based on location based services has sprung up that is providing new solutions. This chapter identifies three broad upcoming trends in the area of localization of navigation. Firstly, there has been a rise in the types of signals that can now be used for navigation. For example, the newest 5G signals can provide a much better localization solution compared to their predecessors. Smartphones can now receive and process carrier phase observations, making them much more powerful and precise. The newer Wi-Fi standards are being designed to provide improved range estimates between the router and the Wi-Fi device (e.g., smartphone) using RTT methods, to enable improved navigation of these devices in indoor environments. Technologies such as UWB, Zigbee, or Dedicated Short Range Communication (DSRC) are gaining popularity for their navigational assistance capabilities. Secondly, there has been a tremendous development in designing better, robust and efficient navigation processors capable of integrating various complementary sensors. Deep learning architectures are now being investigated for their potential in providing navigation estimates and have shown significant promise. Efficient graphical SLAM approaches have been in the works for quite some time and are becoming mature. Thirdly, due to the proliferation of users relying on navigation technologies and availability of a wealth of signals, there is an increasing emphasis on developing *cooperative* solutions wherein, different users assist the neighboring users in localization and navigation. Since, some of the users may be better placed in terms of navigational capabilities, they may assist their neighbors (e.g., other cars in the vicinity of a car) who may not be so fortunate or capable (in terms of available navigational accuracy) by sharing their own information and some knowledge about their neighbors. This also opens up the doors for cooperation among different types of platforms, for example, Unmanned Aerial Vehicles (UAVs) assisting ground vehicles in navigating complex terrain.

## 4.5 Summary and Conclusions

Navigation has evolved from the days of relying on landmarks/celestial observations or DR to using multiple ubiquitous signals and integrated multi-sensor systems utilizing a wide variety of sensors. This chapter provided a brief overview of the navigation technologies currently available and being used across the world, and discussed the challenges and limitations of each of these technologies. Fur-

ther, the chapter briefly explained the broad principles involved in integration of complementary sensors to develop robust navigation systems. While existing technologies are quite capable of providing navigation solutions in complex urban environments, challenges still exist, primarily in developing efficient navigation processors that can make the best possible use of complementary sensors. Therefore, developing novel filters and estimation frameworks has been an important research area for quite some time and will continue to be so in the near future. The sensors will improve over time, and more signals may become available in the future, but these signals and sensors can be best utilized only when efficient and robust navigation processors are available.

## 4.6 Further Reading

While this chapter has provided a broad overview of a field, I provide also a suggested reading list for those who want to go deeper and explore in more detail (Abbas et al., 2019; Atia and Waslander, 2019; Chang et al., 2019; Chao et al., 2020; Feng et al., 2020; Gabela et al., 2019; Gao et al., 2016; Goel et al., 2017; Guo et al., 2019; Hashemi and Karimi, 2014; Li et al., 2019; Maaref and Kassas, 2020; Masiero et al., 2020; Mohamed et al., 2019; Retscher et al., 2020; Williams, 1992; Zafari et al., 2019).

## Bibliography

- Abbas, M., Elhamshary, M., Rizk, H., Torki, M., and Youssef, M. (2019). WiDeep: WiFi-based accurate and robust indoor localization system using deep learning. In *2019 IEEE International Conference on Pervasive Computing and Communications, PerCom 2019*.
- Atia, M. M. and Waslander, S. L. (2019). Map-aided adaptive GNSS/IMU sensor fusion scheme for robust urban navigation. *Measurement: Journal of the International Measurement Confederation*.
- Chang, L., Niu, X., Liu, T., Tang, J., and Qian, C. (2019). GNSS/INS/LiDAR-SLAM integrated navigation system based on graph optimization. *Remote Sensing*.
- Chao, P., Xu, Y., Hua, W., and Zhou, X. (2020). A survey on map-matching algorithms. In Borovica-Gajic, R., Qi, J., and Wang, W., editors, *Databases Theory and Applications*, Lecture Notes in Computer Science, pages 121–133, Cham. Springer International Publishing.
- Feng, X., Zhang, T., Lin, T., Tang, H., and Niu, X. (2020). Implementation and performance of a deeply-coupled GNSS receiver with low-cost MEMS inertial sensors for vehicle urban navigation. *Sensors (Switzerland)*.

- Gabela, J., Retscher, G., Goel, S., Perakis, H., Masiero, A., Toth, C., Gikas, V., Kealy, A., Koppányi, Z., Błaszczak-Bąk, W., Li, Y., and Grejner-Brzezinska, D. (2019). Experimental evaluation of a UWB-based cooperative positioning system for pedestrians in GNSS-denied environment. *Sensors (Switzerland)*.
- Gao, N., Wang, M., and Zhao, L. (2016). An integrated INS/GNSS urban navigation system based on fuzzy adaptive Kalman filter. In *Chinese Control Conference, CCC*.
- Goel, S., Kealy, A., Gikas, V., Retscher, G., Toth, C., Brzezinska, D. G., and Lohani, B. (2017). Cooperative localization of unmanned aerial vehicles using GNSS, MEMS inertial, and UWB sensors. *Journal of Surveying Engineering*.
- Guo, G., Chen, R., Ye, F., Peng, X., Liu, Z., and Pan, Y. (2019). Indoor Smartphone Localization: A Hybrid WiFi RTT-RSS Ranging Approach. *IEEE Access*, 7:176767–176781.
- Hashemi, M. and Karimi, H. A. (2014). A critical review of real-time map-matching algorithms: Current issues and future directions. *Computers, Environment and Urban Systems*, 48:153 – 165.
- Li, W., Xiong, Z., Sun, Y., and Xiong, J. (2019). Cooperative positioning algorithm of swarm UAVs based on posterior linearization belief propagation. In *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference*.
- Maaref, M. and Kassas, Z. M. (2020). Ground vehicle navigation in GNSS-challenged environments using signals of opportunity and a closed-loop map-matching approach. *IEEE Transactions on Intelligent Transportation Systems*.
- Masiero, A., Perakis, H., Gabela, J., Toth, C., Gikas, V., Retscher, G., Goel, S., Kealy, A., Koppányi, Z., Błaszczak-Bak, W., Li, Y., and Grejner-Brzezinska, D. (2020). Indoor navigation and mapping: Performance analysis of UWB-based platform positioning. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*.
- Mohamed, S. A., Haghbayan, M. H., Westerlund, T., Heikkonen, J., Tenhunen, H., and Plosila, J. (2019). A Survey on Odometry for Autonomous Navigation Systems. *IEEE Access*.
- Retscher, G., Kealy, A., Gabela, J., Li, Y., Goel, S., Toth, C. K., Masiero, A., Błaszczak-Bąk, W., Gikas, V., Perakis, H., Koppányi, Z., and Grejner-Brzezinska, D. (2020). A benchmarking measurement campaign in GNSS-denied/challenged indoor/outdoor and transitional environments. *Journal of Applied Geodesy*.
- Williams, J. E. D. (1992). *From sails to satellites: The origin and development of navigational science*. Oxford University Press, Oxford.

Zafari, F., Gkelias, A., and Leung, K. K. (2019). A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2599.

## 5 Computer Vision Techniques for Urban Mobility

KOUROSH KHOSHELHAM

### Abstract

This chapter provides an overview of computer vision techniques with applications in urban mobility and transport systems. Focusing on imagery and Light Detection and Ranging (LiDAR) point clouds as the main data modalities, the chapter reviews relevant computer vision tasks, including classification, segmentation, object detection and tracking. Example applications of these techniques to data captured by stationary sensors installed in the environment as well as mobile sensors onboard vehicles will then be discussed.

### Keywords

Detection, tracking, classification, segmentation, localization, pose estimation

### 5.1 Introduction

The increasing prevalence of surveillance cameras in urban environments in recent years has provided an opportunity to develop new solutions to overcome challenges in urban mobility and transport systems. Cameras mounted on vehicles also offer the potential to sense the road environment and develop self driving capabilities which make urban mobility safer and more efficient. In addition to cameras, LiDAR sensors are becoming a preferred sensor for spatial perception in autonomous vehicles. These opportunities have led to a surge in the development of computer vision methods for automated interpretation of imagery and LiDAR point clouds with the ultimate aim of improving urban mobility.

This chapter reviews some promising applications of computer vision techniques for improving urban mobility. The focus will be on imagery and LiDAR point clouds as the more common data modalities for computer vision algorithms. Further, this chapter will focus on individual mobility, i.e. pedestrians, vehicles and cyclists. Other modes of mobility, such as freight, air, and maritime mobility, have received less attention from the computer vision research community, and are excluded from the present discussion.

While this chapter reviews example applications of computer vision techniques to urban mobility, it is not meant to serve as a classic review of the state of the art and is by no means exhaustive and comprehensive. Instead, the chapter aims to identify potential application areas where computer vision techniques can provide novel solutions to problems in urban mobility and transport systems. In the following, we first discuss common computer vision tasks for mobility applications, and then review promising examples of computer vision techniques applied to imagery and LiDAR point clouds captured by stationary sensors installed in the environment as well mobile sensors on board vehicles.

## 5.2 Common Computer Vision Tasks for Mobility Applications

Computer vision includes a wide range of algorithms developed to carry out specific tasks with the common goal of enabling a computer to understand the world by analyzing sensor observations in the form of images and point clouds. Common computer vision tasks for mobility applications include classification, segmentation, object detection, and tracking.

### 5.2.1 Classification

Classification is the task of assigning one or more category labels that identify the type of object or objects present in the data. The common approach to the classification of images and point clouds is supervised machine learning, where a mapping between the input data and the output category label is learned from a set of training examples. The category labels can be deterministic (hard labels) or probabilistic scores (soft labels).

Research on image classification made significant progress after the introduction of the ImageNet Challenge (Russakovsky et al., 2015) in 2010. The success of AlexNet (Krizhevsky et al., 2012) in the ImageNet 2012 Challenge led to the popularity of deep convolutional neural networks (CNNs) for image classification. Since then, many different CNN architectures have been proposed, such as VGG (Simonyan and Zisserman, 2015), GoogLeNet (Inception-v1) (Szegedy et al., 2015), and ResNet (He et al., 2016), which have achieved outstanding results on the ImageNet dataset. The classification of point clouds has achieved less success compared to image classification. State-of-the-art approaches to point cloud classification are either point-based methods, such as PointNet (Qi et al., 2017), or voxel-based methods, such as VoxNet (Maturana and Scherer, 2015).



### 5.2.2 Segmentation

Segmentation is the task of partitioning the data into segments that represent objects or parts thereof. A typical segmentation algorithm generates an output the same size as the input data, where each pixel or point is assigned a segment ID. If additionally, a category label is also assigned to each pixel or point, then the process is called semantic segmentation. The task of semantic segmentation is therefore a combination of segmentation and classification tasks.

Similar to classification, state-of-the-art segmentation and semantic segmentation methods are based on deep neural networks (Liu et al., 2019; Guo et al., 2020). The majority of these methods are based on supervised machine learning, where a deep network is trained using manually annotated images or point clouds available from public datasets.

### 5.2.3 Object Detection

Object detection is the task of localizing one or more objects of a certain category in the data. As such, object detection is a combination of classification and localization tasks. The localization is typically done by computing a bounding box around the object or a mask representing the object boundaries.

State-of-the-art approaches to object detection in imagery and point clouds are either based on region proposals or based on single shot classification and bounding box regression. Region proposal-based methods for object detection in images include Faster-RCNN (Ren et al., 2015) and Mask RCNN (He et al., 2017), and single shot methods include SSD (Liu et al., 2016) and the different versions of YOLO (Redmon et al., 2016). Methods for object detection in point clouds include PointRCNN (Shi et al., 2019), which is based on region proposals, and 3DSSD (Yang et al., 2020), which is a single shot method. These methods have been used for detecting vehicles, cyclists and pedestrians in images and LiDAR point clouds.

### 5.2.4 Tracking

In computer vision, tracking is the task of localizing an object in a sequence of data. Object tracking in a sequence of images or LiDAR scans usually involves detecting the object in the first image frame or LiDAR scan, and estimating its location in the subsequent frames or scans. The output of a tracking algorithm is the trajectory of the object in the sensor coordinate frame, which can be easily transformed to a trajectory on the ground by georeferencing the camera or the LiDAR sensor.

Recent methods for object tracking in images based on a convolutional Siamese network (Bertinetto et al., 2016; Wang et al., 2019) have achieved promising re-

sults in tracking people and vehicles. The Siamese network has also been extended for 3D tracking of pedestrians and cyclists in LiDAR data (Zarzar et al., 2019).

### 5.3 Computer Vision with Stationary Sensors

Recent advances in computer vision together with the prevalence of surveillance cameras installed in outdoor and indoor urban environments have made it possible to develop smart solutions for problems in mobility and urban transport. In the following pages, we review a few promising examples of such solutions made possible by computer vision methods. Most of the methods discussed in this section are based on imagery, as the use of stationary LiDAR sensors for monitoring urban environments is not currently common.

#### 5.3.1 Pedestrian Detection and Tracking

Pedestrian detection and tracking using surveillance cameras and LiDAR sensors has been used in various urban mobility applications including pedestrian traffic management, prevention of overcrowding, origin-destination estimation, and monitoring intersections and pedestrian crossings. A practical application of pedestrian tracking in a video footage was shown by Kong et al. (2007) where the authors demonstrated that the tracking results can be used to proactively respond to incidents in a railway station. Another practical application of image based pedestrian tracking in indoor environments was demonstrated by Georgoudas et al. (2010) who developed an evacuation guidance system based on pedestrian tracking to prevent congestion during evacuations. For outdoor environments, image-based pedestrian tracking has been used to monitor intersections and provide useful information to improve the design of pedestrian crossings and adjust the signal timing (Malinovskiy et al., 2008).

A limitation of surveillance cameras for pedestrian tracking is their susceptibility to low light conditions especially in emergency situations in indoor environments. Li et al. (2019b) demonstrated the poor performance of color images for pedestrian origin-destination estimation during an emergency in a dark indoor environment, and proposed a deep convolutional network to fuse color, infrared and depth images for origin-destination estimation in emergency scenarios.

While pedestrian tracking using a single camera has been successfully applied in simple and small indoor and outdoor environments (Acharya et al., 2017), for large and more complex environments a multi-camera approach is preferred. Multi-camera pedestrian tracking includes the additional challenge of identity association across different camera views. Wu et al. (2020) formulated the 'identify' association as a graph-cut problem and showed an application of multi-camera

pedestrian tracking for analyzing the shopping behavior of customers in an indoor market hall.

Pedestrian detection and tracking in LiDAR data has also received a great deal of attention in recent years. Compared to cameras, LiDAR sensors are independent of ambient light and are less susceptible to poor lighting and adverse weather conditions. Zhao et al. (2018) demonstrated the application of pedestrian tracking using a roadside LiDAR sensor to infer the crossing intention of pedestrians.

Current methods for pedestrian detection and tracking in images and LiDAR data are successful in less crowded scenes where individual pedestrians are clearly visible. For crowded scenes, where extracting the complete trajectories of individual pedestrians may not be feasible, extracting global parameters such as crowd density, global velocity (Yi et al., 2015), and congestion is more convenient.

### 5.3.2 Crowd Congestion Classification

Crowd congestion information automatically extracted from surveillance images in real time provides valuable insights for the management of busy transport hubs especially during peak commute times. Crowd congestion is usually measured as the average occupancy area available per person, commonly referred to as level of service. As such, it can be estimated by detecting and counting the pedestrians in surveillance images and computing the crowd density (Ryan et al., 2015). However, in crowded scenes, where pedestrians are partly occluded in the images, counting, and density estimation will be inaccurate and may lead to incorrect congestion classification results.

An alternative approach is to directly classify local image regions into different congestion classes based on crowd appearance features. Li et al. (2019a) trained a long short term memory (LSTM) network using manually labelled images of different crowd densities to classify image patches corresponding to a grid on the ground and generate a level of service map of a railway platform (Figure 5.1). The resulting map overlaid on a 3D model of the platform provides an effective visualization of both spatial and temporal variations of congestion classes in real time. To avoid the influence of occlusion in a single view, Li et al. (2020) extended this approach to multiple views by classifying image patches corresponding to a grid in each camera view and combining the classification results using an ensemble combination rule. This multi-view approach was shown to produce a more accurate level of service map than that obtained from each individual view.



**Figure 5.1:** Direct generation of crowd congestion map from surveillance images.

### 5.3.3 Parking Occupancy Detection

Parking occupancy detection using surveillance cameras provides a low-cost yet accurate and reliable solution for smart parking systems in crowded cities. The common approach is to train a binary classifier to classify image regions corresponding to parking spaces as either occupied or vacant. Early works such as True (2007) used hand crafted features based on the appearance of vehicles and achieved modest accuracies. But recent advances in feature learning using deep convolutional networks made it possible to achieve much higher accuracies in vehicle detection and determining the occupancy of parking spaces. Valipour et al. (2016) trained a deep VGG network using labelled images from PKLot dataset (De Almeida et al., 2015) and reported an occupancy detection accuracy of 99% on a test set from the same dataset. Acharya et al. (2018) investigated the feasibility of transfer learning, where a deep network trained on a public dataset such as PKLot is applied to images captured in a different parking setting. They tested this approach using an SVM classifier plugged into a VGG network and reported an accuracy of 97% for detecting the occupancy of parking spaces. Chapter 11 provides a tutorial on the transfer learning approach to image-based parking occupancy detection using a ResNet architecture.

### 5.3.4 Detection of Anomalous Driving Behaviors

An interesting application of computer vision techniques in urban mobility is automated detection of anomalous driving behaviors, such as swerving, speeding, and crossing solid lines, in surveillance images. While methods for detecting different anomalous behaviors may be different, a common ingredient of these methods is vehicle detection, tracking and reconstruction of vehicle trajectories. An early work on image-based analysis of driving behaviors is the work of Song et al. (2014), who used simple background elimination and feature point extraction to detect and track vehicles in video footages of several roads in Xian,

Shanghai, and Fuzhou. They used the reconstructed vehicles trajectories to estimate the speed and identify various anomalous behaviors such as lane changing, sudden stopping, and sudden slowing down. Zheng et al. (2019) proposed a taxonomy of anomalous driving behaviors and developed a vehicle detection and tracking system based on Mask RCNN (He et al., 2017) to detect speed anomalies, solid line crossing, and vehicles entering restricted zones such as a bus lane. They also proposed a web mapping application to visualize anomalous driving behaviors on different roads as a guide for vulnerable road users such as cyclists and pedestrians.

## **5.4 Computer Vision with Mobile Sensors**

The widespread interest in autonomous vehicles in recent years has resulted in the development of computer vision techniques for spatial perception of road environments using cameras and LiDAR sensors on board vehicles. This section reviews a few examples of promising applications of computer vision techniques applied to imagery and point clouds captured by vehicle-borne cameras and LiDAR sensors.

### **5.4.1 Driving Scene Perception**

Automated perception and understanding of the driving scene is a critical capability for the successful operation of fully autonomous vehicles. A first computer vision task for autonomous vehicles is to detect the road boundaries and lane markings. Many modern vehicles already have the lane detection and lane keeping capability on well marked roads. The challenge, however, is the detection of road and lane boundaries on unmarked and weakly marked roads. The KITTI Road Detection Benchmark (Fritsch et al., 2013) provides an evaluation and comparison of road detection methods based on images and LiDAR data on several challenging datasets. When road markings are not clearly visible in the data, the fusion of images and LiDAR point clouds can provide more reliable detection results. Chen et al. (2019) train a convolutional network to learn and fuse image and LiDAR features to detect road boundaries, and achieve state of the art performance on KITTI road detection dataset. Prior knowledge and existing maps can also be used to support road and lane detection in sensor data. Wang et al. (2020) take advantage of road information from OpenStreetMaps and combine it with image features in a search-based optimization algorithm to estimate the correct location of lane boundaries.

Another important computer vision task for autonomous vehicles is the recognition of traffic signs. State of the art deep learning methods for image classification generally achieve high accuracies in traffic sign recognition in images. The German traffic sign recognition benchmark (Stallkamp et al., 2012) demon-

strated that deep convolutional networks can achieve correct classification rates up to 99.46 % on test images of various traffic signs.

Detection of vehicles, pedestrians and cyclists in the road environment is another important computer vision task for autonomous vehicles. It is a particularly challenging task due to the dynamic nature of objects which can result in occlusion and obscure images. The KITTI Vision Benchmark Suite (Geiger et al., 2012) provides a dataset comprising imagery and LiDAR data of vehicles, pedestrians, and cyclists at three levels of occlusion: fully visible (easy), partly occluded (moderate), and difficult to see (hard). The results of the benchmark show that current methods are generally better at 2D detection than 3D detection. For example, the current top performing method for 2D detection of pedestrians achieves an average precision of 90.50 %, 83.06 %, and 78.35 % on easy, moderate, and hard test samples respectively, whereas the best average precision for 3D pedestrian detection is only 53.10 %, 45.37 %, and 41.47 % for easy, moderate, and hard test samples. Also, the detection of vehicles seems to be an easier task, while the detection of pedestrians and cyclists is a greater challenge. For example, the current best average precision for 3D car detection in KITTI Benchmark is 82.33 % on moderate test samples, whereas for 3D detection of cyclists and pedestrians the best average precision on moderate samples drops to 71.86 % and 45.35 %, respectively.

Other computer vision tasks related to autonomous driving include the detection of road incidents and road surface conditions. Levering et al. (2020) proposed a taxonomy of unsigned road incidents and developed a deep learning model to recognize eight types of road incidents in driver view images, namely vehicle crash, tree-fall, fire, landslide, collapse, flood, snow, and animal on road. Pena-Caballero et al. (2020) proposed a system to detect potentially hazardous road surface conditions such as potholes and cracks using driver view images. These methods can be used in a crowd-sourcing approach to collect information about road conditions and use centralized or decentralized communication systems to disseminate the information among all road users.

#### **5.4.2 Generation of High-definition Maps of Road Environments**

High-definition (HD) maps are highly detailed 3D maps containing the 3D location of all traffic signs, traffic lights, trees, and every relevant object in the road environment. HD maps are considered an essential component of fully autonomous vehicles. An HD map enables the autonomous vehicle to localize itself accurately with respect to the road environment and recognize and react to events on the road, which might not be detected by the sensors on board the vehicle.

While there is currently no standard specifying the format and structure of HD maps, it is widely accepted that the raw material for the generation of HD maps are 3D data, such as LiDAR point clouds, with semantic information representing



**Figure 5.2:** An example of raw point cloud collected by a mobile LiDAR sensor (left), and the classified point cloud (right).

the type of objects present in the data. Efficient generation of HD maps requires automated recognition and classification of various objects in the point cloud. Figure 5.2 shows an example of raw 3D point cloud acquired by a mobile LiDAR sensor and the classified point cloud containing semantic information about the type of objects present in the environment.

Classification methods applied to point clouds of road environments have thus far been less successful due to the complexity of the objects involved. For example, the current top performing approach in the Paris-Lille-3D benchmark (Roynard et al., 2018) achieves a mean intersection over union (IoU) score of only 82.7% (Boulch et al., 2020). The most complex objects for classification are small objects with intra-class variability such as traffic signs, traffic lights, and light poles. For instance, the highest mean IoU for the recognition of poles in the the Paris-Lille-3D benchmark at present is 79.7% (Luo et al., 2020). A fundamental problem contributing to the poor performance of classification methods applied to point clouds, when compared to images, is the scarcity of labelled data for training. The Paris-Lille-3D dataset, which is one of the largest urban point cloud datasets, contains 2479 labelled segments across 50 categories (Roynard et al., 2018), that is an average 50 training samples per category. Other LiDAR datasets for autonomous driving, such as KITTI (Geiger et al., 2012), nuScenes (Caesar et al., 2019), and Waymo Open (Sun et al., 2020), have annotations for fewer object categories. In comparison, the ImageNet Challenge dataset contains roughly 1000 training images in each of 1000 categories (Krizhevsky et al., 2012). The limited availability of training samples from urban point clouds is mainly due to the complexity of annotating point clouds as compared to image labelling.

### 5.4.3 Vehicle Localization

Estimating the location of the vehicle with respect to a map is a basic requirement for autonomous navigation. While the Global Navigation Satellite System

(GNSS) is the primary technology for vehicle localization, in urban environments where GNSS signals are not available, e.g. urban canyons and tunnels, computer vision techniques using images and LiDAR data can be used to estimate the location of the vehicle. Vehicle localization methods based on imagery and LiDAR point clouds can be divided into two categories: local motion estimation and global position estimation (Khoshelham and Ramezani, 2017).

In local motion estimation, the position of the vehicle is determined by estimating its motion with respect to a previously known position. Local motion estimation methods using imagery and LiDAR data are mainly based on visual odometry (Ramezani and Khoshelham, 2018; Ramezani et al., 2018) and simultaneous localization and mapping (SLAM) (Bresson et al., 2017). A major limitation of visual odometry and SLAM approaches to vehicle localization is the drift of the estimated trajectory caused by the accumulation of errors in each local motion estimation step. Overlap detection and loop closing methods, such as OverlapNet (Chen et al., 2020), can be used to correct the drift. However, for vehicle localization correct location estimates are needed in real time and correction of the trajectory with some delay is not practical.

In global position estimation, the position of the vehicle is estimated directly in a global reference coordinate frame by matching the images or LiDAR scans with a georeferenced source of spatial data. Image-based pose regression methods, such as PoseNet (Kendall et al., 2015), estimate the pose of the camera by learning a regressor from a set of images with known pose. LiDAR-based methods, such as L3Net (Lu et al., 2019), learn correspondences between a current LiDAR scan and a set of pre-existing LiDAR scans of the environment to estimate the position of the vehicle. Other methods detect landmarks, such as road signs (Ghallabi et al., 2019) and curbs (Wang et al., 2017), in LiDAR data and match these with a pre-existing map to estimate the location of the vehicle. The prerequisite for all these approaches is the availability of a set of georeferenced images, LiDAR scans, or HD maps of the environment.

Computer vision approaches to vehicle localization are generally considered complementary to GNSS rather than competitive. As such, location estimates from imagery and LiDAR data are often fused with GNSS measurements when available. Gao et al. (2015) and Ilci and Toth (2020) propose methods for the integration of LiDAR localization methods with GNSS and inertial measurements.

## 5.5 Concluding Remarks

The potential of computer vision techniques for urban mobility applications has been demonstrated in many recent works as reviewed in this chapter. However, a few challenges still remain to be addressed. The first challenge is the practicality of machine learning approaches in real scenarios. Most of the existing methods



are based on supervised learning, which requires an off-line training phase and adequate training examples. But, in many practical applications where a plug and play solution is needed unsupervised or semi supervised learning models are preferred. Transfer learning using pre-trained deep networks is a promising solution for image-based methods. However, at present pre-trained models for LiDAR data are scarce and have poor transferability. Generative models and training by synthetic samples are potential approaches to unsupervised and semi-supervised learning which are worth further exploration.

A related challenge for the application of computer vision to urban mobility is the geographical diversity and scene adaptation. Most existing methods are scene-dependent. For example, a machine learning model trained on a dataset captured in Paris might perform poorly on data captured in Melbourne. Domain adaptation methods such as sample weighting and distribution alignment, e.g. using adversarial training, have received little attention so far and are worth further investigation.

Robustness to poor lighting and adverse weather conditions is another important challenge for computer vision methods. Recent research is paying more attention to the development of more robust computer vision methods. This is further promoted by the development of public datasets for autonomous driving which provide annotated images and LiDAR data captured in rain, snow, and night time; see e.g. nuScenes (Caesar et al., 2019) and Canadian Adverse Driving Conditions (CADC) dataset (Pitropov et al., 2020).

## Bibliography

- Acharya, D., Khoshelham, K., and Winter, S. (2017). Real-time detection and tracking of pedestrians in CCTV images using a deep convolutional neural network. In *Proceedings of Research@Locate17, 3-6 April 2017, Sydney, Australia*, volume 1913, pages 31–36.
- Acharya, D., Yan, W., and Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. In *Proceedings of Research@Locate18, 9-11 April 2018, Adelaide, Australia*, volume 2087, pages 33–40.
- Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. (2016). Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pages 850–865. Springer.
- Boulch, A., Puy, G., and Marlet, R. (2020). Fkaconv: Feature-kernel alignment for point cloud convolution. *arXiv preprint arXiv:2004.04462*.
- Bresson, G., Alsayed, Z., Yu, L., and Glaser, S. (2017). Simultaneous localiza-

- tion and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220.
- Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2019). nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*.
- Chen, X., Läbe, T., Milioto, A., Röhling, T., Vysotska, O., Haag, A., Behley, J., Stachniss, C., and Fraunhofer, F. (2020). Overlapnet: Loop closing for lidar-based slam. In *Proc. of Robotics: Science and Systems (RSS)*.
- Chen, Z., Zhang, J., and Tao, D. (2019). Progressive lidar adaptation for road detection. *IEEE/CAA Journal of Automatica Sinica*, 6(3):693–702.
- De Almeida, P. R., Oliveira, L. S., Britto Jr, A. S., Silva Jr, E. J., and Koerich, A. L. (2015). Pklot—a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937–4949.
- Fritsch, J., Kuehnl, T., and Geiger, A. (2013). A new performance measure and evaluation benchmark for road detection algorithms. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1693–1700. IEEE.
- Gao, Y., Liu, S., Atia, M. M., and Noureldin, A. (2015). Ins/gps/lidar integrated navigation system for urban and indoor environments using hybrid scan matching algorithm. *Sensors*, 15(9):23286–23302.
- Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Georgoudas, I. G., Sirakoulis, G. C., and Andreadis, I. T. (2010). An anticipative crowd management system preventing clogging in exits during pedestrian evacuation processes. *IEEE Systems Journal*, 5(1):129–141.
- Ghallabi, F., El-Haj-Shhade, G., Mittet, M.-A., and Nashashibi, F. (2019). Lidar-based road signs detection for vehicle localization in an hd map. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1484–1490. IEEE.
- Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M. (2020). Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969.

- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Ilici, V. and Toth, C. (2020). High definition 3d map creation using gnss/imu/lidar sensor integration to support autonomous vehicle navigation. *Sensors*, 20(3):899.
- Kendall, A., Grimes, M., and Cipolla, R. (2015). PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946.
- Khoshelham, K. and Ramezani, M. (2017). Vehicle positioning in the absence of gnss signals: Potential of visual-inertial odometry. In *Joint Urban Remote Sensing Event*. IEEE.
- Kong, S., Sanderson, C., and Lovell, B. C. (2007). Classifying and tracking multiple persons for proactive surveillance of mass transport systems. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007*, pages 159–163, Los Alamitos, CA, USA. IEEE Computer Society.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.
- Levering, A., Tomko, M., Tuia, D., and Khoshelham, K. (2020). Detecting unsigned physical road incidents from driver-view images. *IEEE Transactions on Intelligent Vehicles*.
- Li, Y., Khoshelham, K., Sarvi, M., and Haghani, M. (2019a). Direct generation of level of service maps from images using convolutional and long short-term memory networks. *Journal of Intelligent Transportation Systems*, 23(3):300–308.
- Li, Y., Sarvi, M., and Khoshelham, K. (2019b). Pedestrian origin-destination estimation in emergency scenarios. In *9th International Conference on Fire Science and Fire Protection Engineering (ICFSFPE)*, pages 1–5. IEEE.
- Li, Y., Sarvi, M., Khoshelham, K., and Haghani, M. (2020). Multi-view crowd congestion monitoring system based on an ensemble of convolutional neural network classifiers. *Journal of Intelligent Transportation Systems*, pages 1–12.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer.
- Liu, X., Deng, Z., and Yang, Y. (2019). Recent progress in semantic image segmentation. *Artificial Intelligence Review*, 52(2):1089–1106.

- Lu, W., Zhou, Y., Wan, G., Hou, S., and Song, S. (2019). L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6389–6398.
- Luo, H., Chen, C., Fang, L., Khoshelham, K., and Shen, G. (2020). Ms-rrfsegnet: Multiscale regional relation feature segmentation network for semantic segmentation of urban scene point clouds. *IEEE Transactions on Geoscience and Remote Sensing*.
- Malinovskiy, Y., Wu, Y.-J., and Wang, Y. (2008). Video-based monitoring of pedestrian movements at signalized intersections. *Transportation Research Record*, 2073(1):11–17.
- Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE.
- Pena-Caballero, C., Kim, D., Gonzalez, A., Castellanos, O., Cantu, A., and Ho, J. (2020). Real-time road hazard information system. *Infrastructures*, 5(9):75.
- Pitropov, M., Garcia, D., Rebello, J., Smart, M., Wang, C., Czarnecki, K., and Waslander, S. (2020). Canadian adverse driving conditions dataset. *arXiv preprint arXiv:2001.10117*.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660.
- Ramezani, M. and Khoshelham, K. (2018). Vehicle positioning in gnss-deprived urban areas by stereo visual-inertial odometry. *IEEE Transactions on Intelligent Vehicles*, 3(2):208–217.
- Ramezani, M., Khoshelham, K., and Fraser, C. (2018). Pose estimation by omnidirectional visual-inertial odometry. *Robotics and Autonomous Systems*, 105:26–37.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.
- Roynard, X., Deschaud, J.-E., and Goulette, F. (2018). Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, 37(6):545–557.

- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.
- Ryan, D., Denman, S., Sridharan, S., and Fookes, C. (2015). An evaluation of crowd counting methods, features and regression models. *Computer Vision and Image Understanding*, 130:1–17.
- Shi, S., Wang, X., and Li, H. (2019). PointRCNN: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779.
- Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
- Song, H.-S., Lu, S.-N., Ma, X., Yang, Y., Liu, X.-Q., and Zhang, P. (2014). Vehicle behavior analysis using target motion trajectories. *IEEE Transactions on Vehicular Technology*, 63(8):3580–3591.
- Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332.
- Sun, P., Kretschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al. (2020). Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- True, N. (2007). Vacant parking space detection in static images. *University of California, San Diego*, 17:659–662.
- Valipour, S., Siam, M., Stroulia, E., and Jagersand, M. (2016). Parking-stall vacancy indicator system, based on deep convolutional neural networks. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 655–660. IEEE.
- Wang, L., Zhang, Y., and Wang, J. (2017). Map-based localization method for autonomous vehicles using 3d-lidar. *IFAC-PapersOnLine*, 50(1):276–281.
- Wang, Q., Zhang, L., Bertinetto, L., Hu, W., and Torr, P. H. (2019). Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the*

*IEEE Conference on Computer Vision and Pattern Recognition*, pages 1328–1338.

Wang, X., Qian, Y., Wang, C., and Yang, M. (2020). Map-enhanced ego-lane detection in the missing feature scenarios. *arXiv preprint arXiv:2004.01101*.

Wu, X., Winter, S., and Khoshelham, K. (2020). Multi-camera tracker for monitoring pedestrians in enclosed environments. In *International Conference on Tools with Artificial Intelligence, ICTAI 2020*.

Yang, Z., Sun, Y., Liu, S., and Jia, J. (2020). 3DSSD: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048.

Yi, S., Li, H., and Wang, X. (2015). Pedestrian travel time estimation in crowded scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3137–3145.

Zarzar, J., Giancola, S., and Ghanem, B. (2019). Efficient tracking proposals using 2d-3d siamese networks on lidar. *arXiv preprint arXiv:1903.10168*.

Zhao, J., Xu, H., Wu, J., Zheng, Y., and Liu, H. (2018). Trajectory tracking and prediction of pedestrian's crossing intention using roadside lidar. *IET Intelligent Transport Systems*, 13(5):789–795.

Zheng, X., Wu, F., Chen, W., Naghizade, E., and Khoshelham, K. (2019). Show me a safer way: Detecting anomalous driving behavior using online traffic footage. *Infrastructures*, 4(2):22.

## 6 Urban Mobility and Parking Demand

STEPHAN WINTER

### Abstract

Parking demand, both current and future, depends on two aspects: the long-term impact of urbanization and urban planning on parking demand, which is not addressed here, and, secondly, the choice of mobility modes, which is discussed here. The choice of mobility modes may, on one hand, require smart parking management and parking information, which is a result of the tracking technology discussed before. On the other hand an informed or incentivized choice of mobility modes may even lead to less parking demand.

### Keywords

Active mobility, disruptions to motorized mobility, mobility as a service

### 6.1 Introduction

Of the triad of instruments to counter parking pressure (avoiding parking demand – shifting mode choice – improve parking space supply, (National Transport Development Policy Committee, 2012)), this chapter will point out the potential of emerging geospatial technology to either avoid, or at least reduce, the demand for parking in urban areas, or to ease the use of other forms of urban mobility than using the private car. People travel into the city for a reason. They wish to participate in activities offered by the city – work, learn, shop, entertain – and are received by scarce space due to the high price of land. The challenge of high demand on one hand, and low supply on the other is exacerbated by the general preference to travel in private cars, and by, on average, low occupancy rates in these cars, which jointly produces competition for inner-urban street space as much as for parking space.

With our focus on geospatial technology we leave out a wide range of other measures – including other technological measures – to reduce this parking pressure on urban centers. For example, we abstain from discussing urban planning and design questions here, although they are absolutely critical to foster active mobility. A city that is not ‘walkable’ (Speck, 2013) or ‘bikeable’ (McNeil,

2011; Winters et al., 2016) will not convince citizens to change their mobility behavior just by providing information services.

We also abstain from discussing non-geospatial technologies that may impact actual travel demand, such as technologies supporting working from home, on-line learning, or tele-health. Their theoretical potential has been dramatically shown in the lockdowns in the 2020 pandemic. The imposed work-from-home regulations in Melbourne, Victoria, for instance, led to a reduction of inner-city traffic by 88% in the first lockdown (March), or by 85% in the second (July), respectively<sup>1</sup>. Such data is corroborated by Google's *COVID-19 Community Mobility Reports* from around the world<sup>2</sup>, which are based on phone activities and are heavily aggregated, hence less specific. According to Google, Melbourne's mobility trends in relation to workplaces had decreased by 48% on 21 August 2020 (in the middle of the second lockdown) compared to a longer-term baseline. On the same day, the average for the whole of India has been down by 32%, despite less reliable internet and thus less reliable technologies for working from home.

Instead, we look at providing services that support the current level of mobility demand by making better decisions on mode choices or travel behavior. In this chapter we will look at:

- technologies that support more active mobility (again, we abstain from urban planning and design questions here);
- technologies that change the nature of motorized mobility and as a by-product the demands for parking; and
- technologies that ease the interaction with urban mobility and improve the coordination between modes of urban mobility.

With these foci, we highlight opportunities that are fully aligned with the sixth of the ten Aalborg Commitments<sup>3</sup>, which were agreed at the Fourth European Conference on Sustainable Cities and Towns (2004) and signed by more than 700 mayors from across Europe and Africa. This sixth commitment simply states "Better mobility, less traffic – We recognize the interdependence of transport, health and the environment and are committed to strongly promoting sustainable mobility choices. We will therefore work to:

1. reduce the necessity of private motorized transport and promote attractive alternatives accessible to all.
2. increase the share of journeys made by public transport, on foot and by bicycle.

---

<sup>1</sup><https://ab.co/34YXcuY> – Australian Broadcasting Corporation, 22 July 2020

<sup>2</sup><https://www.google.com/covid19/mobility/>

<sup>3</sup><https://sustainablecities.eu/the-aalborg-commitments/> – ICLEI, 2004



3. encourage transition to low-emission vehicles.
4. develop an integrated and sustainable urban mobility plan.
5. reduce the impact of transport on the environment and public health.”

In order to discuss mode shifts towards sustainable mobility choices one might think that a certain theoretical equilibrium might exist for a sustainable city, for instance, 30 % for the active modes such as walking and cycling, and 30 % for public transport. Such aspirational targets, however, are missing from the Aalborg Commitments and for good reasons. One reason is that such figures would also ignore the largely different starting conditions in the cities around the world (Langeland, 2015) – Table 6.1 illustrates this argument by mode share figures, without diving into the reasons for these observed mode shares.

**Table 6.1:** Mode share in various cities.

	walking	cycling	public transport	private car	(year)
Melbourne <sup>4</sup>	4 %	2 %	19 %	76 %	2016
Mumbai <sup>5</sup>	7 %	6 %	52 %	15 %	2011
Münster <sup>6</sup>	16 %	38 %	10 %	36 %	2007

In fact, sustainable transport research provides a range of tools to improve urban mobility but in a sustainable manner (e.g., Goldman and Gorham, 2006; Litman, 2007; Malasek, 2016). A remaining challenge – for the research community as well as the society at large – is to define what a sustainable activity is, including urban mobility. Broad agreement prevails that current urban mobility is not sustainable (e.g., Greene and Wegener, 1997), and that *more sustainable* solutions have to be found: solutions that are *less harmful* to the environment, society, and less detrimental to sustainable economic development, especially, when considering the future generations.

Litman (2007), referring to some indicators, defines a sustainable transportation system as one that satisfies the mobility needs in a society safely and inclusively in a manner that “limits emissions and waste within the planet’s ability to absorb them, minimizes consumption of nonrenewable resources, limits consumption of renewable resources to the sustainable yield level, reuses and recycles its components, and minimizes the use of land and the production of noise” (Litman, 2007, p. 11). Note that these terms are still, and unavoidably, accepting some non-sustainable consumption of resources and land.

<sup>4</sup>Commuter trips, [https://en.wikipedia.org/wiki/Modal\\_share](https://en.wikipedia.org/wiki/Modal_share)

<sup>5</sup>Commuter trips, [https://en.wikipedia.org/wiki/Modal\\_share](https://en.wikipedia.org/wiki/Modal_share)

<sup>6</sup>Data is taken from (Langeland, 2015).

Urban parking has hardly been addressed in the research on sustainable transport. But this is unjustified. Parking is only a by-product of urban mobility, but – returning to Litman’s definition of a sustainable transportation system above – parking contributes significantly to the space consumption of urban mobility, and its contributions to urban traffic and emissions are widely acknowledged as well. Accordingly, the mechanisms to address parking in the context of sustainable transport are by reducing individual car usage and thus to reduce both cruising for parking and the demand for space for parking. The following sections focus on these mechanisms.

## 6.2 Active Mobility

*Active mobility* (or *active transportation*) is the generic term for all non-motorized travel modes. Prime examples are walking and cycling. Active mobility involves moving around (e.g., by walking or cycling), transporting other people (e.g., by rickshaw runners – where *rickshaw* literally means *human-powered vehicle*), or transporting goods (e.g., by bicycle couriers). The first category – moving around – uses in its purest form only locomotion: human physical activity without further means of support, such as walking or running or – in aquatic environments – swimming. Other forms of moving around are supported by some agent or non-motorized vehicle. These forms include traveling on the back of an animal, or using mechanical vehicles such as bicycles, roller skates, kick scooters, wheelchairs, or carts. The other two categories – being moved around – rely on somebody else’s physical activity. A categorization has been made in Table 6.2.

**Table 6.2:** Categories of active mobility, with examples.

	locomotion	vehicle
self	walking	bicycling
other	riding	rickshaw

Often active mobility is promoted for health reasons. A proverb attributed to Hippocrates says “Walking is man’s best medicine”. (More) walking helps prevent non-communicable diseases that are dramatically increasing globally, including Type II diabetes, obesity, and coronary heart diseases (Batman, 2012; Giles-Corti et al., 2016; Pucher et al., 2010).

In the context of this book other aspects are more relevant though. Most importantly, the different modes of active mobility have a significantly smaller footprint on parking space demand compared to private cars (see for example Figure 8.2), with locomotion having no parking space demand at all. On the other hand, the various forms of active mobility have different ranges compared to the private car, i.e., they do not directly replace the car. The obvious argument is illustrated by Table 6.3, which shows the use of specific modes dependent on the distance

traveled, based on household travel survey data from England (Department of Transport, 2020). For instance, in England 81 % of all trips shorter than 1.6 km (1 mile) were undertaken on foot, 1.1 % on a bicycle, 11.4 % on driving a private car, and 6.7 % as a passenger in a vehicle (including public transport). These percentages shift significantly towards either driving a car or being a passenger with increasing travel distances.

**Table 6.3:** Percentages of trips by mode, in distance ranges.

	≤ 1.6 km	≤ 3.2 km	≤ 8 km	≤ 16 km	≤ 40 km	≤ 80 km
walk	80.9 %	33.4 %	4.7 %	0.3 %	0.1 %	0.0 %
bicycle	1.1 %	3.1 %	2.8 %	1.6 %	1.0 %	0.6 %
driver	11.4 %	39.9 %	60.5 %	65.4 %	68.3 %	67.7 %
passenger	6.7 %	23.6 %	31.9 %	32.7 %	30.6 %	31.8 %

While these numbers illustrate that active mobility will not replace the car, their magnitudes will differ between cultures, between social groups, between age groups, and also between towns and countries. However, urban planners work with an often cited 5-minute rule (van Soest et al., 2020). It suggests that people are willing to walk distances of 400 to 800 meters, corresponding to five to ten minutes, before considering other modes of mobility. This rule of thumb is often used for catchment planning of public transport stops: five minutes for bus or tram stops, and ten minutes for rail. Specific studies tend to confirm this broad rule, and shed more light on the causes of this variability. This rule can also be inverted to assess the quality of a public transport system in a city. For example, El-Generidy et al. (2014) show that Montreal has an average distance of 524 m between home-based trip origins and the public transport stop taken, and 1,259 m for home-based commuter rail trip origins. But the willingness to walk depends also on the purpose: According to Table 6.3 people are willing to walk farther than only 400 m to 800 m. Obviously the willingness to walk to public transport, shopping, or personal errands is on the shorter side, while people are willing to walk farther for other purposes, such as recreation.

Pedestrian and cyclist behavior – i.e., the willingness of people to choose active modes of mobility – will also be influenced and shaped by the information provided to them: information creating a better awareness of the urban layout and travel options (Laurier et al., 2016; Peiravian et al., 2014), integration with data on environmental or health risk levels (Wang et al., 2020), integration with live public transport data (Daniels and Mulley, 2013; Mavoa et al., 2012), and record keeping for personal fitness or competition purposes (Gu et al., 2017; Higgins, 2016).

Last but not least, active mobility is a matter of choice only for the physically abled, and excludes the elderly (Musselwhite and Haddad, 2010), parents with children, and other people handicapped in their mobility. An inclusive society

has always provided alternative modes of mobility. Accordingly, a significant part of research and the development of services are focused on the integration of active mobility (walking, mostly), and public transport. An underdeveloped area, however, is the integration of walking and driving a private car, which is relevant for this book as well. When driving a car, the first and last leg of a trip are always traveled on foot. This walking part has a tendency of being ignored when choosing the car, at least in the literature. When literature studies the perceived costs of parking, the focus is solely on the cruising for “cheap” parking (Shoup, 2005). We are not aware of initiatives combining parking with encouragement for active mobility.

But let us have a closer look at walking, which is by far the most frequent form of active mobility according to Table 6.3 and all similar data. Such a closer look reveals an astonishing complexity in human mobility behavior: there is a lack of (accessible) network data, and gaps in conceptual research – all in comparison to the better understood and modeled motorized forms of mobility, and even cycling, which, at least in cities, is happening in shared spaces with cars.

Obviously, the digital infrastructure of a smart city requires a comprehensive representation of the complex domain of pedestrian access. The common fundamental data structure for representing mobility space is a *graph*, consisting of *nodes* representing locations, *edges* representing access connections between these locations, and any constraints on these connections. A graph, as a linear structure, is the result of a projection of a volume – the space pedestrian movement takes – to the plane – a plan or map – and a further projection of the two-dimensional map space to the linear structure of a graph.

Yet, even now, pedestrian mobility studies rely on graphs derived as by-products of primarily road network data or indoor floor plans, rather than from primary mapping. But walking in urban space happens to a large part also in separation from the road network (Chin et al., 2008), and in both elongated and more compact spatial forms (Table 6.4). Walking spaces are spaces such as sidewalks, laneways closed for vehicles, pedestrian zones, city squares, park paths, park greens, pedestrian under and overpasses, private passages with restricted access, such as passages through malls or train stations, and corridors or halls (Figure 6.1). Pedestrians do move freely in these spaces, which they consider open for their locomotion – inspiring Seamon to call it a *place ballet* (Seamon, 1980) and others to identify simple rules of motion that aggregate to this complex behavior (Moussaïd et al., 2011). Pedestrians are also less constrained by traffic regulations than, for example, cars or bicycles. For example, they can turn around at any time and anywhere. Pedestrians also use shared space with vehicles when they cross a road. Road crossings can be unmarked (e.g., a suburban crossroad), marked, regulated by zebra crossings, or controlled by traffic lights. Also, in many countries it is legal to cross a road anywhere.



**Figure 6.1:** Pedestrian movements in large spaces (here a pedestrian zone (top) and an airport terminal (bottom)) form complex patterns, in contrast to regulated road traffic. Source (top): <https://bit.ly/3vStqUL> (© Mattes, 2005, public domain, modified). Source (bottom): <https://bit.ly/360dLHm> (© Minseong Kim, 2016, CC BY-SA 4.0).

Given the complexity of walking pathways these derivation processes introduce significant under-specification (Brezina et al., 2017; Pafka, 2017; Marshall et al., 2018) and contradictions from the various levels of detail of the maps (Ericson et al., 2020). The competing, varied modeling approaches of pedestrian mobility space (e.g., Stoffel et al., 2007; Becker et al., 2008; Jensen et al., 2009; Liu and Zlatanova, 2012; Isikdag et al., 2013; Kim et al., 2014; Yang and Worboys, 2015; Walton and Worboys, 2012; Stahl and Schwartz, 2010) lead to incompatible graphs (Vanclooster et al., 2016), ranging from purely topological connectivity graphs to numerous forms of geometrically embedded graphs (Figure 6.2). In addition, pedestrians conceptualize indoor and outdoor environments differently, which leads to a confusion between graph elements and mental concepts (Guidice et al., 2010; Schaap et al., 2011; Rüetschi and Timpf, 2005).

To derive comprehensive graph representations of the pedestrian mobility space in a city – in support of more active mobility – requires mechanisms to capture and integrate data on the pedestrian mobility space. The graph should maintain desirable detail for active mobility, such as width, surface, and barriers – which is essential information for parents pushing perambulators, for wheelchairs, or for



**Figure 6.2:** A graph connecting the spaces that can be accessed from each other (topological connectivity – left) and a graph closely approximating human movement (Stahl and Schwartz, 2010) (geometrically embedded graph – right).

skateboards – and the restrictions on private pedestrian mobility spaces, such as temporal ones (e.g., opening hours), or authorization ones (e.g., key or smart card access, or payment gates) (Richter et al., 2011). Furthermore, the integration of various data sources (or various graph models; see above) requires conceptual mappings of any of the various modeling approaches listed above onto an agreed standard model, and as long as this standard model does not exist globally (Harrison et al., 2020; Beil and Kolbe, 2017) at least within the information domain of the jurisdiction of a (smart) city a common model needs to be agreed upon.

**Table 6.4:** Categories of pedestrian traffic space, with examples, with various complexities of pedestrian movement behavior.

	<b>public</b>	<b>private</b>
<b>elongated</b>	sidewalk	passage
<b>compact</b>	city square	train station hall

Such graph representations of active mobility spaces – here of walking – are essential for smart city applications. Developing support for other forms of mobility than the private car, especially for short-distance trips in inner-urban areas, will help to reduce demand for the private car, and thus for parking a private car in inner-urban areas. A data infrastructure covering the active mobility spaces in these areas enables the deployment of mobile location-based services to motivate people for physical activity. These services could apply data analysis to suggest personalized activities either to achieve set goals, or to adapt to physical capabilities. Network analysis, combined with pedestrian counters in the inner-urban areas (see Section 3.3), offers to simulate pedestrian mobility, for example, to prepare for disasters (evacuation planning) or to upgrade the pedestrian infrastructure (urban planning). Finally, this graph can be integrated with graphs of other modes of transportation, enabling the development of multimodal travel planners – with a special focus on encouraging the walking components of trips (integrated transport planning). The positive effect on parking pressure would primarily emerge from avoiding short-distance car trips.

## 6.3 Motorized Mobility

Intelligent transportation systems must cater for improvements of the capacity of the *parking* infrastructure as well. Improvements of the environmental impact of the existing transport infrastructure would at least cope with the negative impact of parking-related behavior, such as cruising for parking. Thus, we will review here three emerging technologies that will significantly impact on urban parking: electrification of vehicles, autonomy, and collaboration. All three together will have a significant impact on urban mobility, as we will discuss in the next section.

### 6.3.1 Electrification

The current electrification of vehicles has an immediate effect on the environmental impact of motorized traffic in a city: emissions – both greenhouse gases and particles – are no longer generated by the vehicles themselves (but potentially in the energy generation process if it is not renewable). Also, noise emissions by vehicles are reduced. But resource consumption is still a challenge, since the replacement of fossil fuels by renewable energy increases the demand for batteries, and the current techniques for battery recycling are still highly loss-making – as long as mining the resources is cheaper. Hence, while the life-cycle assessment of the ecological performance of electrical vehicles is controversial, the aspect of importance in our context is that electric vehicles have different parking patterns due to their battery recharging needs. At least in the introduction period of electrical vehicles not every parking spot will be equipped with a charging station (Figure 6.3), meaning that electrical vehicles will, at least initially, travel further in search of parking.



**Figure 6.3:** Parking with a charging station for electrical vehicles. Source: <https://bit.ly/3rfRnSu> (© Aschroet, 2019, public domain, modified).

### **6.3.2 Autonomous Vehicles**

The Society of Automotive Engineers defined six levels of driving automation that are widely adopted or referred to. They range from 0 (fully manual) over levels of driver assistance systems and partial automation to automation under certain conditions (Level 3), high automation where cars do not require human interaction in most circumstances (Level 4 – mostly operating in limited environments) and full automation (Level 5). At the time of writing this book, Level 4 vehicles are already commercially available and on the road, for example the early adapting self-driving passenger shuttles. Level 5 vehicles are currently being tested but are not yet commercially available. Furthermore, the regulatory frameworks for their legal operation on the road are still missing. Autonomous vehicles will be electrical vehicles because of better integration of the two technologies of energy and automation. An electric engine is simpler and has fewer moving pieces than a combustion engine. Also, wireless charging of batteries integrates seamlessly with autonomy.

The main driver for the development of autonomously driving vehicles is the elimination of the human factor in the driving process, which is perceived as a risk for safety: the computer controlling an autonomous vehicle never gets tired or distracted, and is generally faster in data processing. But autonomous driving at Level 5 has an economic impact as well: the computer is significantly cheaper than a commercial human driver, and not bound to hours of work.

Most importantly, however, autonomous vehicles (Level 4 or 5) impact urban parking. They no longer need a parking spot at the destination of a trip, but can autonomously search for a parking spot after dropping off their passengers – or continue traveling, and pick up other passengers, i.e., not require a parking spot at all. This disjunction of travel demand and parking demand will reduce the pressure on parking in high-demand destinations such as city centers. It will also disrupt commercial parking business models that have made significant real-estate investments in city centers. It will not, however, reduce the traffic in city centers since vehicle kilometers traveled will increase – by empty cruising.

An autonomous vehicle heavily relies on the geospatial foundations that we have discussed so far. Equipped with a large range of sensors (Chapters 4, 5), it is constantly tracking its own location (Chapter 3) both in a global reference frame, as well as in a local reference frame, i.e., relative to the road marks and to other vehicles or road users around (Chapter 2).

### **6.3.3 Connected and Collaborating Vehicles**

Finally, autonomous vehicles will have another opportunity to shine, namely when enabled with vehicle-to-vehicle communication (see Section 3.2.2). While communication between human drivers is severely limited by lack of a suitable com-



munication channel, as much as the time people require to communicate, vehicle-to-vehicle communication (V2V, as well as all other V2X variants) facilitate near real-time collaboration between autonomously driving vehicles. Vehicle-to-vehicle communication, since it is radio-based, has short latency – critical for safety relevant applications – and is not bound to lines of sight. Coming back to parking, however, collaboration opens new avenues for crowdsourcing and sharing parking information (Bock and Sester, 2016; Bock and Di Martino, 2017).

Collaboration, however, can extend beyond other vehicles. It can involve traffic management platforms that maintain a real-time awareness of traffic and interact for management purposes, or intermodal transport platforms that coordinate connectivity in transit, or vehicle makers that can maintain an awareness of the health of a vehicle and interfere for maintenance purposes. A significant domain of collaboration, however, is with people. This kind of collaboration, can express through modes that are directly perceptible by the human senses, such as sound, light, or driving behavior – e.g., (Gupta et al., 2019) – but also radio-based, with a smart communication device on the other end mediating this communication with people. This latter collaboration, with people, can be applied for example to coordinate the movements in road space between vehicles, pedestrians and cyclists. It can also be used, as we will see in the next section, for coordinating with people on their mobility demands.

Vehicle-to-anything communication (V2X) can be put into two categories: the local communication of a vehicle to another vehicle or other road users, or reaching out to global communication channels (vehicle-to-infrastructure, vehicle-to-network) for global vehicle coordination and collaboration. In the domain of urban parking, both have their merits. A parking information system requires a global picture of available parking spaces, and if this system should be fed by vehicles in a crowdsourced manner a communication channel to the central system is required. However, there is a strong reason to consider local collaboration, and this is because only nearby parking spaces are relevant for a vehicle currently searching for a parking space. The high demand for urban parking, especially on-street parking, means that any information is caught between *velocity* and *veracity* of big data (Li et al., 2016; Zhu et al., 2019), gets outdated fast, and hence, is of value only in a local context.

Local communication and collaboration about detectable spatial information (such as occupancy of parking spots) is the domain of mobile wireless geosensor networks (Duckham, 2013), a specialization of the more general wireless sensor networks (Zhao and Guibas, 2004). Other terms used in the literature, which are less emphasizing the sensing aspect and more emphasizing the carrier, are vehicular ad-hoc networks, or VANETs (Hartenstein and Laberteaux, 2010). Their ability of ad-hoc connectivity is also emphasized by wireless ad-hoc networks, or WANETs, or mobile ad-hoc networks, MANETs. All these networks rely on Wi-Fi, mobile telephony, and other communication services. The principle

is explained by (Toh, 2001): “An ad-hoc wireless network is a collection of two or more devices equipped with wireless communications and networking capability. Such devices can communicate with another node that is immediately within their radio range or one that is outside their radio range. For the latter scenario, an intermediate node is used to relay or forward the packet from the source toward the destination. An ad-hoc wireless network [. . .] can be [formed and] de-formed on-the-fly without the need for any system administration.”

Adding sensors to the communication network, and looking at parking occupancy as detected by vehicles, we can again distinguish two kinds of observations. The vehicle can observe its own behavior, e.g., when leaving a parking spot (see Chapter 13), or the vehicle can observe parking spots nearby, in passing. The prior observation is part of a trajectory, and thus an observation in a Laplacian frame of reference, while the second kind of observation happens in a Eulerian frame of reference (see Section 3.4). In addition, a vehicle can be an intermediate node, transferring information that it has not observed itself.

## 6.4 Disrupting Urban Mobility

Many reasons speak against owning a private car. Here is another one: What if we could move with the comfort, privacy and flexibility of a vehicle without the necessity to own it? To maintain it? To park it? And – because this is about driving without a driver – for significantly lower costs than current taxis or their gig counterparts, the ride-hailing services? In fact, for lower costs than taking our own car? Since with autonomous driving most of the parking-related costs are gone, and yet the passenger gets dropped off at their destination there is no additional loss of time in searching for a parking space and walking back.

This is the promise of autonomous driving, connected and collaborating vehicles: they will provide mobility services on demand by sharing themselves as a resource. The sharing goes beyond current car sharing schemes (Millard-Ball et al., 2005) by vehicles that do not need to be parked by the driver and that, autonomously and collaboratively, re-balance following the actual or anticipated demand (Agatz et al., 2012). The sharing of the resource can happen exclusively with an individual or a group (‘taxi’), or allow for pooling of individuals with different but compatible travel needs (‘shuttle’). In this future scenario too, the boundaries between ridehailing and ridesharing disappear: Ridehailing services are providing taxi services outside the regulated taxi market where a commercial driver offers transportation (Clewlow and Mishra, 2017; Henao and Marshall, 2019; Young and Farber, 2019) while ridesharing services provide a resource sharing of a driver with her own mobility demand (Chan and Shaheen, 2012; Furuhashi et al., 2013). If, in the future, no driver is required, due to full automation, ridehailing and ridesharing merge with carsharing (Shaheen and Cohen, 2007; Millard-Ball et al., 2005), leaving the car as the only commodity. In addition,

this (shareable) commodity itself will become cheaper, due to the mechanically simpler electric motor and the use of cheap energy from renewable resources (Granovskii et al., 2006; Bösch et al., 2018). Another perspective on urban mobility – that of equity in access and participation – suggests that we also rethink the commodification of transport, and instead set up basic services. Cheap and shared vehicles could provide these mobility services. For example, in areas that are currently not well served by mass transport (outer suburbs), or in areas with highly diverse demands (city centers). Literature studies already propose ridesharing as a solution to the first/last leg problem (Shaheen and Chan, 2016; Navidi et al., 2019).

For these reasons it is becoming clearer that autonomous and collaborating vehicles will have a significant impact on urban mobility, so much that some call the impact *disruptive* (Meyer and Shaheen, 2017). While this can be said for the economic arguments just made, another reason for disruption lies in the need for redesigning of urban traffic space as well as of regulatory frameworks to adapt to (full) autonomy. Again another reason lies in the opportunity for rethinking the mode collaboration and integration – options for a *mobility-as-a-service* (Jit-trapirom et al., 2017) – and reshaping people’s interaction with urban mobility.

## 6.5 Interacting with Urban Mobility

An intelligent transportation system, with its design goal of making smarter use of existing transport infrastructure, is not just intelligent because it is able to operate autonomously (“to use data to improve the capacity, safety, and environmental impact of existing transport infrastructure resources”, as we said in Chapter 3.1). It is also intelligent because it communicates with people on their terms (Turing, 1950; Winter and Wu, 2009).

Such an abstract expectation – communicating on human terms – means different things to different groups of people. Road authorities and transport managers have a different view on urban mobility, including parking, than the travelers themselves. A managerial interaction with urban mobility includes dashboards and life maps supporting human decision making, and traffic interventions that realize these human decisions. These tasks can be further automated to some degree but still need to communicate to the people overseeing the urban space and the mobility happening in this space. Travelers, on the contrary, interact with the mobility options available to them, and the travelers’ decision making is limited to their own transport benefit. They seek information about their options, choose, pay, and adapt flexibly to changing circumstances. They may look at their urban mobility options from a system’s perspective, across modes and providers, which is the emerging domain of Mobility-as-a-Service (Exposito-Izquierdo et al., 2017). Or they may look at their urban mobility options from a pre-conceived mode’s perspective, such as a car driver’s perspective, which is

addressed by mode specific navigation tools. They also interact with embodied intelligent systems: “Intelligent” vehicles. Examples are driver assistance systems communicating to their drivers, autonomously driving vehicles interacting with their passengers, or pedestrians interacting with autonomously driving vehicles.

Each of these tools should consider parking as a core component in their own way. From a transport management perspective, this includes tracking parking lot occupancy and providing parking guidance systems. For a traveler, when a mode mix includes a private car this car has to be parked somewhere. Thus, intelligent travel support systems ease the finding of a suited and unoccupied parking space.

One interface to intelligent transportation systems is a particular human one: common language. While conversational assistants have made some inroads to general question-answering already, conversations about spatial and spatiotemporal configurations remain a challenge. The challenge has multiple reasons, but for a start, already Klein (1982) and Wunderlich and Reinelt (1982) identified four phases in the communication between a human wayfinder and an informant, which go beyond the design of current question-answering principles:

- the initial phase: a wayfinder asks an informant for directions,
- the center phase: the informant provides route directions,
- the securing phase: either the wayfinder or the informant want to make sure that the wayfinder has understood the given route directions, and
- the closing phase: participants look for closure and separation.

The question (initiation) and answer (center phase) are covered in this structure, although conversational assistants are already challenged to just answer spatial questions (Hamzei et al., 2019). The neglected, and truly challenging part, however, is the securing phase: a capacity of the machine to engage in discourse on spatial configurations, temporal relationships, and movement.

Besides a verbal channel of communication, intelligent transportation systems can also communicate graphically with people. The transition of a traditional interaction with taxi services (hailing at the curbside, or calling) to map-based interfaces of ridehailing platforms has significantly shifted the goalposts and were critical to the market success for ridehailing services. But even these map-based interfaces can be scrutinized and improved (Rigby et al., 2016; Rigby and Winter, 2016). A further step in complexity is required from mode- (or platform-) specific interfaces to intermodal mobility. For example, Pandey et al. (2019) illustrate that ridehailing operators would benefit from integrating their services on a meta-platform.

## 6.6 Conclusions

This chapter discussed the technologies and incentives that affect peoples' mode choice behavior, and thus, the potential impact on parking demand within the city. It covered the push for active mobility as a focus area, the future demands of motorized mobility, and the future usability of urban mobility systems from an interaction perspective. It did neglect aspects of urban planning and transport systems planning that will have a more long-term impact on parking demand.

## Bibliography

- Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303.
- Batman, D. C. (2012). Hippocrates: 'walking is man's best medicine!'. *Occupational Medicine*, 62(5):320–322.
- Becker, T., Nagel, C., and Kolbe, T. H. (2008). A multilayered space-event model for navigation in indoor spaces. In Lee, J. and Zlatanova, S., editors, *3D Geo-Information Sciences*, Lecture Notes in Geoinformation and Cartography, pages 61–77. Springer, Berlin.
- Beil, C. and Kolbe, T. H. (2017). Citygml and the streets of new york - a proposal for detailed street space modelling. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-4/W5:9–16. ISPRS-Annals.
- Bock, F. and Di Martino, S. (2017). How many probe vehicles do we need to collect on-street parking information? In *5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 538–543. IEEE Xplore.
- Bock, F. and Sester, M. (2016). Improving parking availability maps using information from nearby roads. *Transportation Research Procedia*, 19:207–214.
- Bösch, P. M., Becker, F., Becker, H., and Axhausen, K. W. (2018). Cost-based analysis of autonomous mobility services. *Transport Policy*, 64:76–91.
- Brezina, T., Graser, A., and Leth, U. (2017). Geometric methods for estimating representative sidewalk widths applied to vienna's streetscape surfaces database. *Journal of Geographical Systems*, 19(2):157–174.
- Chan, N. D. and Shaheen, S. A. (2012). Ridesharing in north america: Past, present, and future. *Transport Reviews*, 32(1):93–112.

- Chin, G. K. W., Van Niel, K. P., Giles-Corti, B., and Knuiiman, M. (2008). Accessibility and connectivity in physical activity studies: The impact of missing pedestrian data. *Preventive Medicine*, 46(1):41–45.
- Clewlow, R. R. and Mishra, G. S. (2017). Disruptive transportation: The adoption, utilization, and impacts of ride-hailing in the united states. Research report, Institute of Transportation Studies, University of California, Davis.
- Daniels, R. and Mulley, C. (2013). Explaining walking distance to public transport: The dominance of public transport supply. *Journal of Transport and Land Use*, 6(2):5–20.
- Department of Transport (2020). Mode of travel: NTS0308 – average number of trips by trip length and main mode: England.
- Duckham, M. (2013). *Decentralized Spatial Computing*. Springer, Heidelberg.
- El-Geneidy, A., Grimsrud, M., Wasfi, R., Tétreault, P., and Surprenant-Legault, J. (2014). New evidence on walking distances to transit stops: identifying redundancies and gaps using variable service areas. *Transportation*, 41(1).
- Ericson, J. D., Chrastil, E. R., and Warren, W. H. (2020). Space syntax visibility graph analysis is not robust to changes in spatial and temporal resolution. *Environment and Planning B: Urban Analytics and City Science*, 0(0).
- Exposito-Izquierdo, C., Exposito-Marquez, A., and Brito-Santana, J. (2017). Mobility as a service. In Song, H., Srinivasan, R., Sookoor, T., and Jeschke, S., editors, *Smart Cities: Foundations, Principles, and Applications*, pages 409–436. John Wiley & Sons, Inc., Hoboken, NJ.
- Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57(0):28–46.
- Giles-Corti, B., Vernez-Moudon, A., Reis, R., Turrell, G., Dannenberg, A. L., Badland, H., Foster, S., Lowe, M., Sallis, J. F., Stevenson, M., and Owen, N. (2016). City planning and population health: a global challenge. *The Lancet*, 388(10062):2912–2924.
- Goldman, T. and Gorham, R. (2006). Sustainable urban transport: Four innovative directions. *Technology in Society*, 28(1):261–273.
- Granovskii, M., Dincer, I., and Rosen, M. A. (2006). Economic and environmental comparison of conventional, hybrid, electric and hydrogen fuel cell vehicles. *Journal of Power Sources*, 159(2):1186–1193.
- Greene, D. L. and Wegener, M. (1997). Sustainable transport. *Journal of Transport Geography*, 5(3):177–190.

- Gu, F., Khoshelham, K., Shang, J., Yu, F., and Wei, Z. (2017). Robust and accurate smartphone-based step counting for indoor localization. *IEEE Sensors Journal*, 17(11):3453–3460.
- Guidice, N. A., Walton, L. A., and Worboys, M. (2010). The informatics of indoor and outdoor space: A research agenda. In *2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, page 47–53. ACM.
- Gupta, S., Vasardani, M., and Winter, S. (2019). Negotiation between vehicles and pedestrians for the right of way at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):888–899.
- Hamzei, E., Winter, S., and Tomko, M. (2019). Initial analysis of simple where-questions and human-generated answers. In Timpf, S., Schlieder, C., Kattenbeck, M., Ludwig, B., and Stewart, K., editors, *14th International Conference on Spatial Information Theory*, Leibniz International Proceedings in Informatics, pages 12:1–12:8. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany.
- Harrison, J., Sorenson, M., and Portele, C. (2020). Routing standards working group charter. Report, Open Geospatial Consortium.
- Hartenstein, H. and Laberteaux, K. (2010). *VANET: Vehicular Applications and Inter-Networking Technologies*. John Wiley & Sons, New York.
- Henao, A. and Marshall, W. E. (2019). The impact of ride-hailing on vehicle miles traveled. *Transportation*, 46(6):2173–2194.
- Higgins, J. P. (2016). Smartphone applications for patients’ health and fitness. *The American Journal of Medicine*, 129(1):11–19.
- Isikdag, U., Zlatanova, S., and Underwood, J. (2013). A bim-oriented model for supporting indoor navigation requirements. *Computers, Environment and Urban Systems*, 41(9):112–123.
- Jensen, C., Lu, H., and Yang, B. (2009). Graph model based indoor tracking. In *Tenth International Conference on Mobile Data Management*, pages 122–131. IEEE.
- Jittrapirom, P., Caiati, V., Feneri, A.-M., Ebrahimigharehbaghi, S., González, M. J. A., and Narayan, J. (2017). Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges. *Urban Planning*, 2(2):13–25.
- Kim, J.-S., Yoo, S.-J., and Li, K.-J. (2014). Integrating indoorgml and citygml for indoor space. In Pfoser, D. and Li, K.-J., editors, *Web and Wireless Geographical Information Systems*, volume 8470 of *Lecture Notes in Computer Science*, book section 12, pages 184–196. Springer, Berlin.

- Klein, W. (1982). Local deixis in route directions. In Jarvella, R. J. and Klein, W., editors, *Speech, Place, and Action*, pages 161–182. John Wiley & Sons, Chichester.
- Langeland, A. (2015). Sustainable transport in cities: Learning from best and worst practice? In Brebbia, C. A. and Miralles i Garcia, J. L., editors, *Urban Transport XXI*, volume 146, pages 93–103. WIT Transactions on The Built Environment.
- Laurier, E., Brown, B., and McGregor, M. (2016). Mediated pedestrian mobility: Walking and the map app. *Mobilities*, 11(1):117–134.
- Li, S., Dragicevic, S., Anton, F., Sester, M., Winter, S., Coltekin, A., Pettit, C., Jiang, B., Haworth, J., Stein, A., and Cheng, T. (2016). Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115(May):119–133.
- Litman, T. (2007). Developing indicators for comprehensive and sustainable transport planning. *Transportation Research Record*, 2017(1):10–15.
- Liu, L. and Zlatanova, S. (2012). A semantic data model for indoor navigation. In Claramunt, C. and Li, K.-J., editors, *Fourth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, pages 1–8, 2442618. ACM.
- Malasek, J. (2016). A set of tools for making urban transport more sustainable. *Transportation Research Procedia*, 14:876–885.
- Marshall, S., Gil, J., Kropf, K., Tomko, M., and Figueiredo, L. (2018). Street network studies: from networks to models and their representations. *Networks and Spatial Economics*, 18:735–749.
- Mavoa, S., Witten, K., McCreanor, T., and O'Sullivan, D. (2012). Gis based destination accessibility via public transit and walking in auckland, new zealand. *Journal of Transport Geography*, 20(1):15–22.
- McNeil, N. (2011). Bikeability and the 20-min neighborhood: How infrastructure and destinations influence bicycle accessibility. *Transportation Research Record*, 2247(1):53–63.
- Meyer, G. and Shaheen, S. A. (2017). *Disrupting Mobility – Impacts of Sharing Economy and Innovative Transportation on Cities*. Springer International, Cham, Switzerland.
- Millard-Ball, A., Murray, G., Ter Schure, J., Fox, C., and Burkhardt, J. (2005). *Car-Sharing: Where and How It Succeeds*. The National Academies Press, Washington, DC.



- Moussaïd, M., Helbing, D., and Theraulaz, G. (2011). How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108(17):6884–6888.
- Musselwhite, C. and Haddad, H. (2010). Mobility, accessibility and quality of later life. *Quality in Ageing and Older Adults*, 11(1):25–37.
- National Transport Development Policy Committee (2012). Working group on urban transport – final report. Report, Ministry of Urban Development, Government of India.
- Navidi, Z., Nagel, K., and Winter, S. (2019). Toward identifying the critical mass in spatial two-sided markets. *Environment and Planning B: Urban Analytics and City Science*, online first.
- Pafka, E. (2017). Integration is not walkability: The limits of axial topological analysis at neighbourhood scale. In *11th Space Syntax Symposium*, pages 102.1 – 102.10. Instituto Superior Técnico, Departamento de Engenharia Civil, Arquitetura e Georrecursos, Portugal.
- Pandey, V., Monteil, J., Gambella, C., and Simonetto, A. (2019). On the needs for maas platforms to handle competition in ridesharing mobility. *Transportation Research Part C: Emerging Technologies*, 108:269–288.
- Peiravian, F., Derrible, S., and Ijaz, F. (2014). Development and application of the pedestrian environment index (pei). *Journal of Transport Geography*, 39:73–84.
- Pucher, J., Buehler, R., Bassett, D. R., and Dannenberg, A. L. (2010). Walking and cycling to health: A comparative analysis of city, state, and international data. *American Journal of Public Health*, 100(10):1986–1992.
- Richter, K.-F., Winter, S., and Santosa, S. (2011). Hierarchical representation of indoor spaces. *Environment and Planning B*, 38(6):1052–1070.
- Rigby, M. and Winter, S. (2016). Usability of an opportunistic interface concept for ad hoc ridesharing. *International Journal of Cartography*, 2016(2):115–147.
- Rigby, M., Winter, S., and Krüger, A. (2016). A continuous representation of ad hoc ridesharing potential. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2832–2842.
- Rüetschi, U.-J. and Timpf, S. (2005). Modelling wayfinding in public transport: Network space and scene space. In Freksa, C., Knauff, M., Krieg-Brückner, B., Nebel, B., and Barkowsky, T., editors, *Spatial Cognition IV: Reasoning, Action, and Interaction*, volume 3343 of *Lecture Notes in Artificial Intelligence*, pages 24–41. Springer, Berlin.

- Schaap, J., Zlatanova, S., and Oosterom, P. (2011). Towards a 3d geo-data model to support pedestrian routing in multimodal public transport travel advices. In Zlatanova, S., Ledoux, H., Fendel, E. M., and Rumor, M., editors, *Urban and Regional Data Management*, pages 59–76. CRC Press, London.
- Seamon, D. (1980). Body-subject, time-space routines, and place-balletts. In Buttimer, A. and Seamon, D., editors, *The Human Experience of Space and Place*, pages 148–165. Croom Helm Ltd, London.
- Shaheen, S. and Chan, N. (2016). Mobility and the sharing economy: Potential to facilitate the first- and last-mile public transit connections. *Built Environment*, 42(4):573–588.
- Shaheen, S. A. and Cohen, A. P. (2007). Growth in worldwide carsharing: An international comparison. *Transportation Research Record*, 1992(1):81–89.
- Shoup, D. (2005). *The High Cost of Free Parking*. Routledge, London.
- Speck, J. (2013). *Walkable City: How Downtown Can Save America, One Step at a Time*. Farrar, Straus and Giroux, New York.
- Stahl, C. and Schwartz, T. (2010). Modeling and simulating assistive environments in 3-d with the yamamoto toolkit. In Mautz, R., Kunz, M., and Ingersand, H., editors, *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE.
- Stoffel, E.-P., Lorenz, B., and Ohlbach, H. J. (2007). Towards a semantic spatial model for pedestrian indoor navigation. In Hainaut, J.-L., Rundensteiner, E. A., Kirchberg, M., Bertolotto, M., Brochhausen, M., Chen, Y.-P. P., Cherfi, S. S.-S., Doerr, M., Han, H., Hartmann, S., Parsons, J., Poels, G., Rolland, C., Trujillo, J., Yu, E., and Zimányi, E., editors, *Advances in Conceptual Modeling – Foundations and Applications*, volume 4802 of *Lecture Notes in Computer Science*, pages 328–337. Springer, Berlin.
- Toh, C. K. (2001). *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall, Boston.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460. in reader cognitive science.
- van Soest, D., Tight, M. R., and Rogers, C. D. F. (2020). Exploring the distances people walk to access public transport. *Transport Reviews*, 40(2):160–182.
- Vanclooster, A., Van de Weghe, N., and De Maeyer, P. (2016). Integrating indoor and outdoor spaces for pedestrian navigation guidance: A review. *Transactions in GIS*, 20(4):491–525.

- Walton, L. A. and Worboys, M. (2012). A qualitative bigraph model for indoor space. In Xiao, N., Kwan, M. P., Goodchild, M. F., and Shekhar, S., editors, *Geographic Information Science*, Lecture Notes in Computer Science, pages 226–240. Springer, Berlin.
- Wang, Z., Novack, T., Yan, Y., and Zipf, A. (2020). Quiet route planning for pedestrians in traffic noise polluted environments. *IEEE Transactions on Intelligent Transportation Systems*, (early access):1–12.
- Winter, S. and Wu, Y. (2009). Intelligent spatial communication. In Navratil, G., editor, *Research Trends in Geographic Information Science*, pages 235–250. Springer, Berlin.
- Winters, M., Teschke, K., Brauer, M., and Fuller, D. (2016). Bike score: Associations between urban bikeability and cycling behavior in 24 cities. *International Journal of Behavioral Nutrition and Physical Activity*, 13(1):Paper 18.
- Wunderlich, D. and Reinelt, R. (1982). How to get there from here. In Jarvella, R. J. and Klein, W., editors, *Speech, Place, and Action*, pages 183–201. John Wiley & Sons, Chichester.
- Yang, L. and Worboys, M. (2015). Generation of navigation graphs for indoor space. *International Journal of Geographical Information Science*, 29(10):1737–1756.
- Young, M. and Farber, S. (2019). The who, why, and when of uber and other ride-hailing trips: An examination of a large sample household travel survey. *Transportation Research Part A: Policy and Practice*, 119(1):383–392.
- Zhao, F. and Guibas, L. J. (2004). *Wireless Sensor Networks*. Elsevier, Amsterdam.
- Zhu, L., Yu, F. R., Wang, Y., Ning, B., and Tang, T. (2019). Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):383–398.

## **Parking as a Challenge for Urban Mobility**

## 7 Parking as a Challenge for Urban Mobility: Introduction

STEPHAN WINTER AND SALIL GOEL

### Abstract

This part of the book collects smart city approaches to support parking, mostly focusing on the parking pressure in inner-urban areas. The presented recipes for parking information and management rely on a *smart* – sensor-infused, connected, digitally enhanced – urban parking infrastructure that incorporates and utilizes the smart geospatial technologies presented in the first part. It also complements the approaches presented at the end of the first part, which focused on avoiding and shifting private motorized trips in cities, and thus alleviated parking pressure. The approaches presented also weigh also their options when confronted with traffic on roads with less infrastructure and less discipline.

### Keywords

Parking, mobility, traffic

Parking is a necessity. A vehicle is a means to an end, and once the end has been accomplished – the travel destination reached – the vehicle can be disposed of, or *parked*, for reuse. This means, parking is intrinsically tied to urban mobility. All forms of urban mobility require some form of parking: self-directed vehicles, shared vehicles (e.g., Figure 7.1), and even autonomous vehicles – at some times and at some places. In this book, however, we focus on the parking of self-directed vehicles, and these vehicles need to be parked close to the destination of the trip.

In principle, the individuals participating in urban mobility – satisfying their need for some access – have a choice of how to travel, when to travel, and, if choosing to travel in a self-directed vehicle, where to park. Thus, since parking is intrinsically tied to mobility, parking can be addressed from three angles:

1. By avoiding trips, or reducing the need to travel. This first angle is mostly in the minds of urban planners thinking about densification. If work, services, and goods are offered at shorter distances – and thus, outside the city center – travel demand decreases as a consequence.

2. By shifting trips to other modes. This second angle relies on choice, and then on making the alternatives attractive by some tangible or intangible incentives. Ways to use data and information services to shift mobility demand away from the use of private vehicles have been discussed in the first part of this book, or are often subsumed as mobility as a service, or *MaaS* (Exposito-Izquierdo et al., 2017). This includes the disruption of urban mobility (Meyer and Shaheen, 2017) driven by three technologies: the shared (platform) economy, autonomous driving, and electrification of power trains. Each of these has an impact on parking demand and parking space management.
3. By building the smart infrastructure to better manage the existing parking supply or parking demand. Managing parking supply includes, for example, informing drivers of the occupancy status of parking spaces around, while managing parking demand includes, for example, dynamically influencing the drivers' behavior or choice depending on prior information. This third angle will be the topic of this part of the book.



**Figure 7.1:** Tram depot in Adelaide, South Australia. Source: <https://bit.ly/3d24m54> (© SCHolar44, 2018, original by © Adelaide h class, 2009, public domain).

Smart infrastructure, a loose term used here for the overlap of data (sensors), information, and communication technologies (ICT), and the Internet of Things (IoT), facilitates urban parking in two ways: informing any decision making and collaboration:

1. Drivers of self-directed vehicles have a limited information horizon. Using the terminology of Montello (Montello, 1993), we can distinguish between the space that these drivers immediately perceive – their *vista space* – and

the space that their current travel can take them to – their *environmental space*. Drivers make decisions on their search for parking based on their perceptions in their current vista space and their experience or their estimate about the state in their environmental space. Smart infrastructure, however, can widen the information horizon and provide *advanced parking information*, for example, by replacing experience or estimates with real-time information on parking space occupancy, or simply with mapped data of available parking spaces.

2. Choosing a parking space is currently an individual decision, at the most, informed by the parking guidance systems described above. These individual decisions are made out of self-interest, i.e., each individual driver is trying to optimize his/her total cost of parking, which is influenced by many factors, such as parking fees, search time, or distance to the trip destination (walking time). For one, drivers are not completely rational about their choices, biased by what psychologists call the prospect theory (Kahneman and Tversky, 1979), a behavior that is influenced by risk avoidance. But the sum of all individual interests is not necessarily the global optimum for a system: If each driver is aiming for the next parking space, for example, some may lose out and end up with very long search times. *Advanced parking management* interferes with individual strategies and interests by incentivizing behavior that will lead to parking closer to the global optimum. It includes some form of collaboration.

The very first section in this part, however, will look at the nature of parking in global cities.

These approaches also take into account the traffic conditions in countries where private motorization has not yet reached saturation levels, for example, Indian cities. We have seen before (“parking is not a right, but a privilege” (National Transport Development Policy Committee, 2012)) that parking is a private use of public resources, where the common good includes land opportunity costs, capital costs, and operation and maintenance costs. If the public provides space for parking, this provision cannot grow with demand because space is limited – the dilemma of the commons (Hardin, 1968). Countries where private motorization has not yet reached saturation levels experience this dilemma twice as hard, since urbanization increases the pressure on mobility in the city, and rapid growth of motorization levels only make things worse.

Cities in these countries often lack the infrastructure to cope with parking. But where on-road parking is unregulated or free, induced parking demand is high. For Indian cities, for example, with exponential growth of motorized mobility and from levels that are far from saturated (National Transport Development Policy Committee, 2014), the typical consequence is haphazard parking<sup>1</sup> To stem the

<sup>1</sup><https://bit.ly/3ctGBUi> – Smart Cities Council India, 2018

demand, the Indian National Transport Development Policy Committee made a number of recommendations that are based on a strategy of *avoid, shift, improve* (National Transport Development Policy Committee, 2012) – *avoid* increased demands for mobility by reducing the number of trips as well as also reducing the length of trips, *shift* trips to more sustainable modes, and *improve* the infrastructure. This strategy's main recognized management tool is the pricing of parking. The Committee states (National Transport Development Policy Committee, 2014, p. 425): "Land is valuable in all urban areas. Parking places occupy a large part of such land. This should be recognized in determining the principle of allocating parking space. Levy high parking fees that represent the value of the land occupied."

Of course mechanisms are available that reduce urban parking pressure by advertising or incentivizing the use of other modes of traveling, such as park-and-ride, or ridesharing, or discouraging the use of the private car, such as by levying a congestion tax. These mechanisms have been discussed before. In this part, we focus on smart tools to monitor parking pressure, generate information about parking pressure, and track the use of this information by the drivers of private vehicles. A particular focus is put on the cruising time in search for (affordable) parking space, which is known to contribute significantly to inner city (peak hour) traffic – with large variations in numbers reported in the literature. Cruising for parking and parking itself affects land use, air quality, traffic congestion, travel behavior, people's safety, people's moods, people's disposable times, and the economic development of a city.

Most of the solutions currently suggested for smart cities are developed and tested in countries with private motorization at saturation, and are not directly transferable to Indian conditions. Unlike in western countries, reserved parking space is scarce in India; instead, parking is often uncontrolled. The reasons for this are manifold and include a reluctance to pay for parking space, inadequate monitoring mechanisms to detect unauthorized or illegal parking, and accordingly, Indian drivers being less accustomed to following traffic rules. For the same reasons, the current parking infrastructure in India cannot sustain costly sensors. The wide variety in types of vehicles on India's roads adds to the problem of uncontrolled parking.

## Bibliography

Exposito-Izquierdo, C., Exposito-Marquez, A., and Brito-Santana, J. (2017). Mobility as a service. In Song, H., Srinivasan, R., Sookoor, T., and Jeschke, S., editors, *Smart Cities: Foundations, Principles, and Applications*, pages 409–436. John Wiley & Sons, Inc., Hoboken, NJ.

Hardin, G. (1968). The tragedy of the commons. *Science*, 162(3859):1243.



- Kahneman, D. and Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291.
- Meyer, G. and Shaheen, S. A. (2017). *Disrupting Mobility – Impacts of Sharing Economy and Innovative Transportation on Cities*. Springer International, Cham, Switzerland.
- Montello, D. R. (1993). Scale and multiple psychologies of space. In Frank, A. U. and Campari, I., editors, *Spatial Information Theory*, volume 716 of *Lecture Notes in Computer Science*, pages 312–321. Springer, Berlin.
- National Transport Development Policy Committee (2012). Working group on urban transport – final report. Report, Ministry of Urban Development, Government of India.
- National Transport Development Policy Committee (2014). *Urban Transport*, book section 5, pages 380–476. Routledge, New Delhi.

## 8 The Nature of Urban Parking

STEPHAN WINTER

### Abstract

Parking is a result of derived and often induced demand, and thus suitable for active management by tools of data and information services that influence demand and search behavior. But what is the subject of this management? This is the topic of the current chapter.

### Keywords

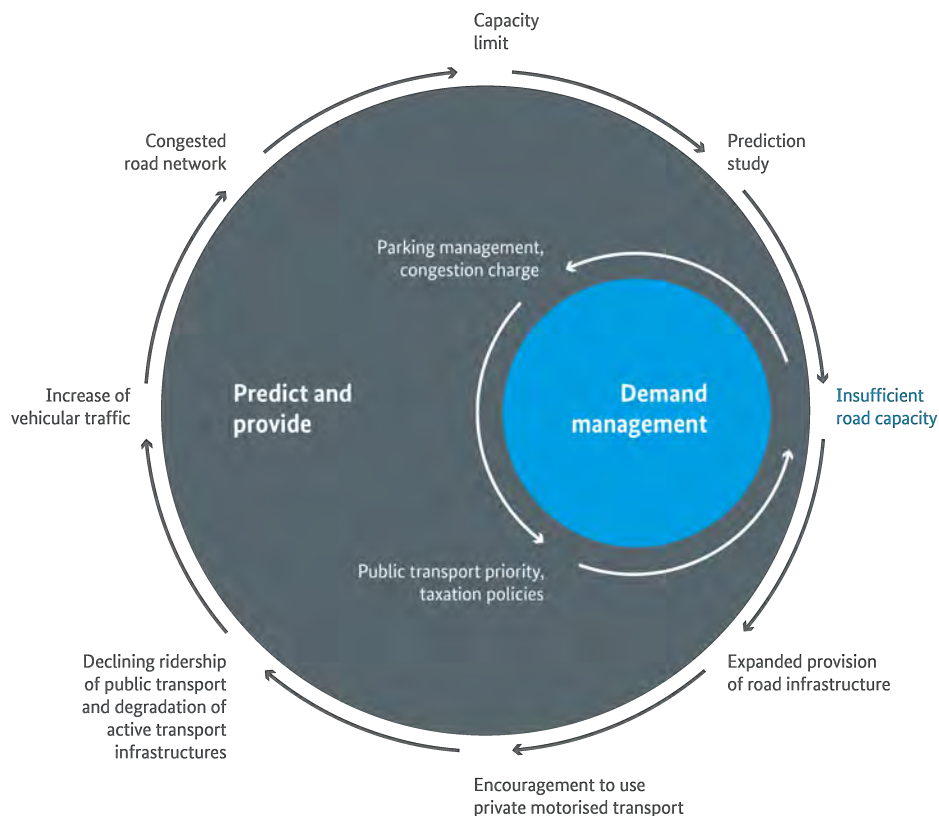
Parking, cruising, occupancy

### 8.1 Demand for Parking

As we said in the introduction to this part, parking is a necessity. A vehicle is a means to an end, and once the end has been accomplished – the travel destination reached – it can be disposed of, or *parked*, for reuse. Economists call such dependency a *derived demand*: a demand for a service in one sector (here: temporarily occupying some real estate) occurring as a result of demand from another sector (here: movement). In these terms parking is a second derivative, since movement of people or goods has already been considered a derived demand: Transport facilitates access to satisfy other individual needs (such as work, education, or recreation) or other economic needs (such as delivery of goods). Such stationary demands as parking, derived from the derived demand of mobility is also called an *indirect derived demand* (Rodrigue et al., 2017).

This dependency of parking from mobility makes the management of parking a harder problem: any interference with demand or supply for parking has implications also for mobility, and vice versa. Some of the effects of interference with this complex system between mobility and parking can even be counter-intuitive. Most prominently, easing parking in the city center, e.g., by means of smart parking management, will most likely attract more demand for parking: this step *induces* demand. Induced demand describes an increase in the demand of a good after supply increases, and is well known in transportation studies (Goodwin, 1996). If more roads are provided, more people tend to use

their car because space has become cheaper. If more cheap on-street parking spaces are made available, or the currently non-occupied on-street parking spaces can be found easily, more people choose the convenience of their own car instead of public transport. Induced demand can easily outweigh the benefits of an intervention, leading to the well-known paradox originally postulated by Braess (Braess, 1969). This paradox states that traffic, after any intervention, always returns to an equilibrium. In practice this means that all forms of cities experience pressure on road space and parking space, independent of their current offering – and the cities vary largely in their car (parking) friendliness. This cycle between further investment, e.g., in parking management, and then returning to operating at capacity limits is illustrated in Figure 8.1.



**Figure 8.1:** The cycle that induces demand. Source: <https://bit.ly/2NIL1gV> (© Transformative Urban Mobility Initiative (TUMI), 2019, CC BY-SA 4.0, modified).

While investment in smart parking induces demand, there is not much point in considering whether parking supply created the demand or whether parking demand created the supply. A critical factor for specifying this demand is the level of motorization in a city, and the mix of vehicles used. While parking was already an issue with horse-carts in earlier times, it is still an issue in the cities of our

times where person-directed vehicles – i.e., vehicles with parking requirements – cater to a large range, including private cars, motorbikes, bicycles, scooters, three-wheelers, and, at least in some parts of the world, animals and tour buses. That is, demand for parking is not limited to private cars only and includes other vehicles above. An answer to the problem of parking pressure in high-income countries, with their saturation of private car ownership at about 0.5 cars per capita, is the increasingly active management of parking supply. This approach is using technology for efficiency gains, which is seen as sufficient. The bigger challenge is probably the dramatically growing private car ownership in other countries, where more efficient management of the existing parking supply is not sufficient. While twenty years ago the car ownership rate in countries with lower economic standards of living was at 0.06 (60 cars per 1000 persons) (Ingram and Liu, 1999), these rates are now closer to 0.2 for cities such as Delhi and Shanghai (Trouve et al., 2018), and even 0.3 for Jakarta (Kresnanto, 2019).

Obviously, parking requirements depend on the type and size of the vehicles (a similar consideration has often been made for street space). Bicycles' parking space is a fraction of a car's (Figure 8.2). An on-street parking bay for a private car can cater up to ten bicycles, which gives a strong motivation for cities to repurpose car parking space.



**Figure 8.2:** Bicycle parking footprint. Source: <https://bit.ly/3gnWNr4> (© HensleyStudios, free use, 2021).

A more subtle point has to be made about the size(s) of private cars. US American cities, which have grown largely with the private car around, i.e., during the twentieth century, have also grown with, on the average, larger cars, and thus have not only more, but also larger parking spaces than elsewhere. The older European or Asian cities had to adapt to the private car, and have generally smaller parking spaces for more compact cars. The average on-street standard parking space is about 14 square meters – less in Europe, and more in the United States of America. Residential parking spaces vary also in size, determined by local building regulations, but are generally smaller than public on-street parking spaces.

But then cars have become, on average, larger over time as well, while existing parking space cannot grow. For example, the building regulations in one European country set minimum requirements for a parking space on residential property to 4.8 m × 2.3 m – big enough for a Volkswagen Golf I (1974), which had a length of 3.72 m and a width of 1.63 m without mirrors. But the Volkswagen Golf VIII (2020) has already a length of 4.29 m and a width of 1.79 m without mirrors (Figure 8.3). As a consequence, getting into the vehicle in the same parking space has become more difficult. Luxury limousines of today now exceed 5 m in length, and do not fit on the smaller car parking spaces any longer.



**Figure 8.3:** Cars are getting bigger – parking spaces becoming smaller.

Another consideration concerns the location and the quality of the parking space required. For example, valet parking services relieve from the demand for parking space in the immediate vicinity of certain trip destinations without varying the size of parking demand itself (except by inducing demand). A taxi serves, on average, a similar number of passengers as a private vehicle and thus requires as much road space as a private vehicle for a similar service, but does not require a parking space near a requested destination. Similarly, future autonomously driving vehicles do not require a parking space nearby a requested destination, because they can offload their passengers and then look for a cheap parking opportunity elsewhere if they do not continue cruising empty – but the latter can be controlled by dissuasive market mechanisms to avoid the undesirable impact on road space consumption. Furthermore, since autonomously driving vehicles will be electrically powered they presumably also expect a parking infrastructure with charging stations. At least for the next few decades, such infrastructure will be supplied only for a subset of urban parking spaces. Finally, autonomously driving vehicles might be operating similarly to other carsharing systems, which means that they have higher occupancy rates than private cars and thus lower parking time needs compared to private cars. Simulations (with certain assumptions: that all autonomous vehicles are shared but no rides are shared) showed 70%-85% reduction of parking space demand (Kondor et al., 2018), a variability that depends on passenger waiting time tolerances. Similarly, current carshare systems already contribute to reduced private car ownership, and thus, may indirectly reduce required parking space. Stated preference sur-

veys indicated that up to 20 % of car drivers are willing to give up their car for carsharing (Liao et al., 2020).

According to all these considerations, today's parking demand has potential for change in the future without significant impact on mobility and access.

But a vehicle does not need only one parking space. It occupies different parking spaces at different times, for example, one at home, one close to work, and at the supermarket. Some of these parking spaces are time-shared, but others are reserved (for example, the private garage). Since parking spaces are not occupied all the time – for example, the private garage tends to be empty during daytime – but a vehicle is, on average, parked 95 % of all time (Shoup, 2005). A vehicle needs, on average, more than a single parking space.

It has also been estimated that over time the parking of a vehicle takes up twice as much space than driving this vehicle (Meyer et al., 1965). This is partly due to the fact that it is not just the parking space that is taken over. Accessing the parking space – searching for a parking space, and maneuvering into a parking space – take space as well. These maneuvers, because of their lower speed, send significant waves through road traffic (Lighthill and Whitham, 1955). In addition, the temporal signature of parking demand impacts these space costs. Short-term parking, requiring frequent maneuvering in and out a parking space, impacts traffic and street use more than long-term parking.

This means the searching for parking space has to be included in the costs of parking. These costs are hidden costs because they are difficult to detect. For example, the point in time when a driver switches from goal-directed driving to cruising for parking can hardly be identified, among other reasons because people show very different strategies when searching for parking space (Krapivsky and Redner, 2019). This challenge alone makes it hard to estimate the search time of drivers for a parking space (Polak and Axhausen, 1990; Shoup, 2006; Belloche, 2015). However, Shoup reports cruising times between 3.5 and 14 minutes to find an on-street parking space in urban centers, and he estimates that vehicles cruising for parking contribute between 8 % and 74 % of the inner-urban traffic (Shoup, 2006). This number is significant not only with regard to road space use and contributions to congestion, but also with regard to the passengers' times, fuel costs, and emissions produced – all factors that are frequently not considered when comparing modes of traveling or the impact of free or cheap on-street parking (Belloche, 2015).

## 8.2 Supply for Parking

Looking into parking demand is one side of the coin, and is, as we have seen, already complicated enough. The other side of the coin is looking at parking space supply.

It is reported that in US cities about 30%-60% of the ground surface is used for roads and parking, reserving for each car about two off-street and two on-street parking spaces (Rodrigue et al., 2017). These numbers are justified by demand: currently 95% of US households own a car, and 85% of Americans commute by car (US Department of State<sup>1</sup>). Motorization rates and thus, the demand for road space is lower elsewhere. Compared to an average of 30% of road surface in the car-reliant North American cities, Western European cities show 15%-20% use of road space, and cities in low and middle-income countries, about 10% (Rodrigue et al., 2017). The city-state of Singapore, which has smarter policies on car usage, better public transport, and only inner-city traffic, still has 2.1% of its precious surface reserved for parking spaces (Kondor et al., 2018). This way Singapore is deliberately keeping parking demand down.

Public on-street parking is only one response to the scarcity of inner-city parking spaces. The commercial sector adds parking spaces for one of two reasons. Where the scarcity in supply increases the willingness to pay, private infrastructure is provided for the direct benefit of parking fees. Alternatively, the private sector invests in parking spaces for indirect returns, for example by providing free parking at shopping centers. Private parking spaces are always provided off-street, in a large variety of forms: private parking spaces at homes, car parks, parking houses, or valet parking systems. The free or cheap public parking spaces are typically not managed, but in commercial parking spaces information technology is often deployed for guidance and coordination, and for optimizing space use. This optimization can include parking management strategies (Litman, 2008). Among the latter are pricing strategies, such that at times of high demand, higher fees should shift some of the demand to times when demand is low. Other competitive advantages can come from better service strategies that improve the usability of the dedicated parking space. For example, an improved design or increased capacity can improve a park-and-ride system and attract higher patronage. Similarly, intelligent guidance systems can improve the usability of a large parking house or a distributed parking infrastructure (Rizvi et al., 2019).

Drivers searching for parking – even if they tend to ignore some factors (see above) – will generally consider the costs involved in parking: the cruising, the parking fees, the time spent in managed parking facilities (cruising, walking, paying), and the travel time spent between parking space and trip destination. These costs for the individuals, some tangible (parking fees, consumption), some intangible (time, tear-and-wear, emissions), form a complex system for self-interest driven decision making on the choice of travel mode and parking choice. Neglected in this decision process (but not by public and private suppliers) are the costs involved in the provision of parking spaces, be it private investment in a garage, the provision of on-street parking spaces, or the commercial supply of

<sup>1</sup><https://bit.ly/31HKZSN> – US Government, 2010

managed parking. These costs are often recovered through internalizing externalities, for the public hand, for example by investing income from fuel taxes. Another common way for city councils to counter the tragedy of the commons is regulation, especially by:

- limiting parking periods;
- requesting parking permits for access of parking spaces;
- impacting demand by congestion taxes or other mechanisms to limit access to city areas;
- zoning codes requiring the private provision of parking spaces with construction and development.

### 8.3 Parking Space in the City

Parking is regulated by acts, rules and regulations set by countries, states, and cities. Accordingly the regulations vary largely across the globe. Parking is also at the intersection of a range of domains, such that related legislation can be found in domains such as road safety, housing, congestion levies, or land conservation. This variety is also an indicator that the act of parking can happen on public grounds, on private grounds, under the open sky, under trees or roofs, in parking houses, or garages. Obviously, any parking space must have access to a road, wherever located and however regulated. But not every space accessible from the road is permitted for parking.

When we use in this book the terminology of *on-street parking* and *off-street parking* then we distinguish not only the spatial configuration of a place used for parking a vehicle (i.e., on the street or off the street), but automatically also refer to the ownership of this place. Since road space is owned by the public, on-street parking is always happening on public grounds. Off-street parking, in contrast, can happen on public or on private grounds. On private grounds the owner sets access regulations, traffic regulations, and parking regulations.

Parking is typically restricted. Some of the restrictions are applicable generally (and thus, they are not signed), and others are applicable locally (and thus, signed). Generally applicable restrictions concern, for example, minimum distances from intersections or driveways, or allowing adequate space for vehicles to pass, or the prohibition of double parking. Locally applicable parking restrictions concern, for example, time limits, parking fees, and places – if parking bays are marked, parking is only allowed within the marks.

The definition of a parking space is relatively broad, though. A parking space is any space that is either currently *used* for parking a vehicle, or a space *set aside* for parking a vehicle. Correspondingly, a parking space does not need



to be marked out, and in many cases is not. Parking at the street curb, for example, is happening often on shared road space. Where a space is set aside for parking and has no markings, some standard size of a parking space (in the local regulations) determines the capacity of the parking space. In Victoria, this standard size is 25.2 sq/m.<sup>2</sup>

### 8.3.1 On-street Parking

On-street parking is a private use of public resources. The resource is provided either for free or for comparatively cheap parking fees, with its own challenges of inducing traffic (Shoup, 2005). The establishment and maintenance costs of on street parking space include land opportunity costs, capital costs, and operation and maintenance costs. Since these costs are ‘common for all’, public parking is susceptible to what economists call the tragedy of the commons (Hardin, 1968), referring to a behavior of individuals that depletes common resources by self-interest. In contrast to off-street parking, the (public) on-street parking cannot grow arbitrarily with demand because the road space in cities is limited. “[On-street] parking is not a right, but a privilege” (National Transport Development Policy Committee, 2012).

The prototypical form of a parking space is the marked out one. Marked out parking bays on streets comes typically in one of three forms: bays parallel to a curb, bays perpendicular to the curb – which produces more spaces per street length – or bays at an angle to the curb – which is easier to park into, i.e., requires less street space for maneuvering, but also allows narrower aisles. Only these individually marked parking bays allow for parking management: used for fees, tracked by parking sensors, and counted in parking guidance systems. The marking requires some standardized size of the space reserved for one vehicle, which depends on local regulations. In Germany, for example, the bays for parallel on-street parking (Figure 8.4) are 2 m wide and 5.70 m-6.70 m long, and the bays for angle on-street parking are wider (2.50 m) and shorter (5 m). However, regulations are lagging behind the actual size of vehicles, which is growing.



**Figure 8.4:** A vehicle parking in a marked on-street parking bay.

Space set out for parking can also be marked by a separating line between traffic and parking space (Figure 8.5), or may be taken in a more opportunistic way from shared road space (Figure 8.6).

In the latter two instances the number of available parking spaces remains

<sup>2</sup><https://www.legislation.vic.gov.au/in-force/acts/congestion-levy-act-2005/017>

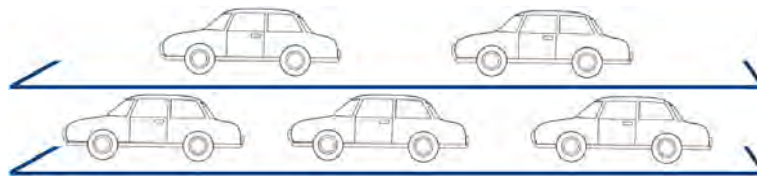


**Figure 8.5:** A vehicle parking on a marked on-street parking strip.



**Figure 8.6:** A vehicle parking at the curbside of the road, on shared road space.

undefined. The above mentioned standard sizes help to determine a theoretical capacity of a parking space, but this capacity may not be reached depending on where the first vehicles park and how that constrains the remaining spaces (Figure 8.7).



**Figure 8.7:** The number of vehicles that can be parked in unmarked parking spaces depends on other factors.

With our broad definition of a parking space – a space that is currently used for parking, or a space set aside for parking – the notion especially of the unmarked parking spaces becomes vague: A currently used space is not necessarily also set aside for parking. In countries with high parking pressure informal and illegal parking ‘convert’ street space or sidewalks temporarily into parking places. To what extent this behavior can be used to legalize and dedicate parking space where demand is high has been shown as well (My Thanh and Friedrich, 2017).

### 8.3.2 Off-street Parking

Large outdoor parking spaces, common at shopping malls for example (Figure 8.8), or at large business premises, are typically marked out in bays but rarely managed. They are also rarely the solution to inner-city parking pressures, where both, high demand and high real-estate prices, justify the erection of parking garages (Figure 8.9) or of automated vertical parking systems.

### 8.3.3 Off-street Parking

Large outdoor parking spaces, common at shopping malls for example (Figure 8.8), or at large business premises, are typically marked out in bays but



**Figure 8.8:** A private parking space at a shopping mall. Source: <https://bit.ly/31WtRsZ> (© Benh Lieu Song 2019, CC BY-SA 2.0, modified).

rarely managed. They are also rarely the solution to inner-city parking pressures, where both, high demand and high real-estate prices, justify the erection of parking garages (Figure 8.9) or of automated vertical parking systems.



**Figure 8.9:** A private parking garage. Source: <https://bit.ly/3rfWMZX> (© Rachmaninoff 2016, CC BY-SA 4.0, modified).

Further private off-street parking concerns smaller parking lots of companies, and the parking in driveways or garages. Public off-street parking is bound to road-related spaces that are publicly accessible and dedicated to parking.

## Bibliography

- Belloche, S. (2015). On-street parking search time modelling and validation with survey-based data. *Transportation Research Procedia*, 6:313–324.
- Braess, D. (1969). Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258–268.
- Goodwin, P. B. (1996). Empirical evidence on induced traffic. *Transportation*, 23(1):35–54.
- Hardin, G. (1968). The tragedy of the commons. *Science*, 162(3859):1243.
- Ingram, G. K. and Liu, Z. (1999). Determinants of motorization and road provision. Policy research working paper 2042, World Bank.
- Kondor, D., Zhang, H., Tachet, R., Santi, P., and Ratti, C. (2018). Estimating savings in parking demand using shared vehicles for home-work commuting. *IEEE Transactions on Intelligent Transportation Systems*, 20(8):2903–2912.
- Krapivsky, P. L. and Redner, S. (2019). Simple parking strategies. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(9):093404:1–17.
- Kresnanto, N. C. (2019). Model of relationship between car ownership growth and economic growth in java. In *IOP First International Conference of Construction, Infrastructure, and Materials*, volume 650, page Paper 012047. IOP Publishing.
- Liao, F., Molin, E., Timmermans, H., and van Wee, B. (2020). Carsharing: the impact of system characteristics on its potential to replace private car trips and reduce car ownership. *Transportation*, 47(2):935–970.
- Lighthill, M. J. and Whitham, G. B. (1955). On kinematic waves: II. A theory of traffic flow on long crowded roads. *The Royal Society: Proceedings A*, 229(1178):317–345.
- Litman, T. (2008). Parking management best practices. *ITE Journal on the Web*, 78(September):69–73.
- Meyer, J., Kain, J., and Wohl, M. (1965). *The Urban Transportation Problem*. Harvard University Press.
- My Thanh, T. T. and Friedrich, H. (2017). Legalizing the illegal parking, a solution for parking scarcity in developing countries. *Transportation Research Procedia*, 25:4950–4965.
- National Transport Development Policy Committee (2012). Working group on urban transport – final report. Report, Ministry of Urban Development, Government of India.

- Polak, J. W. and Axhausen, K. (1990). Parking search behaviour: a review of current research and future prospects. Report Working Paper 540, Transport Studies Unit, Oxford University.
- Rizvi, S. R., Zehra, S., and Olariu, S. (2019). Aspire: An agent-oriented smart parking recommendation system for smart cities. *IEEE Intelligent Transportation Systems Magazine*, 11(4):48–61.
- Rodrigue, J.-P., Comtois, C., and Slack, B. (2017). *The Geography of Transport Systems*. Routledge, New York, 4th edition.
- Shoup, D. (2005). *The High Cost of Free Parking*. Routledge, London.
- Shoup, D. C. (2006). Cruising for parking. *Transport Policy*, 13(6):479–486.
- Trouve, M., Lesteven, G., and Leurent, F. (2018). Private motorization in world-wide developing countries' metropolitan areas: Patterns in the early 21st century. In *PIARC International Seminar on Transport in the Fourth Revolution: The Dynamical Low-Income World*.

## 9 A Review of Smart Car Parking as IoT Systems

ALI ALIEDANI AND SENG W. LOKE

### Abstract

Car parking systems have been investigated extensively for minimizing the waste of time and traffic congestion due to vehicles cruising to park. Internet of Things (IoT) technologies such as sensing and networking have been utilized in car parking systems to provide connectivity of car parking components and to determine occupancy. Also, cooperation among vehicles for car parking based on vehicle-to-vehicle (V2V) communications has been investigated. However, interoperability, i.e., the ability for different systems to work together easily or to be used seamlessly, is one of the challenges for car parking systems, and for IoT applications in general. Car parking systems as part of city services can be owned by different stakeholders, which have different parking policies and use different car parking technologies. In this chapter, we outline conceptual architectures for parking systems, and highlight and discuss the challenges and approaches of interoperability in a range of smart car parking systems (implemented and proposed) in the literature.

### Keywords

Car parking systems, Internet of Things (IoT), interoperability

### 9.1 Introduction

Increasing urbanization calls for governments to undertake “smart city” innovations to effectively manage and utilize urban resources (Bélissent et al., 2010). The smart city is aided by adopting Internet of Things (IoT) technologies (Atzori et al., 2010), e.g., connected embedded sensors/devices to monitor and manage urban resources, such as transportation systems and healthcare. IoT is defined by Gubbi et al. (2013) as a capability to access information from interconnected ubiquitous sensing and actuation devices via an integrated framework.

Parking is considered one of the common challenges in urban areas. It is estimated based on an IBM global survey that 20 minutes is the average time per day to find a car park (Gallivan, 2011). Furthermore, vehicles cruising to find car parking can contribute to traffic congestion, which, in turn, increase fuel consumption and air pollution as studied by Shoup (2006). Numerous car parking systems have been investigated. Smart car parking is a smart city application

that facilitates locating parking spaces across the city, which can affect the time spent to arrive at a destination and the level of traffic congestion.

Moreover, given the different types of car parking systems, there is an issue of interoperability, which is the focus of this review. Unless online reservation is supported, information about parking spaces are generally siloed or available in a limited way (e.g., one has to come near to a parking center or an area to obtain partial availability information) so that wider scale (e.g., city-scale or suburb-scale) parking search, parking analysis, and parking coordination among vehicles, are not possible. Even a large shopping center with multiple car parks may not have integrated parking information for visitors, so that visitors have to manually navigate from one car park to another in search of spaces.

Indeed, the benefits of IoT for parking have been well articulated by Chandran et al. (2020). IoT smart parking solutions are being developed in some of the most populous countries of the world (e.g., Yadav and Prasad, 2019; Mu-faqih et al., 2020).<sup>1</sup> Smart parking systems have been reviewed extensively (Diaz Ogás et al., 2020).

This chapter aims to review car parking systems, in particular IoT based solutions, i.e., solutions involving sensing, and thing-to-thing communications (e.g., V2X, including V2V based solutions), with a focus on conceptual architectural approaches and interoperability issues, i.e., the ability for different systems to work together easily or to be used seamlessly.

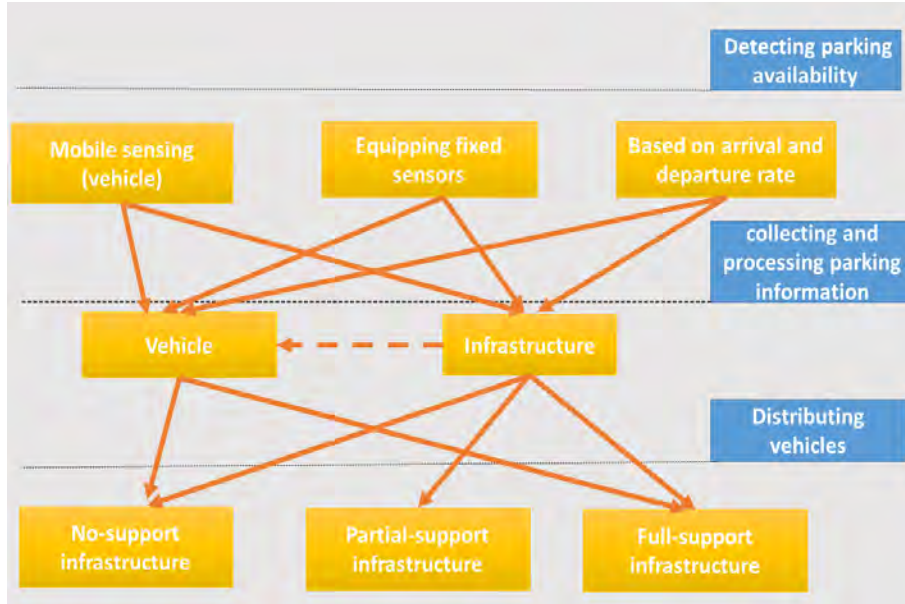
## 9.2 Car Parking Overview

Figure 9.1 describes a general architecture for car parking systems. There are three main stages: sensing parking, collecting and processing parking information, and distributing vehicles (e.g., via provided information, “nudges” or directed allocations) among the available parking spaces. The arrows represent the presence of data flows in a car parking system.

Firstly, the car parking space information is sent from car parking sensors, which are located in various geographic areas, to the database center, which can be centralized or distributed; parking information can be sent to the vehicles directly from sensors supported with wireless capability to the passing vehicles, or via local units which collect the parking information and broadcast the information to the vehicles (which is represented in the figure with the dashed arrow).

<sup>1</sup>See also the NB-IoT parking system in China (<https://bit.ly/3lCsgbu> – GSMA, 2018, or <https://bit.ly/3lDdXn4> – China Telecom, 2021), and viewpoints from Indonesia (<https://bit.ly/3vLgG2t> – Shankar Gautamon, 2019) and India (<https://on.tcs.com/2QkWbcv> – Tata Consultancy Services, 2019).

After processing the car parking information, the next stage is allocating or distributing cars to parking spaces, which can be based on a fixed car parking infrastructure or messages sent directly to the moving vehicles.



**Figure 9.1:** The conceptual architecture of a car parking system.

### 9.2.1 Sensing Parking Spaces

There are a variety of sensor technologies used to sense parking occupancy such as the inductive loop (Kianpisheh et al., 2012), RFID (Abdullah et al., 2013), ultrasonic sensing (Moguel et al., 2014), cameras (Bulan et al., 2013), and using magnetometers (Zoeter et al., 2014). The types of sensors used for smart car parking have been reviewed by Rupani and Doshi (2019) and Saleem et al. (2020). Chapter 10 in this book provides a comprehensive overview of methods for parking occupancy detection using sensors, and Chapter 11 discusses computer vision and deep learning techniques for parking space detection using CCTV cameras.

In addition, scanning parking space status (i.e., vacant/occupied) can be done by installing fixed sensors, which detect the parking occupancy of parking spaces within the sensor coverage, or mobile sensors, via sensors installed in vehicles to scan parking availability during vehicle cruising (Mathur et al., 2010).

Two approaches to determining car parking availability in car park lots have been considered. One, each car parking space is scanned by the equipped sen-



sors and the other way is via determining the number of arrival and departure vehicles via the car parking gate. Sensor information can be collected and processed by parking lot owners or by cruising vehicles.

The parking spaces can either be on-street parking, which are managed mainly by the council, or off-street parking, which can be owned by different private stakeholders, and can adopt various car parking technologies. For example, Hammoudi et al. (2018) present a car parking assistance system using cameras for detecting car parking spaces with machine learning for image recognition. The collected data is stored in a central database. The allocated car parking space for a vehicle is selected based on distance to the vehicle's final destination and the traveling distance from the current vehicle location to the parking space; an A\* algorithm is used to determine the shortest path.

Yan et al. (2011) use a vehicular network to inform the passing vehicles nearby the car park with a list of available car parking spots that can be detected and reserved by utilizing parking belts, using an infrared device for each parking spot. The system by Nawaz et al. (2013) exploits WiFi network signal characteristics to detect occupying and leaving parking spaces.

### **9.2.2 Processing Parking Information**

A data-driven approach together with a number of connected car parking sensors to collect massive car parking information can be used to enhance the performance of the car parking system.

Rajabioun and Ioannou (2015) study improving the accuracy of presenting car parking information in a parking guidance and information (PGI) system across a city via a prediction algorithm based on the multivariate autoregressive model. Adewumi et al. (2014) use a particle swarm pattern searching algorithm, based on collected historical car parking data at a university in order to optimize allocation of car parking spaces. Nguyen et al. (2018) propose a car parking paradigm based on cloud computing that can collect big car parking data. The authors adopted Hadoop MapReduce in implementing an IoT car parking system that covers a sizable geographic area. The proposed car parking system is organized in layers to reduce the transmission and processing of information at the cloud server via clustering and processing at the fog-computing level.

### **9.2.3 Distributing Vehicles to Parking Spaces**

In the process of distributing/allocating vehicles to parking spaces, there are three possible roles of the car park infrastructure which have been investigated: full support, partial support, and no-support.

- In *full-support infrastructure car parking*, the clients' role is restricted to

sending parking requests and receiving the parking reservation tickets. The car parking infrastructure is responsible for selecting parking spaces.

- In *partial-support infrastructure car parking*, the car parking infrastructure provides the searching vehicles with supervised information that can affect the searching pattern. Such information includes providing vehicles with parking availability information, including variation in parking price policies. For example, suppose a car park area has few vacant parking spaces and a large number of vehicles inside are looking to park, then the car park infrastructure can provide the vehicles with parking offers with increasing parking prices to motivate some vehicles to search at alternative car park areas. It is then up to the vehicles to go to and find/select parking spaces.
- In the *no-support infrastructure car parking* perspective, the vehicles have to collect information and resolve the competition by themselves, so that the infrastructure has only a minor role or could even be completely absent.

### 9.3 Cloud and Agent-Based Architectures

In the previous section, we provided a general three-layered architecture for parking and outlined approaches in each of the layers. This section describes another dimension to the architecture of parking systems: centralized and decentralized schemes. In particular, we consider centralized car parking based cloud technologies, agent-based car parking systems and standardized vehicular cooperation.

#### 9.3.1 Centralized and Cloud Based Car Parking Technologies

In this approach, the car parking paradigm is presented as a unified system to facilitate car parking, sensing parking occupancy status, analyzing the collected information, and interfacing with users through visualizing the parking availability or reserving parking based on a predefined plan. Cloud technologies have been used in this approach to connect car parking components which can be distributed in different areas.

For example, Sewagudde et al. (2016) introduced a car parking system for the Helsinki area, which manages parking permits, estimates parking availability and uses a parking payment plan. It applies LoRa (which is an acronym for Long range, low power wireless) technology to handle interoperability at the network level. In the same context, Lanza et al. (2016) introduce a field trial of car parking which presents the parking occupancy of indoor and outdoor spaces, for the 'smart' city of Santander downtown.

Ji et al. (2014) describe a cloud-based car parking system which consists of sensor, communication and application layers, using OSGi (Open Services Gateway initiative)-based Web and Android mobile devices for a car park at a university campus. Mainetti et al. (2015) propose a car parking system that helps drivers to the nearest available parking spaces with a payment method for users. The car parking system utilized 6LoWPAN and RFID sensors and applied the Constrained Application Protocol (CAP) to provide interoperability to access the sensing data of different sensors.

In the cloud-based smart parking solution of Atif et al. (2016), parking service providers are proposed which play the role of a broker, where parking lots can be advertised on a shared cloud platform. This can provide a degree of sharing across different parking systems, provided they agree to participate, and focuses the points of search for car park seekers.

Filtering the collected sensors' data in the car parking guidance system with the objective to save power by reducing the amount of information sent is studied by Alturki and Reiff-Marganiec (2017). Also, Alsafery et al. (2018) use a data fusion and filtering approach for a mobile app based smart parking solution.

### **9.3.2 Agent-Based Car Parking Systems**

The main objective of this approach is to arrange agreements between clients and a number of car parking stakeholders. The car parking stakeholders are represented by local gateways responsible for managing their car parking spaces and communicating with the coordinator agent, which work as part of a middle-ware layer between car parking stakeholders and users.

In such an approach, in car park systems for cities, although the car parking operators have their own objectives, they are obligated to cooperate with the coordinator agent to provide an interface between different agents and promote reaching of, possibly automatically negotiated, agreements.

An example of a multi-agent system for parking is provided by Di Napoli et al. (2014), which deals with coordinating car parking in a city based on a manager agent. The manager agent negotiates with clients, which are represented by software agents installed in vehicles, and with car parking owners, via the Iterated Contract Net protocol.

Parkres<sup>2</sup> is a car parking system, based on the cloud computing paradigm, to facilitate car park reservations in a city for consumers. Consumers can register or log in to the system with a possibility of viewing nearer vacant car parking space, booking and paying, while considering traffic congestion levels. Parkres is based on an integration of IoT car parking ecosystems. It manages the variations among car parking technologies by adopting a gateway that can communicate

---

<sup>2</sup>At <https://www.parkres.org>.

with local car parking systems and read different sensor data formats.

Jin et al. (2012) introduce a car parking system based on finding matches between clients and car parking operators' preferences using proposals via a coordinator agent. The coordinator agent sends a client request to the nominated car-parking operators, which are nearer to the client's detected destination. The car parking operators can invite the client based on the traffic condition, and the client can accept or reject the invitations.

Pham et al. (2015) study car park selection for drivers considering parking availability in the car parks and the distance to drivers' destination from car parks.

Yang et al. (2009) propose a parking guidance system to assist vehicles in searching, navigation and price negotiation. The work implements the FIPA (foundation for intelligent physical agents) architecture using multi agent messaging for interoperability among system components.

Kubler et al. (2016) provide a proof of concept of a smart parking framework to manage car parking at a sporting event (the FIFA World Cup 2022), with consideration of the variety of car parking areas' ownership. The design of the smart parking system is based on the Open Data Format (O-DF) and the Open Messaging Interface (O-MI) standard.

The car parking guidance system proposed by Wu et al. (2014) offers people a list of car parking areas with annotated parking availability and parking costs.

The variation in parking price policies among car parking areas is investigated by Chou et al. (2008), proposing agent coordination to assist drivers in finding optimum parking by bargaining on parking fees and computing the shortest path to the parking area. In the same vein, an auction mechanism is adopted by Kokolaki et al. (2014) to coordinate car parking based on parking fees. The agent-based approaches can be centralized or decentralized but are distributed.

### **9.3.3 Standardized Vehicle Cooperation for Car Parking**

Connected vehicles is a promising technology that assists commuters by enabling adaptive vehicle routing based on traffic conditions, avoiding an incident or accident via notifications of unpredictable events, and helping to find parking via cooperation between vehicles and with car park infrastructure. For example, Aliedani and Loke (2018) study the benefits of autonomous vehicle cooperation to coordinate the drop-off problem without involving a centralized unit. Also, Aliedani and Loke (2019) evaluate the effect of cooperation for car parking, without help from infrastructure, of some vehicles on the searching time of non-cooperative vehicles (which can be a result of the inability to connect due to interoperability issues).

Datta et al. (2016) introduce an IoT framework to integrate with connected vehicles to deal with vehicle network characteristics of heterogeneity of devices (vehicles and sensors) and short-lived connections. The framework adopts edge computing, which means processing data near its source rather than sending to a distant central server. Also, the framework uses semantic web-based data formats to provide a uniform platform to describe and use data from different sources; also, Named Data Networking (NDN) is used for broadcasting data.

Vehicles need to interact with and potentially connect not just to other vehicles, but also motorcycles, bicycles, pedestrians, and other road-users, as well as with IoT services (including via Road-Side-Units), over Dedicated Short Range Networking (DSRC) or 5G-V2X networking. There is a requirement to use a unified approach or standards that enable vehicles from different automakers to communicate and utilize the received information.

Guerrero-Ibanez et al. (2015) summarize the requirements of connected vehicles:

- connected vehicle *technologies*, which concerns defining a platform for interoperating and developing new technologies,
- connected vehicle *applications*, which focus on developing applications that can use and process the received real-time information,
- connected vehicle *policy*, related to the institutional policies of the participants in local areas such departments of transportation, and carmakers.

Defining Internet of Vehicle standards has received a lot of attention and effort from international organizations such as the Institute of Electrical and Electronics Engineers (IEEE), Internet Engineering Task Force (IETF), and World Wide Web Consortium (Contreras-Castillo et al., 2018). The Society of Automotive Engineers (SAE) released a message set dictionary for standardizing messages exchanged in DSRC communications, such as intersection collision warnings, emergency vehicle alerts and vehicle status information can be shared.<sup>3</sup> There are also large European projects on cooperative-ITS including cooperative vehicles.<sup>4</sup> The European Telecommunication Standard Institute (ETSI) provided the EN 302 637-2 standard<sup>5</sup>, which defined Cooperative Awareness Messages (CAMs). The message set dictionary, however, is not specific to parking applications using V2V communications. More work is required to develop message set dictionaries for cooperative parking.

<sup>3</sup>At <https://www.sae.org/standardsdev/dsrc/> and in particular the message set dictionary SAE J2735 at [https://saemobilus.sae.org/content/j2735\\_200911](https://saemobilus.sae.org/content/j2735_200911)

<sup>4</sup>E.g., <http://c-mobile-project.eu/>

<sup>5</sup><https://bit.ly/3ras0BK> – ETSI, 2021

## 9.4 Discussion and Open Issues on Interoperability

Smart car parking systems, as any large-scale IoT systems, have their issues with interoperability, which is defined by IEEE (Geraci et al., 1991) as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged.” From this definition, there are two problems to be addressed in order to provide end-to-end parking services: connectivity among system components and using the collected information.

Four interoperability levels are defined by Noura et al. (2017) when designing IoT systems:

1. Device interoperability level concerns providing connection among devices utilizing various communication technologies. One common solution of this type of interoperability is adopting a gateway that support various communication standards such as Zigbee, Bluetooth, and Wi-Fi.
2. The network interoperability level concerns a mechanism for exchanging messages among devices, connecting via different networks and providing end to end connectivity. Cloud computing is one example of providing this interoperability level.
3. The syntactic interoperability level deals with the data format and data structure of the transferred data among devices, for instance, Web-based services support XML, and the JSON format for representing data.
4. The semantic interoperability level involves understanding shared data. For example, an ontology is a common approach, which can be a means for different parties to map between descriptions of common concepts.

The innovation of building an application that can manipulate data from multiple IoT systems (IoT silos) require the system’s ability to exchange both raw data (syntactic interoperability that requires defined common data format and encoding) and its meaning (semantic interoperability). In other words, it is required to set up an agreement among IoT systems about the exchange rules, that can be implemented either in building a uniform centralized system or designing a middleware layer across IoT systems.

In most work on car parking systems, the authors proposed their own car parking system as an isolated system, without considering the interoperability or cooperation with the surrounding car parking systems. From a vehicle perspective, generally, the vehicles on the street are manufactured by different companies and for various models, yielding heterogeneity in hardware and software components among vehicles, impacting on the value of vehicle applications in different aspects (safety, traffic management, and entertainments), especially with the absence of vehicular standardization. The U.S. Department of Transportation

(DoT) has sponsored a project on interoperability for commercial vehicles supported with dedicated short-range communication (DSRC) with regard to safety applications (LeBlanc and Belzowski, 2012). In the same vein, there are different car parking information formats and structures, which require interoperability across car parking ecosystems if large scale search and coordination of parking among vehicles is to happen.

All the approaches above, centralized (vehicle-to-infrastructure or vehicle-to-cloud based) or decentralized (i.e., V2V based), require interoperability at the networking and data layer, at least for messages to be exchanged and to be understood. Agent-level communication protocols and standardized message formats and protocols are needed for middleware level and application services level interoperability.

In centralized schemes, agreement on application service protocols will help integration of parking search and tracking services across different parking areas, perhaps managed by different companies; Web protocols provide a basic layer of interoperability but the data needs to be in a format understood by vehicles from different manufacturers.

In decentralized V2V schemes, agreement at the *level of message protocols and formats* is needed so that vehicles from different manufacturers can understand each other's messages; there is a greater degree of cooperation possible among vehicles, which calls for standardization of message formats and protocols at the *cooperation level* above the *networking level*. Interoperability is required for understanding sensor based messages (e.g., situation and status of vehicles and parking spaces), as well as for vehicular cooperation (e.g., if two or more vehicles are to negotiate about contended parking spaces). At the vehicle behavior level, for human-driven vehicles, once messages are received, they can be interpreted by the drivers who then control their vehicles accordingly. But with self-driving vehicles, the vehicle needs to interpret the messages and behave accordingly; there is a need for standards at the *behavioral level*, i.e., rules of conduct, e.g., that autonomous vehicles do not try to compete with each other aggressively on learning that certain parking areas are unoccupied. Then, at the *policy level*, car parking systems will need to be able to agree on policies for data sharing (given the potential competitive nature of different car parking companies), data integrity, and how vehicles are informed of parking availability.

Moreover, a mixed of centralized and decentralized multi-agent based approaches might be adopted by vehicles searching and negotiating for parking spaces; hence, in the absence of provider-level system interoperability, a possible approach to interoperability could be on the vehicle end, a market of vehicle apps, which aggregates information from other vehicles and multiple infrastructure parking systems, and translate the information into a common format for its own reasoning and manipulation (and perhaps also exchange such information with other vehicles).

In summary, referring back to Figure 9.1, interoperability can happen at all three layers of the architecture. For example, sensed information about parking spaces from different sensor systems can be shared for different systems to process. Processed (or summarized) information can be shared for different systems to act on. Decisions on allocations from different systems (or individual vehicle's intentions on where to park) can be shared, so that different systems or vehicles can coordinate among themselves on what to actually do.

Also, depending on the architecture and technology used, centralized cloud based or multiagent based, different mechanisms for interoperability are required, e.g., standardized RESTful APIs or data exchange formats for the former, and standardized agent-level protocols for the latter.

## 9.5 Conclusion

In this chapter, we discussed car parking systems with a focus on the interoperability issue. Car parking systems consist of connecting a number of IoT devices and ecosystems, which, if able to interoperate, could provide a city-scale solution for parking.

Various interoperability techniques are required to provide scalable car parking services, involving data semantics, interfacing with users, connectivity technologies and high-level cooperation and coordination of car parking among local car parking operators via a middleware layer. Brokerage among car parking providers can be considered, but requires cooperation, and third-party capture (e.g., via Low-Earth-Orbit satellites, drones or crowdsourcing) of available parking spaces can be implemented, but may be limited in the extent of services that can be provided (e.g., limited to availability checks, and no booking and payment services) as well as the timeliness of parking information received.

We have also noted the impact of vehicles' cooperation in reducing the time to park, while the inability of vehicles to cooperate can worsen a situation. Indeed, there is a need to define a unified parking solution to enable interoperability for parking, at multiple layers - at the network layer, at the device layer, at the high level data sharing level, and at the vehicle behavior-based cooperation level.

With automated vehicles, parking in some areas may be less of a priority, and drop-off zones then need to be considered. Analogous to search for parking spaces, searching for drop-off spaces in cities will be an upcoming challenge.

## Bibliography

Abdullah, S., Ismail, W., Halim, Z. A., and Zulkifli, C. Z. (2013). Integrating Zigbee-based mesh network with embedded passive and active RFID for



production management automation. In *RFID-Technologies and Applications (RFID-TA), 2013 IEEE International Conference on*, pages 1–6. IEEE.

- Adewumi, A. O., Joel, L. O., and Sawyerr, B. A. (2014). Campus parking space allocation distribution using pattern search and particle swarm pattern search algorithms. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, pages 196–203. IEEE.
- Aliedani, A. and Loke, S. W. (2018). Cooperative autonomous vehicles: An investigation of the drop-off problem. *IEEE Transactions on Intelligent Vehicles*, 3(3):310–316.
- Aliedani, A. and Loke, S. W. (2019). Cooperative car parking using vehicle-to-vehicle communication: An agent-based analysis. *Computers, Environment and Urban Systems*, 77:101256.
- Alsaferi, W., Alturki, B., Reiff-Marganiec, S., and Jambi, K. (2018). Smart car parking system solution for the internet of things in smart cities. In *2018 1st International Conference on Computer Applications Information Security (IC-CAIS)*, pages 1–5.
- Alturki, B. and Reiff-Marganiec, S. (2017). Towards an off-the-cloud iot data processing architecture via a smart car parking example. In *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing, ICC '17*, New York, NY, USA. Association for Computing Machinery.
- Atif, Y., Ding, J., and Jeusfeld, M. A. (2016). Internet of things approach to cloud-based smart car parking. *Procedia Computer Science*, 98:193–198.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- Bélissent, J. et al. (2010). Getting clever about smart cities: New opportunities require new business models. *Cambridge, Massachusetts, USA*.
- Bulan, O., Loce, R. P., Wu, W., Wang, Y. R., Bernal, E. A., and Fan, Z. (2013). Video-based real-time on-street parking occupancy detection system. *Journal of Electronic Imaging*, 22(4):041109.
- Chandran, M., Immanuel, S., Kumaresan, T., Suresh Kumar, R., and Vishnu Murthy, K. (2020). Impulsive parking systems for smart cities based on vehicle location and destination using internet of things. In Kanagachidambaresan, G. R., Maheswar, R., Manikandan, V., and Ramakrishnan, K., editors, *Internet of Things in Smart Technologies for Sustainable Urban Development*, pages 157–171. Springer International Publishing, Cham.

- Chou, S. Y., Lin, S. W., and Li, C. C. (2008). Dynamic parking negotiation and guidance using an agent-based platform. *Expert Systems with Applications*, 35(3):805–817.
- Contreras-Castillo, J., Zeadally, S., and Guerrero-Ibañez, J. A. (2018). Internet of vehicles: Architecture, protocols, and security. *IEEE Internet of Things Journal*, 5(5):3701–3709.
- Datta, S. K., Da Costa, R. P. F., Härrri, J., and Bonnet, C. (2016). Integrating connected vehicles in internet of things ecosystems: Challenges and solutions. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A*, pages 1–6. IEEE.
- Di Napoli, C., Di Nocera, D., and Rossi, S. (2014). Agent negotiation for different needs in smart parking allocation. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 98–109. Springer.
- Diaz Ogás, M. G., Fabregat, R., and Aciar, S. (2020). Survey of smart parking systems. *Applied Sciences*, 10(11).
- Gallivan, S. (2011). IBM global parking survey: Drivers share worldwide parking woes technical report. Technical report, Technical report, IBM.
- Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H., and Springsteel, F. (1991). *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. IEEE Press.
- Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.
- Guerrero-Ibanez, J. A., Zeadally, S., and Contreras-Castillo, J. (2015). Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wireless Communications*, 22(6):122–128.
- Hammoudi, K., Cabani, A., Melkemi, M., Benhabiles, H., and Windal, F. (2018). Towards a model of car parking assistance system using camera networks: Slot analysis and communication management. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1248–1255. IEEE.
- Ji, Z., Ganchev, I., O'Droma, M., Zhao, L., and Zhang, X. (2014). A cloud-based car parking middleware for iot-based smart cities: Design and implementation. *Sensors*, 14(12):22372–22393.

- Jin, C., Wang, L., Shu, L., Feng, Y., and Xu, X. (2012). A fairness-aware smart parking scheme aided by parking lots. In *Communications (ICC), 2012 IEEE International Conference on*, pages 2119–2123. IEEE.
- Kianpisheh, A., Mustafa, N., Limtrairut, P., and Keikhosrokiani, P. (2012). Smart parking system (SPS) architecture using ultrasonic detector. *International Journal of Software Engineering and Its Applications*, 6(3):55–58.
- Kokolaki, E., Karaliopoulos, M., and Stavrakakis, I. (2014). Trading public parking space. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*, pages 1–6. IEEE.
- Kubler, S., Robert, J., Hefnawy, A., Cherifi, C., Bouras, A., and Främling, K. (2016). Lot-based smart parking system for sporting event management. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 104–114. ACM.
- Lanza, J., Sánchez, L., Gutiérrez, V., Galache, J. A., Santana, J. R., Sotres, P., and Muñoz, L. (2016). Smart city services over a future internet platform based on internet of things and cloud: The smart parking case. *Energies*, 9(9):719.
- LeBlanc, D. and Belzowski, B. (2012). Interoperability issues for commercial vehicle safety applications. Technical report.
- Mainetti, L., Patrono, L., Stefanizzi, M. L., and Vergallo, R. (2015). A smart parking system based on iot protocols and emerging enabling technologies. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 764–769. IEEE.
- 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.
- Moguel, E., Preciado, M. Á., and Preciado, J. C. (2014). A smart parking campus: An example of integrating different parking sensing solutions into a single scalable system. *Smart Cities*, page 29.
- Mufaqih, M., Kaburuan, E., and Wang, G. (2020). Applying smart parking system with internet of things (iot) design. *IOP Conference Series: Materials Science and Engineering*, 725:012095.
- Nawaz, S., Efstratiou, C., and Mascolo, C. (2013). Parksense: A smartphone based sensing system for on-street parking. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 75–86. ACM.

- Nguyen, S., Salcic, Z., and Zhang, X. (2018). Big data processing in fog-smart parking case study. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCLOUD/SocialCom/SustainCom)*, pages 127–134. IEEE.
- Noura, M., Atiquzzaman, M., and Gaedke, M. (2017). Interoperability in internet of things infrastructure: Classification, challenges, and future work. In *International Conference on Internet of Things as a Service*, pages 11–18. Springer.
- Pham, T. N., Tsai, M.-F., Nguyen, D. B., Dow, C.-R., and Deng, D.-J. (2015). A cloud-based smart-parking system based on internet-of-things technologies. *IEEE Access*, 3:1581–1591.
- Rajabioun, T. and Ioannou, P. A. (2015). On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2913–2924.
- Rupani, S. and Doshi, N. (2019). A review of smart parking using internet of things (iot). *Procedia Computer Science*, 160:706 – 711.
- Saleem, A. A., Siddiqui, H. U. R., Shafique, R., Haider, A., and Ali, M. (2020). A review on smart iot based parking system. In Ghazali, R., Nawi, N. M., Deris, M. M., and Abawajy, J. H., editors, *Recent Advances on Soft Computing and Data Mining*, pages 264–273, Cham. Springer International Publishing.
- Sewagudde, C. et al. (2016). The design and implementation of a smart-parking system for helsinki area (masters thesis). <https://aaltodoc.aalto.fi/handle/123456789/21622?show=full>.
- Shoup, D. C. (2006). Cruising for parking. *Transport Policy*, 13(6):479–486.
- Wu, E. H. K., Sahoo, J., Liu, C. Y., Jin, M.-H., and Lin, S.-H. (2014). Agile urban parking recommendation service for intelligent vehicular guiding system. *IEEE Intelligent Transportation Systems Magazine*, 6(1):35–49.
- Yadav, A. and Prasad, L. (2019). Application of iot devices for smart car parking system. In *Proceedings of the International Conference on Advances in Electronics, Electrical & Computational Intelligence (ICAECE) 2019*. [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3575462](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3575462).
- Yan, G., Yang, W., Rawat, D. B., and Olariu, S. (2011). Smartparking: A secure and intelligent parking system. *IEEE Intelligent Transportation Systems Magazine*, 3(1):18–30.
- Yang, L., Rongguo, M., and Longfei, W. (2009). Intelligent parking negotiation based on agent technology. In *Information Engineering, 2009. ICIE'09. WASE International Conference on*, volume 2, pages 265–268. IEEE.

Zoeter, O., Dance, C., Clinchant, S., and Andreoli, J.-M. (2014). New algorithms for parking demand management and a city-scale deployment. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1819–1828. ACM.

# 10 Sensors for Parking Occupancy Detection

KOUROSH KHOSHELHAM

## Abstract

This chapter provides an overview of sensor technologies and methodologies for determining the occupancy of parking spaces. It covers a range of sensors including active and passive sensors that can be installed overhead, in or on the ground in both indoor and outdoor environments. The chapter also provides a comparison of sensors, and discusses considerations for sensor selection and open challenges in parking occupancy detection.

## Keywords

Magnetic, ultrasonic, infrared, radar, RFID, camera, visible light, inductive loop, piezoelectric

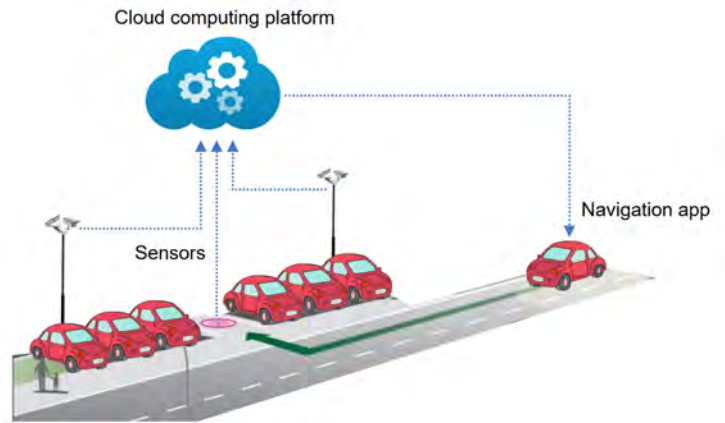
## 10.1 Introduction

The fast increasing urban population is presenting new challenges for the transport infrastructure in large cities. With more vehicles on the roads, parking spaces in busy city districts become scarce and harder to find. Consequently, drivers spend more time cruising for a parking space. This contributes to traffic congestion, increased fuel consumption, and increased carbon emissions. Frustrated drivers cruising for a parking space pose a risk to the safety of other road users, especially cyclists and pedestrians.

Many large cities across the world have realized the need for smart parking solutions that help the drivers find a parking space faster and more conveniently. Examples of smart parking solutions piloted in large cities include SFpark<sup>1</sup> in San Francisco, and Park and Joy<sup>2</sup> in Hamburg and several other cities in Germany. A main component of smart parking solutions are sensors for detecting the occupancy of parking spaces, both on and off street. This chapter provides an overview of various sensor technologies for parking occupancy detection. We first review the concept of smart parking, and then discuss parking occupancy detection sensors, with a focus on sensors that are installed in the environment

<sup>1</sup><https://bit.ly/399SLA6> – San Francisco Municipal Transportation Agency, May 2018

<sup>2</sup><https://bit.ly/3971NgW> – Smart City Hub, 23 March 2017



**Figure 10.1:** Smart parking concept. Source of background: <https://bit.ly/3varaXm> (© Deutsche Telekom, 2014).

(rather than on a mobile platform). Finally, we provide a comparison of different sensor technologies, and discuss considerations for sensor selection and open challenges in parking occupancy detection.

## 10.2 Smart Parking Concept

The basic concept of smart parking is illustrated in Figure 10.1. Sensors installed overhead, in or on the ground sense the occupancy of parking spaces and transmit the data in real time to a cloud computing platform via short range or long range communication devices. The cloud computing platform collects, stores and analyzes the data from all sensors and provides real-time parking availability and price information to mobile and web applications. Analysis of the data collected across different locations and at different times can provide other valuable information such as occupancy patterns, spatial-temporal variations, and correlation with events. The application component of the system enables the driver to find the nearest parking space on a map interface, book the parking space, and provides navigation guidance to reach it. In the following, we focus on the sensor component of smart parking systems.

## 10.3 Parking Occupancy Detection Sensors

A wide variety of parking sensors are available and have been used for parking occupancy detection. These sensors can be classified according to their installation platform. Mobile sensors are installed on a mobile platform such as a vehicle, or on a smartphone carried by the user. Fixed sensors are installed in the en-

vironment and serve as a sensing infrastructure. Fixed sensors can be further classified to overhead sensors, in-ground sensors and surface-mount sensors which can be glued to the surface. In this chapter, we focus mainly on fixed sensors that are installed in the environment. Mobile sensors are less practical for two main reasons. First, the data collected by a mobile sensor is more complex and the detection is more challenging. For example, in ParkNet (Mathur et al., 2010) where GPS and ultrasound sensors installed on the passenger side of taxi cabs were used to detect parked vehicles, generating an accurate parking occupancy map proved to be a challenge due to the complexity of determining the location and spatial extent of parking spaces. Second, parking occupancy detection using mobile sensors usually requires a crowd-sourcing approach which involves location sharing and raises privacy concerns. When parking availability information is crowd-sourced users may choose not to share their information (free riders) or deliberately disseminate false information to keep other drivers away from particular parking spaces for their own benefit (selfish liars) (Kokolaki et al., 2013).

To select the right type of parking sensor for a particular setting several criteria must be taken into consideration. Detection accuracy is perhaps the most important criterion for the selection of a parking sensor. It is defined as the proportion of correct detections in all detection results. Ideally, a parking sensor should be highly accurate, meaning that it always detects the occupancy or vacancy of a parking space correctly. Another important criterion for the selection of a parking sensor is reliability, which specifies how consistently the sensor performs under different environmental conditions. For example, a highly accurate sensor in normal conditions might completely fail in a rainy or snowy day or in a noisy environment, and is therefore rendered unreliable. In addition to accuracy and reliability, cost is another important factor when selecting a parking sensor. Sensors that can sense multiple parking spaces concurrently usually incur lower installation costs as fewer sensors will be needed to monitor a large parking lot. Sensors that are inexpensive to install might incur high maintenance costs. For example, sensors that wear out quickly, such as contact sensors, or sensors with a high power consumption, such as active sensors that emit sound or electromagnetic waves, are more expensive to maintain.

### 10.3.1 Magnetic Sensors

Magnetic sensors are passive sensors that measure the earth's magnetic field along three orthogonal axes. A magnetic sensor can detect the presence of a vehicle by measuring the distortion in the magnetic field caused by the vehicle. A simple detection algorithm based on magnetic measurements is to apply a threshold to changes of the magnetic field strength with respect to reference measurements made in a vacant parking space. In practice, however, the distortion of the magnetic field varies for different types of vehicles. Even for the



same vehicle the distortion in the magnetic field can vary from one point to another under the vehicle. To reduce detection errors caused by these variations, more advanced algorithms based on machine learning can be used. However, these algorithms are more computationally complex and require more data, higher sampling rates, and more computing resources, resulting in higher power consumption and faster battery drainage.

Magnetic sensors can be used in both indoor and outdoor parking spaces. In-ground and surface-mount installations are more common for magnetic sensors as these sensors have a limited range and can detect a vehicle only within a short distance. The main advantage of magnetic sensors are their low cost and low power consumption. Magnetic sensors in the market can be as cheap as \$1 per unit and typically have a battery life of up to 10 years<sup>3</sup>. The disadvantage of magnetic sensors is their short measurement range and susceptibility to other sources of magnetic interference. The short measurement range, typically 1 meter, means that vehicles with high clearance from the ground such as trucks, vans and SUVs might be difficult to detect. Magnetic interference can be caused by overhead power lines or other passing vehicles. Another important point to consider is that modern electric vehicles are made of lightweight material, such as carbon fiber and aluminum, which may be difficult, if at all possible, to detect by magnetic sensors.

Overall, magnetic sensors are moderately accurate but are less reliable due to their short range and susceptibility to magnetic interferences. On the positive side, they are inexpensive, consume little power, and have a long battery life, resulting in low installation and maintenance costs.

### **10.3.2 Ultrasonic Sensors**

Ultrasonic sensors are active sensors that use sound waves to measure distance to an object. An ultrasonic sensor emits ultra-high frequency (above 20 KHz) sound waves and detects the returned wave reflected off the surface of an object. By measuring the round-trip time, and assuming a constant velocity for the sound wave, the distance to the object is determined. An ultrasonic sensor can detect the presence of a vehicle by measuring the distance to the latter. A simple detection algorithm compares the measured distance to a reference distance representing a vacant parking space. A parked vehicle is detected if the absolute difference between the measured distance and the reference is larger than a distance threshold for a period of time longer than a time threshold.

Ultrasonic sensors are usually installed overhead and are more suitable for indoor parking lots because environmental conditions such as wind, rain, snow, and fog can influence ultrasonic distance measurements. Sound velocity also

---

<sup>3</sup><https://www.pnicorp.com/placepod/>

varies with humidity and air temperature resulting in inaccurate distance measurements. High frequency noise, e.g., generated by a whistle or the hissing of compressed air in pneumatic devices, and multipath effects where sound waves bounce off multiple surfaces, can also influence the performance of ultrasonic sensors. Acoustic sensors that use lower frequency sound waves (below 20 KHz) are more sensitive to ambient noise and are less common for parking occupancy detection.

The detection accuracy and reliability of ultrasonic sensors are generally considered high especially for sensors installed in indoor environments. Commercial products are claimed to achieve detection accuracies up to 99.9%<sup>4</sup>. However, ultrasonic sensors are relatively expensive with prices ranging from \$20 to \$100 per unit. They also have a moderate power consumption and require regular maintenance.

### 10.3.3 Infrared Ranging Sensors

Infrared ranging sensors use a similar sensing principle to ultrasonic sensors except they use infrared light instead of sound waves. The sensor emits pulses of infrared light and measures the returned light reflected off the object surface. Infrared ranging is based on either the intensity or the time of flight of the returned light. Intensity-based infrared ranging is sensitive to the reflectivity of the object surface and is therefore less reliable. Time-of-flight infrared sensors emit infrared laser light and measure the round trip time of flight of the returned light which is then converted to a distance measurement. Similar to ultrasonic sensors, detection of a parked vehicle is based on the comparison of the measured distance to a reference distance representing the absence of any vehicle.

Infrared ranging sensors can be installed overhead or on the ground<sup>5</sup>. However, they are generally prone to interference by ambient light and are therefore more suitable for indoor parking lots. Time-of-flight infrared sensors are less sensitive to ambient light, but broad daylight will still likely hamper the performance of an infrared sensor installed overhead in an outdoor setting. Environmental conditions such as rain and snow and obstruction by leaves or trash also influence the performance of infrared ranging especially for sensors installed on the ground.

Overall, infrared ranging sensors are considered moderately accurate but less reliable due to their sensitivity to environmental conditions. They are also relatively expensive, have a moderate power consumption and require regular maintenance.

---

<sup>4</sup><https://bit.ly/3vNN30m>

<sup>5</sup><https://www.nedapidentification.com/products/sensit/sensit-ir/>

#### 10.3.4 Radar Sensors

Radar, short for radio detection and ranging, is very similar to infrared ranging except it uses low frequency radio waves to make distance measurements. The radar sensor emits short pulses of low frequency radio waves, typically between 15-20 GHz, and detects the returned pulses reflected off the object surface. The distance measurement is based on the time of flight of the returned pulse. The presence of a vehicle is detected by comparing the measured distance with a reference distance representing the vacant parking space.

The main advantage of radar over infrared and ultrasonic ranging is that low frequency radio waves (corresponding to a wavelength of 1.5 to 2 cm) are not affected by small particles in the air. As such, radar sensors can operate in different weather conditions such as wind, rain, fog, humidity, and even light snow. This makes radar a reliable sensor for parking occupancy detection in both indoor and outdoor parking lots. Radar sensors can be installed overhead, in, or on the ground, although in-ground and surface-mount installations are more common<sup>6</sup>.

Radar sensors provide high accuracy and high reliability in parking occupancy detection. However, they are more expensive and consume more power as compared to infrared sensors. Therefore, installation and maintenance costs of radar sensors are relatively high.

#### 10.3.5 RFID Sensors

RFID, short for radio frequency identification, is a technology for the transmission of small packets of data using radio waves. It consists of a tag and a reader. For parking occupancy detection, the RFID tag is installed on the vehicle and stores information about the vehicle such as the make, model and registration details. When the vehicle is within the range of a reader installed in a parking space, the reader detects the tag, reads the data stored in it, identifies the vehicle, and determines whether it is parked in the parking space.

RFID readers can be installed in both indoor and outdoor parking spaces. Overhead installation is the common choice for RFID readers. The main disadvantage of RFID technology for parking occupancy detection is that it requires RFID tags installed on all vehicles. This is expensive and its implementation is logistically complex. An argument in favor of RFID sensors for smart parking systems is that in some cities many vehicles are already equipped with RFID tags for electronic toll collection (ETC)<sup>7</sup>, which can be used for parking occupancy detection as well.

---

<sup>6</sup><https://www.asmag.com/suppliers/productcontent.aspx?co=nh&id=34962>

<sup>7</sup><https://en.wikipedia.org/wiki/E-TAG>

RFID sensors can provide accurate and reliable parking occupancy information. However, installing RFID tags on vehicles and readers in parking spaces is complex and expensive. RFID tags have a low power consumption and a relatively long battery life of 3 to 5 years. Nonetheless, the maintenance cost of RFID sensors for smart parking solutions is relatively high due to maintenance needs for both tags on the vehicles and readers in the parking spaces.

### 10.3.6 Cameras

Imagery captured by cameras overlooking parking spaces can also be used to detect vehicles and determine the occupancy of parking spaces. This is commonly done by training a machine learning model using a set of training images and applying the trained model to the captured imagery to detect vehicles and parking spaces in real time. Determining the occupancy of parking spaces is significantly simplified if the field of view of the camera is fixed. With a fixed camera, an image can be manually segmented to delineate the parking spaces in a pre-processing step and the segmentation will remain valid for all the images as long as the field of view of the camera remains unchanged. In effect, this will reduce the vehicle detection task to a simpler image classification task. The trained model is applied to each sub-image corresponding to a parking segment and classifies it into one of two categories: vehicle or vacant. Using state of the art deep learning methods and convolutional neural networks the classification of sub-images can achieve accuracies as high as 99% (Valipour et al., 2016; Acharya et al., 2018). An example of parking occupancy detection by classifying image segments is shown in Figure 10.2.

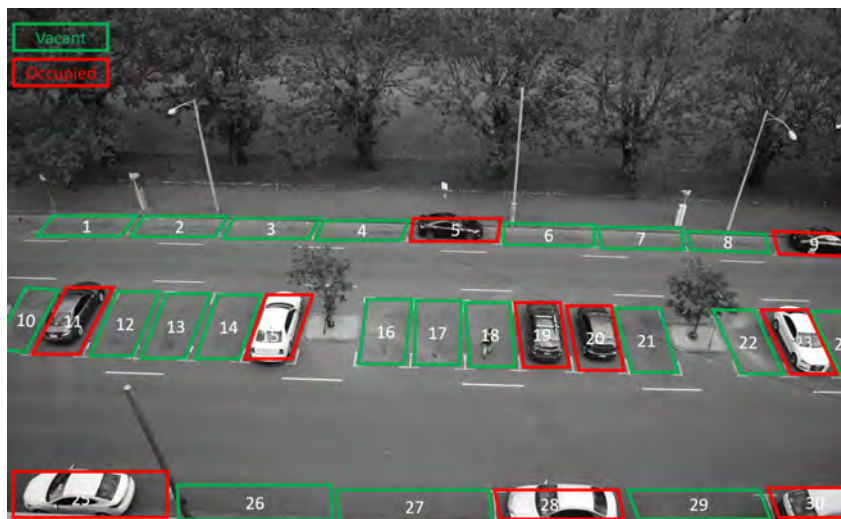


Figure 10.2: Image-based parking occupancy detection.

Cameras are often installed overhead and can be used in both indoor and outdoor parking lots. Image-based parking occupancy detection provides high accuracy at low cost, since cameras are relatively inexpensive and one camera can monitor multiple parking spaces. However, as passive sensors cameras are dependent on ambient light or a separate light source. Another important limitation is that the image-based approach is not a typical plug-and-play solution as the machine learning model needs to be trained on the images captured at the specific setup to achieve optimal performance. Also, to improve robustness to environmental conditions, the machine learning model must be trained on images captured in various lighting and weather conditions (e.g. rain, snow, fog, and low light). To overcome this limitation, recent works have studied the feasibility of transfer learning, where a machine learning model trained on a generic public dataset such as PKLot (de Almeida et al., 2015) is applied to images captured in a specific parking setting. Acharya et al. (2018) showed that this approach performs reasonably well but the achieved accuracy of 97% is slightly lower than that of a model trained on images of the same parking setting (99%).

Image-based parking occupancy detection provides high detection accuracy with moderate reliability due to its susceptibility to lighting and poor weather conditions. Cameras are relatively inexpensive to install and maintain. Also, in many cases pre-existing networks of surveillance cameras can be leveraged for parking occupancy detection.

### **10.3.7 Visible Light Sensors**

Visible light sensors measure the intensity of ambient light in the environment. To detect the occupancy of a parking space the sensor must be installed at a point where the light is obscured by a parked vehicle. This results in a reduced light intensity measured by the sensor, which is the basis for the detection of a parked vehicle.

Visible light sensors can be installed on the ground in both indoor and outdoor parking lots. However, vehicle detection by visible light sensing is easily influenced by the lighting conditions and any changes in the intensity of ambient light can result in detection errors. Transient light sources, such as headlights of other vehicles, and shadow cast by other objects can also seriously hamper the performance of visible light sensors for parking occupancy detection.

Overall, visible light sensors are considered inaccurate and unreliable, and despite their low installation and maintenance costs their use for parking occupancy detection is not common.

### 10.3.8 Contact Sensors

Contact sensors include pneumatic road tubes, inductive loop detectors, and piezoelectric sensors. These sensors generate a signal when they come in contact with a vehicle's tires. A pneumatic tube generates a burst of air pressure when pressed, which is converted to an electrical signal. An inductive loop contains an inductive element and an electronics unit, which can measure a decrease in the inductance of the loop caused by a passing vehicle. A piezoelectric sensor generates a voltage when subjected to pressure, which is proportional to the pressure or the weight of the vehicle. Contact sensors are designed for moving vehicles and are mainly used for monitoring traffic flow. However, if installed properly, they are capable of detecting the occupancy of parking spaces as well.

Contact sensors are installed on the ground and are more common for outdoor usage. Piezoelectric sensors can be placed under the asphalt surface as the load can be transferred through asphalt to the sensor.<sup>8</sup> Contact sensors are generally considered highly reliable as they perform well under different environmental conditions. New piezoelectric sensors can precisely measure the weight and determine the class of the vehicle. On the downside, contact sensors wear out quickly and require repair and maintenance regularly. Also, piezoelectric sensors are known to be sensitive to the temperature of the ground surface (Burnos et al., 2007).

Overall, contact sensors are accurate and reliable as their performance is not influenced by the environmental conditions. They are, however, expensive to install and maintain as they wear out quickly and require regular repair and maintenance.

### 10.3.9 Multi-sensor Parking Occupancy Detection

Different sensors have different strengths and limitations. When the strengths and limitations of two or more sensors are complementary, it makes perfect sense to fuse their data to overcome the limitations and achieve better results. For example, an active sensor used for parking occupancy detection may be more accurate and reliable but require more power or regular maintenance. In contrast, a passive sensor may be less accurate or less reliable but require little maintenance. Combining such sensors with complementary properties can result in high detection accuracy and reliability as well as low maintenance costs. Multi-sensor systems that take advantage of the complementary properties of different sensors have the potential to maximize the detection accuracy and reliability while minimizing the costs by reducing computational requirements and power consumption.

---

<sup>8</sup><http://diamondtraffic.com/product/Roadtrax-BL>

While several combinations of parking occupancy sensors are feasible, a common choice is the integration of magnetic and radar sensors.<sup>9</sup> The magnetic sensor has a very low power consumption but it is prone to magnetic interference. The radar sensor, on the other hand, is accurate and reliable, but also power hungry. In a typical fusion approach, the magnetic sensor samples the magnetic field strength continuously to detect changes that might indicate the presence of a parked vehicle. Once a change in the magnetic field strength is detected, the radar sensor emits a pulse to measure the distance accurately and reaffirm the detection result with high confidence. In this way, the integrated magnetic-radar sensor can detect parked vehicles accurately and reliably while minimizing the power consumption.

Multi-sensor approaches to parking occupancy detection generally achieve high detection accuracies with high reliability. Multi-sensor systems are, however, relatively expensive to install and require regular maintenance.

#### 10.4 Comparison, Considerations and Open Challenges

Table 10.1 provides a summary and comparison of parking occupancy sensors in terms of accuracy, reliability, installation cost and maintenance cost. Ideally, a parking occupancy sensor should provide high detection accuracy with high reliability, and can be installed and maintained at low cost. While none of the existing technologies meet all the above requirements, multi-sensor systems, ultrasonic sensors, and cameras seem more promising. However, when selecting a sensor for parking occupancy detection, it is important to take into consideration the application environment, whether it is outdoors or indoors, as well as environmental factors, such as noise, lighting and different weather conditions.

**Table 10.1:** Comparison of parking occupancy detection sensors.

Sensor	Accuracy	Reliability	Installation cost	Maintenance cost
Magnetic	Moderate	Low	Low	Low
Ultrasonic	High	Moderate	High	Moderate
Infrared	Moderate	Low	High	Moderate
Radar	High	High	High	High
RFID	High	High	High	High
Camera	High	Moderate	Low	Low
Visible light	Low	Low	Low	Low
Contact	High	High	High	High
Multi-sensor	High	High	High	Moderate

<sup>9</sup><https://bit.ly/3s9ob0m> – Bosch, 2020

Despite the advances in parking occupancy detection a few challenges still remain. The first challenge is the lack of a comprehensive quantitative comparison and benchmarking of the accuracy and reliability of parking occupancy sensors. Different sensors have been tested in different settings on sample sets of different sizes. This makes it difficult to compare and benchmark the performance of different parking occupancy detection sensors. Another challenge is the detection of improper parking, e.g., when a vehicle is not parked within the marked lines of a parking space, or illegal parking, e.g., in a disabled parking zone. Most existing sensors are only capable of detecting the presence of a vehicle in a marked parking space, but cannot detect the event where a vehicle is parked improperly or illegally. A third challenge is the recognition of different vehicle types, which can be useful for pricing or identification of improper/illegal parking (e.g., a car parked in a bus zone). Except for the RFID technology, where the reader can read the vehicle make, model and registration information from the tag, for the other sensors accurate recognition of vehicle types remains an open challenge.

## Bibliography

- Acharya, D., Yan, W., and Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. In Peters, S. and Khoshelham, K., editors, *Proceedings of Research@Locate18*, volume 2087, pages 33–40. CEUR Workshop Proceedings.
- Burnos, P., Gajda, J., Piwowar, P., Sroka, R., Stencel, M., and T., Z. (2007). Measurements of road traffic parameters using inductive loops and piezoelectric sensors. *Metrology and Measurement Systems*, 14(2):187–203.
- de Almeida, P. R., Oliveira, L. S., Britto, A. S., Silva, E. J., and Koerich, A. L. (2015). PKLot – a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937–4949.
- Kokolaki, E., Kollias, G., Papadaki, M., Karaliopoulos, M., and Stavrakakis, I. (2013). Opportunistically-assisted parking search: A story of free riders, selfish liars and bona fide mules. In *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 17–24.
- 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, MobiSys '10*, page 123–136. Association for Computing Machinery.
- Valipour, S., Siam, M., Stroulia, E., and Jagersand, M. (2016). Parking-stall vacancy indicator system, based on deep convolutional neural networks. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 655–660.



# 11 Parking Occupancy Detection and Slot Delineation Using Deep Learning: A Tutorial

DEBADITYA ACHARYA AND KOUROSH KHOSHELHAM

## Abstract

This chapter describes a simple method for parking occupancy detection and an automatic parking slot delineation method using CCTV images. These methods will be presented in the form of MATLAB tutorials with code snippets to allow the interested reader to implement the method and obtain results on a sample dataset. The first tutorial will involve fine-tuning a pre-trained deep neural network for vehicle detection in a sequence of CCTV camera images to determine the occupancy of the parking spaces. In the second tutorial, we perform spatio-temporal analysis of the detections made by a state-of-the-art deep learning object detector (Faster-RCNN) for automatic parking slot delineation. The dataset and the code is made public at <https://github.com/DebadityaRMIT/Parking>.

## Keywords

Automatic parking slot delineation, real-time parking occupancy detection, CCTV cameras, deep learning, tutorial

## 11.1 Introduction

Smart parking technologies are an indispensable part of urbanization to facilitate a congestion-free traffic flow. The advantages include less emissions and less waiting periods for drivers. These facts have motivated the research community to develop smart parking technologies, and real-time parking occupancy detection has become one of the key elements for the design of tomorrow's smart cities. While different sensor technologies exist for occupancy detection, they are usually expensive and require regular maintenance. The vision-based methods for parking occupancy detection provide an economical yet reliable alternative to the costly counter-based and sensor-based counterparts.

The rest of the section describes the motivation, related works and challenges of parking occupancy detection and automatic delineation of parking spaces using deep learning. Section 11.2 provides a brief overview on the definitions and the theory of machine learning in general. This is followed by introducing the deep learning architectures used for occupancy classification and parking space

[https://doi.org/10.34727/2021/isbn.978-3-85448-045-7\\_11](https://doi.org/10.34727/2021/isbn.978-3-85448-045-7_11)

This chapter is licensed under a Creative Commons Attribution-ShareAlike 4.0 International licence.

detection. Additionally, the details of the dataset are presented in this section. Section 11.3 presents the first tutorial where a deep learning image classifier is fine-tuning to perform parking occupancy detection. Also, several deep learning architectures are compared based on their performances and run-times. Section 11.4 presents the second tutorial where a deep learning object detector is used to perform automatic parking slot delineation. Subsequently, the model performance is improved by using spatio-temporal and statistical analysis of the detection. Section 11.5 concludes the observations of the tutorial.

### 11.1.1 Parking Occupancy Detection Using Vision-based Methods

The vision-based methods consist of cheap cameras to cover the whole parking area. The closed-circuit television (CCTV) cameras used for surveillance can also be used for occupancy detection. The images taken from these cameras are subsequently processed to provide the occupancy information. For a comprehensive review of other sensors for parking occupancy detection the reader is redirected to Chapter 10.

There are two challenges that limit the broad applicability of the vision-based methods. The first challenge is the low detection accuracy of vision-based methods as compared to the count-based or sensor-based methods (Amato et al., 2017). This lack of precision for vision-based methods can be linked to many factors, such as diverse appearances of the vehicles, environmental factors such as shadows, reflections and haze (due to sun and rain), occlusion by other vehicles (or other objects) in the line-of-view and distortion due to the oblique view of the cameras.

The second challenge is the delineation of the parking slots in the images. This delineation is not necessary for counter-based methods as the number of parking slots are fixed, and for sensor-based methods each parking slot is physically visited once to install the sensor. A parking area can be covered by several cameras, and perhaps hundreds of cameras for on-street parking. Manually labeling each parking slot is a laborious task. Moreover, the parking boundaries can change from time-to-time. Another related challenge arises in areas where the parking slots are not marked, especially in low- and middle-income countries such as India. An equally important challenge is the detection of improperly or illegally parked vehicles, e.g., when a vehicle is parked on the markings between two spaces or when several cars are parked in a large parking space designated for buses. Therefore, automatic ways to delineate the parking slot boundaries (or parking zones for unmarked parking areas) is highly desirable for smart parking solutions.

Robust image representations help in the accurate detection of the parking slots, and the recent deep learning methods have showed promising results in this aspect (Amato et al., 2017; Acharya et al., 2018). Therefore, in the next

sub-section we discuss the background of parking occupancy detection and automatic parking slot delineation using deep learning. The relevant definitions and theory related to the understanding of deep learning can be found in Section 11.2.

#### **11.1.1.1 Parking Occupancy Detection Using Deep Learning**

Parking occupancy detection is usually formulated as an image classification problem, where each image is either empty or occupied by a vehicle. Image classification follows a standard pipeline of feature extraction, and comparing the extracted feature with features belonging to different classes. In the past these images features were hand-crafted (or engineered) and showed poor performance for “unseen” examples. For instance, de Almeida et al. (2015) generated a robust dataset containing parking of different parking slots and used hand-crafted textural descriptors, such as Local Phase Quantization (LPQ) to perform parking occupancy detection. They report an accuracy of over 99 % while validating with the images from the same dataset, and around 89 % while testing with images of a different dataset. Other such examples of parking detection with hand-crafted images features includes the works of True (2007), Ichihashi et al. (2009) and del Postigo et al. (2015).

With the advances in machine learning algorithms, especially the recent deep learning and convolutional neural networks (CNNs), the feature extraction from the images have been automated, and state-of-the-art accuracies in image classification are reported. In the context of parking occupancy classification with CNNs, several past works (Valipour et al., 2016; Amato et al., 2016, 2017; Acharya et al., 2018) report excellent performance. These studies achieve greater than 99 % accuracy for the task of occupancy detection when being validated with unseen samples from the “same” dataset. The performance of the models on unseen samples from a “different” dataset is around 90 %–96 %. This improvement in the “generalizing” ability (or adaptability to unseen examples) of the CNNs demonstrates the robustness of the learnt features compared to the hard-crafted features.

Parking occupancy classification using CNNs can be done in two ways. The first approach involves fine-tuning a pre-trained CNN, like the approaches of Valipour et al. (2016); Amato et al. (2016, 2017). A pre-trained CNN contains the weights of a network that is trained on millions of images and is suitable for classification of several hundreds of classes. These pre-trained CNNs usually require weeks to train on graphics processing units (GPU) and perhaps can take years to train on normal CPUs. Therefore, using pre-trained networks saves the effort of training a network from scratch and can easily be adaptable to a particular problem by transfer learning.

However, the pre-trained networks are not suitable for a two-class classification task, like parking occupancy, where a class is either “empty” or “occupied”. Therefore, to adapt to the classification problem, we fine-tune the CNN by with some example images (often a couple of thousands) to adapt the weights of the network for the specific classification task. This is performed by back-propagating the loss using an objective loss function. For a classification task, cross-categorical loss is usually used. However, one of the disadvantages of this method is the computational power required for the fine-tuning process. It can take several minutes on GPU or several hours for fine-tuning a network with a couple of thousand images.

The second approach involves extraction of the image features using a pre-trained CNN and then performing classification using support vector machines (SVM), like the approach followed by Acharya et al. (2018). SVMs are a kind of machine learning algorithms that project the features into higher dimensional feature spaces to find the optimum hyperplane that separates the classes. This approach of training is faster as compared to fine-tuning the CNNs and the whole training can be performed with CPU within minutes. This reduction in the training time is due to the elimination of the back-propagation, and because the training time of SVM is considerably less (few seconds for couple of thousand samples). For a CPU-friendly MATLAB tutorial of this CNN + SVM method please follow our previous work (Acharya et al., 2018) which is available at <https://github.com/debaditya-unimelb/real-time-car-parking-occupancy>.

We present the fine-tuning approaches in Section 11.3. Additionally, we compare the performance in terms of precision and computational times of different CNN architectures. This information will help the audience to decide the trade-off between performance and computational need to check the suitability for real-time applications.

#### **11.1.1.2 Delineation of the Parking Slots Using Deep Learning**

Delineation of parking slots (knowing the locations of the parking slots) is required prior to accurate parking occupancy detection. Currently, delineation is performed manually (Cai et al., 2019; Khan et al., 2019; Sairam et al., 2020; Paidi et al., 2020). These studies use deep learning-based object detectors such as Faster-RCNN (Ren et al., 2015) to detect the vehicles, subsequently, compare the location of the detections with the manually delineated parking slots to estimate the occupancy.

To eliminate the manual delineation of parking slots, some researchers (Ahmad et al., 2019; Ding and Yang, 2019) have used automatic object detectors to detect the vacant and the occupied parking slots directly in the images using Faster-RCNN, Mask-RCNN (He et al., 2017), Retina-Net (Lin et al., 2017), and YOLO (Redmon and Farhadi, 2018). However, such methods do not take into

account the actual number of the parking slots available in the area, rather they report the number of detections (both empty and occupied) made by the object detector. Because these object detectors always miss some of the parking slots, the parking estimates might not be practical for all applications. Moreover, object detection pipeline involves localizing the objects in the images. This particular step is computationally expensive as the system has to process thousands of proposals to identify the correct detection. For instance, one forward pass through ResNet50 (He et al., 2016) for image classification needs approximately 0.1 seconds on CPU, and processing the same image with the same ResNet50 in Faster-RCNN framework on CPU needs around 12 seconds in MATLAB.

Differently, there are few other approaches that perform automatic parking slot delineation. Víttek and Melničuk (2018) propose an automatic method of delineation of the parking spaces in a multi-camera framework using histograms of oriented gradients (HOG) and a sliding window to perform vehicle space classification using SVM. The authors do not report the detection accuracies, and in addition the HOG features are susceptible to lighting changes. Nieto et al. (2018) use satellite images to manually register the parking slots and use an input of the number of parking slots to automatically delineate the parking slots. However, the method is not completely automatic as it needs input from a skilled operator to actually count the number of parking slots and for entering three common points (ground control points).

Another research direction of automatic delineation of parking slots can be found in the works of Jung et al. (2009); Suhr and Jung (2013); Zhang et al. (2018), however they are vehicle-centric and rely on the cameras installed in the vehicles to detect the parking slot marking automatically. However, such methods have not been applied yet for parking slot delineation from fixed cameras, and are a future research direction.

### 11.1.2 Contributions

The following are the main contributions of the chapter:

- A pre-trained CNN is fine-tuned with the PKLot dataset for parking occupancy detection, and is tested with a Barry Street dataset. Different CNN architectures are compared in terms of their accuracies and run-times to demonstrate the suitability for real-time applications on both CPU and GPU.
- A novel off-line method is proposed for automatic parking slot delineation by performing spatio-temporal analysis of the detected vehicles using state-of-the-art object detector Faster-RCNN (Ren et al., 2015). Compared to the previous approaches, the proposed pipeline eliminates the requirement of localizing the objects on-line, and generates parking slots that would have been otherwise be manually delineated. This off-line method eliminates

the requirement of localizing the vehicles, and reduces the object detection problem to an image classification problem that significantly reduces the computational requirements. We present the approach and the related tutorial in Section 11.4 to prove the concept.

Section 11.2 introduces the prerequisites of the tutorials, such as definitions, related theory, software toolboxes and subsequently the dataset. Section 11.3 demonstrates fine-tuning a pre-trained network, and compares different network architectures in terms of achievable accuracies with the sample dataset and the run-times. Section 11.4 demonstrates the automatic parking slot delineation using a vehicle detector. Section 11.5 concludes the finding of the tutorial.

## 11.2 Prerequisites

In this section, we start by presenting the theory and definitions that are used throughout the tutorials. For completeness, we have repeated some of the related theory already presented at the beginning of the chapter. Subsequently, we introduce the software and toolboxes required running the tutorials. Lastly, we describe the dataset that we used for the tutorials, which we made public.

### 11.2.1 Definitions and Theory

**Machine learning** are a set of computer algorithms that build a mathematical model based on a training data, which can be used to make predictions or decisions of unseen data based on learnt representation of the features. Neural networks and deep learning are included in this class of algorithms. Machine learning can be broadly classified as supervised machine learning, un-supervised machine learning and reinforcement learning. In this chapter we use supervised machine learning approaches, where we provide the samples of training data with their respective labels (ground truth annotations).

**Neural networks** are networks inspired from the biological neural networks of the brain and are composed of artificial neurons (containing weights and biases) to perform many complex operations, such as classification. These networks learn a feature representation automatically and eliminate the manual feature selection process. These networks are composed of connected layers, where each layer contains many neurons and the training process involves back-propagation by minimizing an objective loss function.

**Deep learning** refers to the machine learning algorithms which deal with neural networks that contain many layers of neurons. Adding increased depth to the neural networks provide the networks ability to perceive complex operations that are not possible by their “shallow” counterparts. Recently, deep convolutional

neural networks have achieved the state-of-the-art accuracies in classification and object recognition tasks, sometimes even surpassed the human ability.

**CNNs** consists of many layers of image “convolutions” containing learnable kernels that convolute the whole image and hence create a hierarchy of increasing complex image features. These image features are learnt automatically thereby eliminating the need of fragile hand-engineered image features. These learnt image features are unique representations of the images, and are often used for image classification and object detection. In this chapter we have used many CNN architectures, v.i.z. AlexNet (Krizhevsky et al., 2012), GoogleNet (Szegedy et al., 2015), MobileNet v2 (Sandler et al., 2018), ResNet50 (He et al., 2016), SqueezeNet (Iandola et al., 2016) and VGG-16 (Simonyan and Zisserman, 2014). We selected these networks to demonstrate the effects of network run-time and the achievable accuracy with limited training data. For the scope of the tutorials we only explain ResNet50 in the following lines.

**Pre-trained networks** are trained with millions of images of publicly available image datasets containing different classes. The training can take a couple of weeks, depending on the network architecture and training data. To save the immense training effort before using deep networks, these pre-trained models are often used for other tasks by fine-tuning them.

**Fine-tuning** refers to the process of training a pre-trained network with relatively small examples to adapt to a different task. This is achieved by the process called transfer learning.

**Transfer learning** is the process of applying learnt knowledge in one domain to solve a different but related problem. For instance, a pre-trained network trained to perform image classification of thousands of classes that contain vehicles, cats, and dogs can be used to differentiate between types of insects, a task that it was not trained to do.

**Training data** refers to the samples that are used during training the model.

**Test data** refers to the sample that needs to be classified.

**Over-fitting** refers to a condition where the classification accuracy of the trained model is excellent on the training data, but its performance is poor on test data. Therefore, during the training process, a validation data is generated as a subset of training data which is used for evaluating the accuracy of the trained model independently.

**Loss function** also known as cost function or objective loss function that we try to minimize during the learning process by back-propagating. The simplest form of this function is the difference between observed and the actual values. For classification problems, a cross-entropy loss function is often used.

**Back-propagation** refers to the process of propagating the gradients from output to input to update the weights of the network for the intended operation.

The back-propagation is achieved by using an optimizer, and the weights of the neurons are updated using a hyper-parameter called learning rate.

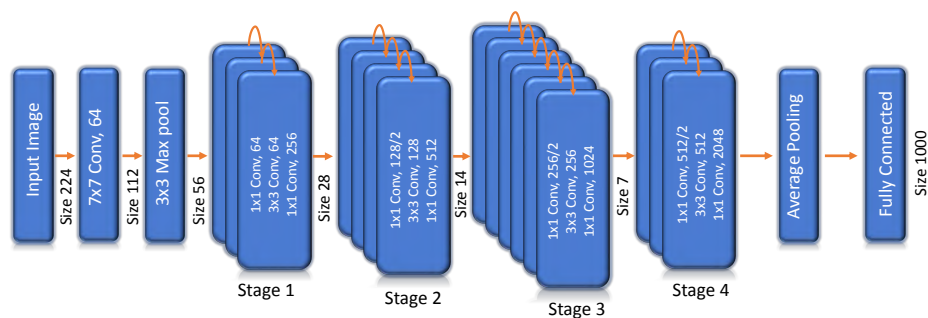
**Optimizers** are the iterative methods of optimizing the loss function by calculating the gradients (or rate of change). They connect the weights of the individual neurons with the loss function with the help of a learning rate. The objective here is to reach the global minima or the minimum possible value of the loss function. The most commonly used optimizer is stochastic gradient descent and its variants.

**Learning rate** refers to the rate of update of the gradients for each individual neuron throughout the network. A higher learning rate might help to reach minima fast, but can end in local minima. The target of the optimization is to reach the global minima, and hence learning rate is one of the key training parameters of a neural network.

**Epoch** refers to the training interval when the neural network is trained with one complete dataset. Usually, a neural network needs to be trained on several epochs of data before it converges to an optimal solution.

**Learning curve** refers to the graphical representation of the model learning with the amount of training data. This curve often contains the training loss, validation loss, training accuracy and validation accuracy, and is used to identify whether the model is over-fitting.

**ResNet50.** One of the challenges of deep CNNs and deep learning in general is the problem of vanishing gradients, where the gradients during the back-propagation becomes infinitely small for the shallow layers. To address this chal-



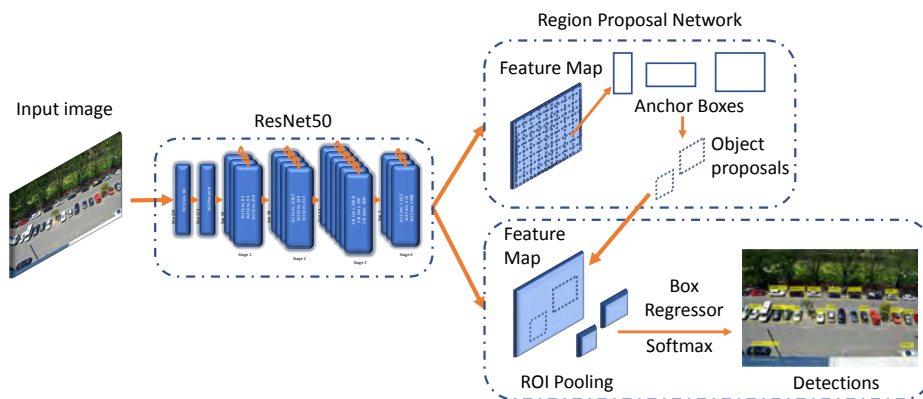
**Figure 11.1:** The architecture of ResNet50 containing 50 layers. Stages 1-4 contain blocks of length 3, 4, 6 and 3 respectively, where each block consists of three convolutional layers.

The main innovation in this architecture is the presence of the “skip connections” or the identity mapping (orange curved lines on the top of blocks), where



the output of a previous block is connected to the next block. This skip connection, helps to alleviate the vanishing gradient problem by skipping one or more layers. The result is a deep network with the state-of-the-art accuracy in image classification. The input to the network is an image of 224 x 244 pixels and the output is a 1000-dimensional feature vector.

**Faster-RCNN** (Ren et al., 2015) is an object detection algorithm that performs the task of localizing objects on the images and its subsequent classification. This algorithm needs a CNN as its backbone for operation, and is shown in Figure 11.2. The CNN extracts features and generates a feature map using the “*activation\_40\_relu*” layer. This feature map serves as the input to the Region Proposal Network (RPN) that generates the object proposals. The RPN searches for potential objects throughout the image at regularly gridded anchor points using anchor boxes of different shapes and sizes. The object proposals from RPN are used to create Region of Interest (ROI) pooling on the feature map to extract the features of the potential objects. The final bounding boxes and the classes



**Figure 11.2:** The architecture of Faster-RCNN containing a ResNet50 backbone.

**t-SNE algorithm** is a non-linear dimensionality reduction algorithms that is used for visualization of the higher dimensional data, and is often used to assess the quality of the features for the task at hand.

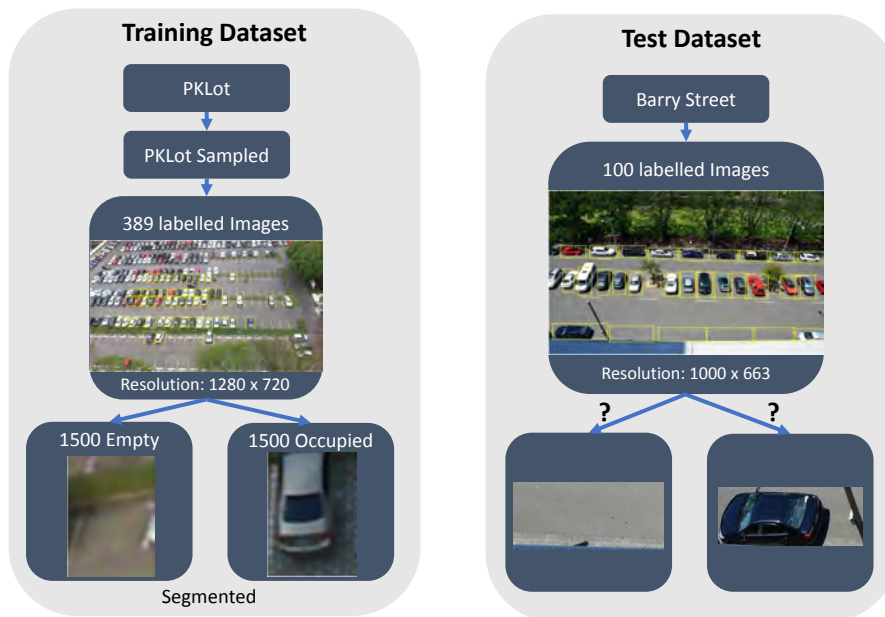
### 11.2.2 MATLAB and Toolboxes

The tutorials are intended to run on MATLAB 2020a, although the code can run in MATLAB versions higher than 2018a. Additional toolboxes might be required to run the experiments that include the computer vision toolbox, statistics and machine learning toolbox, deep learning toolbox, signal processing toolbox and automated driving toolbox. For running the live script smoothly, please ensure

that you increase the Java heap memory of MATLAB, as demonstrated at the start of the live script.

### 11.2.3 Dataset Description

The code is made available at Github (<https://github.com/DebadityaRMIT/Parking>). The provided file is in form of a MATLAB live script (.mlx file) that contains all the outputs embedded within the script. Therefore, the user can view the results of the experiment without running the experiments. By changing the default configurations, an interested reader can run the tutorials online. For running the experiments, the data (including the code) can be downloaded from Figshare (<https://rmit.figshare.com/ndownloader/files/24753887>). Three datasets, namely BarryStreetData, PKLotSampled and PKLotSegmentedSampled, along with the trained models and supporting files are present in the archive. Figure 11.3 shows the training and the test datasets.



**Figure 11.3:** The training and the test datasets. The CNNs are trained with the PKLot dataset and are tested entirely on the Barry Street dataset.

**BarryStreetData** was captured by the authors from the rooftop of the Faculty of Business and Economics Building, The University of Melbourne, and shows on-street parking spaces along Barry Street, Melbourne. This dataset was taken by a camera at different intervals throughout the day (except by night) having an image resolution of 1000 x 663 pixels. We created a subset of the dataset containing 100 images for these tutorials. This serves as our test data throughout

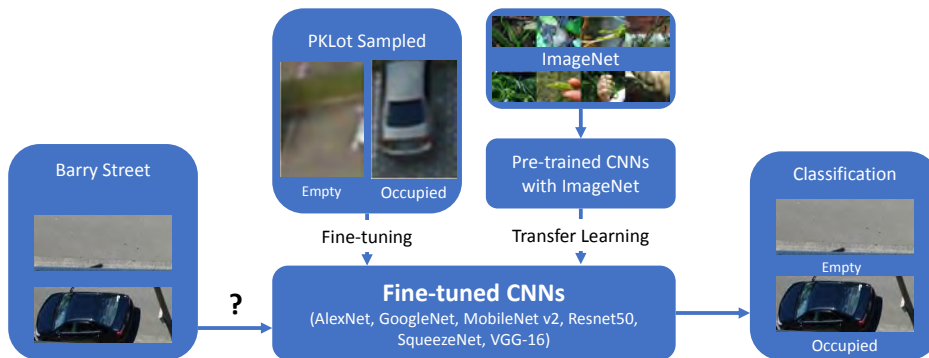
the tutorials. We also provide the ground truth annotations of the parking slots delineations (28 slots) and the occupancy (2800) for evaluating the accuracy.

**PKLotSampled** contains 279 randomly sampled images (having a resolution of 1280 x 720 pixels) from original PKLot dataset (de Almeida et al., 2015)) and an additional 90 images that have been rotated to remove the dataset bias, totaling the number of images to 389. Dataset bias often happens due to the presence of a particular pattern in the training dataset, and in turn biases the classifier to make wrong predictions on unseen data. In the current context, the vehicles in the PKLot dataset (parking area PUCPR) were parked only in up-down orientation. Therefore, we rotated them to make the orientation of the vehicles left-right. The ground truth annotations of the parking slot delineations are provided for fine-tuning the vehicle detector using Faster-RCNN.

**PKLotSegmentedSampled** contains 3000 randomly sampled (1500 empty and 1500 occupied) image crops from the original PKLot dataset of varied resolutions ranging from 32 x 39 pixels to 68 x 63 pixels. These images are used to fine-tune the CNNs.

### 11.3 Tutorial 1: Parking Occupancy Detection by Fine-tuning a Pre-trained CNN

Figure 11.4 shows the pipeline of the tutorial, where we will fine-tune CNNs (pre-trained with ImageNet dataset) with 3000 segmented images of PKLot dataset. We will demonstrate the fine-tuning process for ResNet50 in Section 11.3.1. Subsequently, we will test the fine-tuned ResNet50 with the Barry Street dataset to check the generalizing (adaptability to other dataset) ability of the CNN in Section 11.3.2. Also, we benchmark the accuracies of different CNN architectures and report their results in Section 11.3.3.



**Figure 11.4:** The pipeline of Tutorial 1, where we fine-tune a pre-trained CNN with the PKLot dataset and test with the Barry Street dataset.

### 11.3.1 Fine-tune ResNet50 Network with the PKLot Dataset

```

1 TrainOnline = false;
2 .
3 .
4 else
5 load('TrainedDetectorResnet50.mat');
6 load('trainingInfoResnet50.mat');

```

By default, “TrainOnline” is set as false, and therefore, the fine-tuned CNN is loaded directly without performing fine-tuning online. This option can be changed to perform the training online. GPU should be used for fine-tuning the network and it takes around 30 minutes (in NVIDIA Tesla P100).

“TrainOnline” being set to true, we start by loading pre-trained ResNet50 and the segmented images into an image datastore. A datastore contains the list of the file-names, and does not actually load the images into memory. The datastore also creates labels automatically based on folder names. For instance, it creates “Empty” and “Occupied” labels for each image automatically. Subsequently, we split the images into training (70%) and validation sets (30%) using the following lines of codes:

```

1 % load the pre-trained model in the workspace
2 load('Resnet50FeatureExtractor.mat');
3
4 % Create image datastore from folder and label by folder name
5 imds = imageDatastore([pwd '/PKLotSegmentedSampled/'], ...
6 'IncludeSubfolders',true, 'LabelSource','foldernames');
7
8 % Randomly split the training set (70%) and the validation set (30%)
9 [imdsTrain,imdsValidation] = splitEachLabel(imds,0.7,'randomized');

```

The pre-trained ResNet50 contains 1000 classes, and currently it is unsuitable for making occupancy predictions. Therefore, we need to replace the classification and the fully-connected layers with the two classes which correspond to “Empty” or “Occupied”. Subsequently, we extract the connections of the newly created graph and connect them to form a new CNN.

```

1 numClasses = 2; % Number of classes: Occupied and Empty
2
3 % replace the classification and the fully connected layers
4 lgraph = replaceLayer(lgraph,learnableLayer.Name,newLearnableLayer);
5 lgraph = replaceLayer(lgraph,classLayer.Name,newClassLayer);
6
7 % extract connections of the new graph
8 connections = lgraph.Connections;
9

```

```

10 % connect graph and new layers
11 lgraph = createLgraphUsingConnections(layers,connections);

```

To reduce over-fitting and to improve the generalization ability of the CNN, data augmentation is performed on the fine-tuning dataset. Data augmentation involves transforming the fine-tuning images without changing the total number of images for each epoch. Here we perform two transformations: 1) reflection along X and Y axes, and 2) change the X and Y scales of the images. Also, we need to resize the fine-tuning images according to the input size of the CNN, which is fixed for each CNN architecture. Subsequently, we generate the validation dataset to validate the performance of the CNN.

```

1 % set range of changing the scales of the images along X and Y axes
2 scaleRange = [0.9 1.1];
3
4 % define a data augmenter with steps to perform
5 imageAugmenter = imageDataAugmenter( ...
6 .
7 .
8 'RandYScale', scaleRange);
9
10 % define augmented fine-tuning dataset
11 augimdsTrain = augmentedImageDatastore(inputSize(1:2), imdsTrain, ...
12 'DataAugmentation', imageAugmenter);

```

In the next step we start the fine-tuning process by setting up the training options. We set the optimizer to stochastic gradient descent with momentum with an initial learning rate of 0.005 that we reduce at every 5 epochs by a factor of 0.5. The maximum number of epochs is set to 20. This high initial learning rate helps the model to converge fast, otherwise it might have taken more epochs to reach the same level of performance. We reduce the learning rate slowly to avoid reaching the gradient descent to a local minima. We also set the batch size as 10 (this depends on the memory of the GPU). Increasing the batch size speeds up the fine-tuning, however, higher learning rate should be used, as large batch size usually provides a strong regularization. Also, we shuffle the training data at every epoch to remove any dataset bias due to image sequences. To check the performance of the fine-tuning we set the validation frequency as 3. We could perhaps use a higher frequency, but that would slow the fine-tuning process without any improvement.

```

1 % define validation frequency
2 valFrequency = 3;
3
4 % set fine-tuning options
5 options = trainingOptions('sgdm', ... % stochastic gradient ...
    descent with momentum

```

```

6 'MiniBatchSize',10, ... % number of samples to train together
7 'MaxEpochs',20, ... % maximum number of epochs
8 'InitialLearnRate',5e-3, ... % learning rate
9 'Shuffle','every-epoch', ... % shuffle data to reduce over-fitting
10 'LearnRateDropFactor', 0.5, ... % factor to reduce learning rate
11 'LearnRateDropPeriod', 5, ... % epoch after learning rate dropped
12 .
13 .
14 % Fine-tune the newly created CNN and save training information
15 [net,traininfo] = trainNetwork(augimdsTrain,lgraph,options);

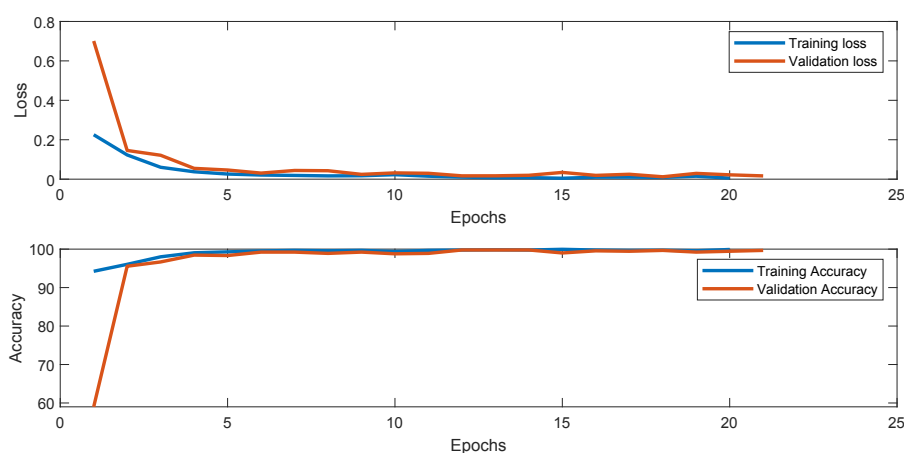
```

Fine-tuning approximately takes 30 minutes to complete on GPU, and we can now plot the training loss and accuracy curves, where the variable “traininfo” contains the details of the training. After post-processing the data, we can visualize the curves using the following lines:

```

1 plot(TrainLoss, 'lineWidth', 2); hold on; % to show two plots
2 plot (ValidationLoss, 'lineWidth', 2)
3 legend('Training loss', 'Validation loss')
4 xlabel('Epochs'); ylabel('Loss');

```



**Figure 11.5:** Training curves of the fine-tuning process. (a) The fine-tuning and validation loss vs epoch. (b) The fine-tuning and validation accuracy vs epochs.

### 11.3.2 Test the Fine-tuned Network with the Barry Street Dataset

We have fine-tuned the CNN in the previous step, and now we will test it with the Barry Street dataset. By default RunOnline is set to false, therefore, the results of classification are loaded into the workspace without running the CNN on-line.

```

1 RunOnline = false;
2 .
3 .
4 else
5 load('BarryStreetTestResults.mat');

```

“RunOnline” being set to true will read the 100 Barry Street images and will crop out individual 28 parking slots of each image. Subsequently, these cropped images will be passed to the fine-tuned CNN for classification. We start by setting path to the directory of the Barry Street images. Note that the variable “pwd” refers to present working directory in MATLAB. In the next step we load the ground truth of the Barry Street dataset containing the occupancy status (2800) and delineation of the parking slots (28). These delineations are in form of bounding boxes (variable “ParkingSlots”) that are used to crop the individual parking slots. A bounding box is defined by  $[x, y, w, h]$ , where  $[x, y]$  represents one corner of the box, and  $w$  and  $h$  denote the width and height of the box. These cropped images are further resized to suit the input size of the CNN. Running on-line the classification should take around 4 minutes on CPU to complete, or approximately 2.5 seconds for each image (for classifying 28 parking slots). On a GPU the whole process takes a couple of seconds. The results of the classification are saved in the variable “YPred” and the classification scores are saved in variable “probs”. Subsequently, we check the accuracy of the classification by comparing YPred with “AnnotationTable”, where variable AnnotationTable contains the ground truth parking occupancy. In the last step we plot the confusion matrix and visualize some of the wrongly classified image.

```

1 % set the directory containing the Barry Street dataset
2 imageName = dir(fullfile(pwd, 'BarryStreetData\', '*.JPG'));
3
4 % load Barry St occupancy and parking slot delineation ground truth
5 load('GroundTruthBarryStreet.mat')
6
7 % crop individual parking slots from the Barry Street image
8 cropImage = imcrop(BarryStreetImage, ParkingSlots(m,:));
9
10 % resize each cropped image to suit the input size of CNN
11 imdsIm = imresize(cropImage, inputSize(1:2));
12
13 % Predict the occupancy status using the fine-tuned CNN.
14 [YPred(count), probs(count,:)] = classify(net, imdsIm);
15
16 % plot the confusion matrix
17 plotconfusion (categorical(AnnotationTable), YPred);

```

From Figure 11.6 we observe that only one occupied parking slot is classified as empty, and 21 empty slots have been wrongly classified as occupied. Therefore, the fine-tuned CNN is slightly less precise while classifying empty parking

**Confusion Matrix**

<b>Output Class</b>	Empty	572 20.4%	1 0.0%	99.8% 0.2%
	Occupied	21 0.8%	2206 78.8%	99.1% 0.9%
		96.5% 3.5%	100.0% 0.0%	99.2% 0.8%
		Empty	Occupied	
		<b>Target Class</b>		

**Figure 11.6:** The confusion matrix of the classification, showing that the overall accuracy is 99.2%. The mis-classifications are largely due to wrong classifications of empty parking



**Figure 11.7:** Some of the wrongly classified image patches with their respective classification scores.

slots. Upon visualizing some of the wrongly classified parking slots in Figure 11.7 we observe most of them contain a part of a vehicle inside the image crop. Also, it is observed that the images that does not contain the vehicles have low classification scores. Finally, we visualize the occupancy of the parking slots in Figure 11.8.





**Figure 11.8:** The visualization of the parking occupancy, where red colour denotes the occupancy and green colour denotes vacant spaces.

### 11.3.3 Time and Accuracy Benchmarking Using Different CNN Architectures

In this section we compare different CNN architectures in terms of their runtime and accuracies achievable to help the audience choosing the right architecture for their needs. For the experiments, an i7 5600U CPU @ 2.6 GHz was used, and the GPU was NVIDIA Tesla P100 with 12 GB of memory. Note we used only one core of the CPU for the experiments. We compared ResNet50 with AlexNet, GoogleNet, MobileNet v2, SqueezeNet and VGG-16.

Table 11.1 shows the accuracy achieved by fine-tuning different CNN architectures, and their run-times and train-times with GPU. It is observed that the generalizing ability of all the fine-tuned networks are excellent, however, we see lower accuracies for AlexNet, MobileNetv2 and VGG-16 networks. In terms of run-time we see SqueezeNet is the fastest one, and the slowest one being the VGG-16 network. Note these run-times are average times and include overheads such as reading files from the disk and cropping images. Therefore, SqueezeNet can process approximately 50 parking lots in just one second via CPU alone with 99.2% accuracy. In regard to training time, all of the CNN are fine-tuned under 10 minutes on the GPU. Note that different learning rates were used for fine-tuning the different networks.

**Table 11.1:** The comparison of the achievable accuracy and the run-times for different CNN architectures by fine-tuning the networks on GPU. The fine-tuning data were the PKLot segmented images, and test data was Barry Street data.

Network	Accuracy	Run-time on CPU (msec)	Train time on GPU (min:sec)	Base learning rate
AlexNet	97.3 %	33.6	1:46	0.001
GoogLeNet	99.2 %	55.4	6:27	0.001
MobileNet v2	98.6 %	62.7	9:23	0.005
ResNet50	99.2 %	100.1	8:09	0.005
SqueezeNet	99.2 %	19.3	2:42	0.005
VGG-16	98.5 %	400.0	4:56	0.001

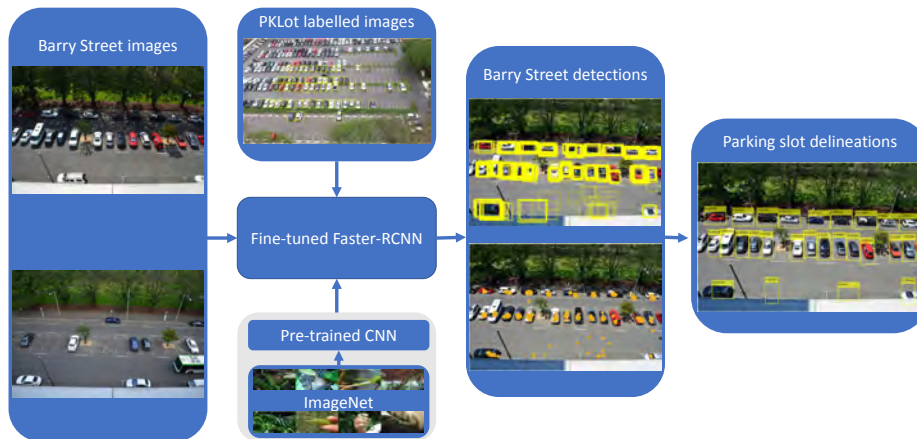
## 11.4 Tutorial 2: Automatic Parking Slot Delineation Using Deep Learning

In Tutorial 1, we have used the ground truth delineations of the parking slots in the images. As shown in Figure 11.9, in this tutorial we describe a novel method to for automatic parking delineation of Barry Street data by fine-tuning the Faster-RCNN object detector with the PKLot dataset. This delineation is performed using spatio-temporal analysis of the detected vehicles in the Barry Street dataset. We detect vehicles in Barry Street images for many frames, and then cluster the detections to individual parking slots using a robust density-based clustering algorithm (Ester et al., 1996). Subsequently, we estimate the coordinates of the parking slots by weighing the coordinates of the individual detections with the scores of the detections. Further we refine and improve the coordinates of the parking slot delineations using the statistics of all the detections.

### 11.4.1 Assumptions and Limitations

The key assumption of the method is that the vehicle detections made by Faster-RCNN will cluster more often in the actual parking slots, as compared to other parts of the image, such as on the roads. This is a valid assumption as the vehicles are parked longer than they are actually on the road. Therefore, we should be able to estimate individual parking slots by combining all the detections in each cluster. The second assumption of the method is that a vehicle takes around 80 % space of the parking slot. Therefore, each parking slot is approximately 1.2 times the length of the parked vehicle. The last assumption is that the size of the parking slots remains approximately the same throughout the camera view.

Coming to the limitations of the assumptions made. Some of the parking slots might be missed as a result of low parking rate. For instance, the vehicles might be parked less often in reserved parking slots (like for people with special needs),



**Figure 11.9:** The pipeline of Tutorial 2, where we use a fine-tuned Faster-RCNN object detector to automatically delineate the parking slots.

or parking slots that are far from the entrance of the parking area. Additionally, dense traffic during peak hours might also result in several detections on road. Although these detections might not form dense clusters as compared to the actual parking slots, the possibility of delineation of parking slots on roads cannot be neglected. Lastly, for a very oblique view of the cameras, the vehicles that are far away from the camera appear smaller than the vehicles that are closer.

#### 11.4.2 Vehicle Detection Using Faster-RCNN

We start the tutorial by loading a pre-trained Faster-RCNN detector that is trained on images of highway taken from a camera inside a moving vehicle. This pre-trained detector performs well to detect vehicles on highways, however, its performance to detect vehicles in CCTV images that have been taken from an oblique view is questionable. Therefore, we will fine-tune this detector with the PKLot dataset that includes the bounding boxes of the detections.

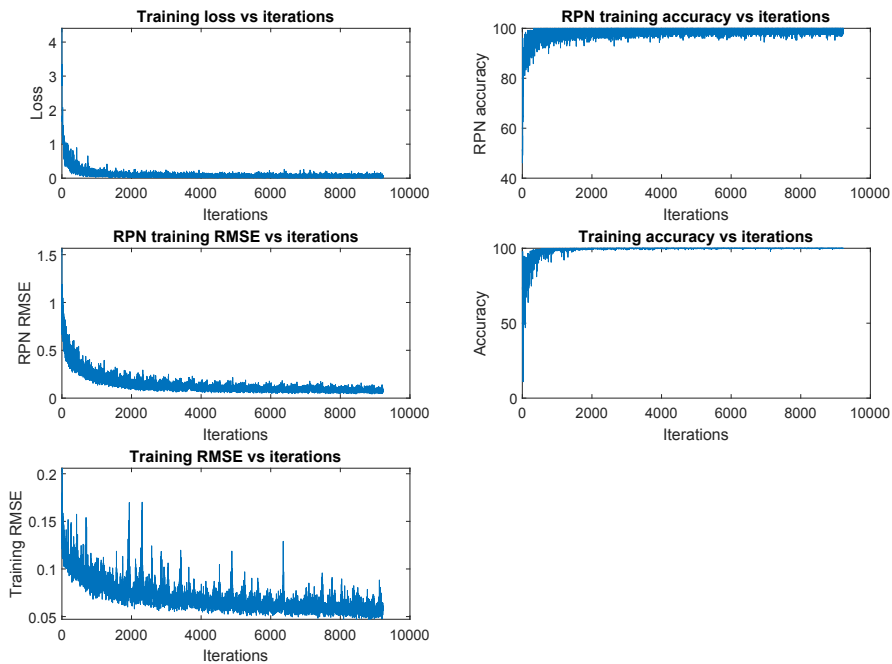
By default, “train” is set to false and the Faster-RCNN detector fine-tuned with the PKLot dataset is loaded into the workspace.

```

1 train = false;
2 .
3 .
4 else
5 load('Fine-tunedFRCNNResnet50.mat'); % load the trained model
6 load('trainingInfoFRCNNResnet50.mat'); % load training info

```

Setting “train” to true will start training the Faster-RCNN detector with the PKLot dataset. We set batch size to 1 to accommodate the model to GPU memory. We set the Negative Overlap Range to {0 0.3} and Positive Overlap Range to {0.6 1}. This means that a detection is considered as negative detection when the Intersect over Union (IoU) falls between 0 and 0.3, and is considered positive when it falls in the range of 0.6 and 1. The IoU is the area of intersection of two bounding boxes. The following lines fine-tune the network.



**Figure 11.10:** Training curves of Faster-RCNN fine-tuned with the PKLot dataset.

Figure 11.10 shows the training curves. There are 369 images in each epoch, and we trained the network for 25 epochs, hence resulting in approximately 10000 iterations. We observe that training RMSE continues to improve up to 9000 iterations. Subsequently, we test the fine-tuned network with the Barry Street images again, and Figure 11.11 shows the results of the detections.

```

1 options = trainingOptions('sgdm', ...
2 'MiniBatchSize', 1, ...
3 .
4 .
5 % Train an R-CNN object detector
6 [rcnn,traininfo] = trainFasterRCNNObjectDetector(vehicleDataset, ...
7 detector, options, ...
8 'NegativeOverlapRange', [0 0.3], 'PositiveOverlapRange',[0.6 1]) ...
9 % train the model

```



**Figure 11.11:** The detections of one Barry Street image with Faster-RCNN fine-tuned with the PKLot dataset.

Note that the image size of the images in the PKLot dataset is 1280 x 720 pixels, whereas for the Barry Street dataset it is 1000 x 663 pixels. Ideally, we should use same image size to reduce any bias. Therefore, we pre-process the Barry Street images to a resolution of 1280 x 720 pixels without distorting the aspect ratio. This is done by adding the Barry Street images to a blank 1280 x 720 image. We use this image as an input the Faster-RCNN detector.

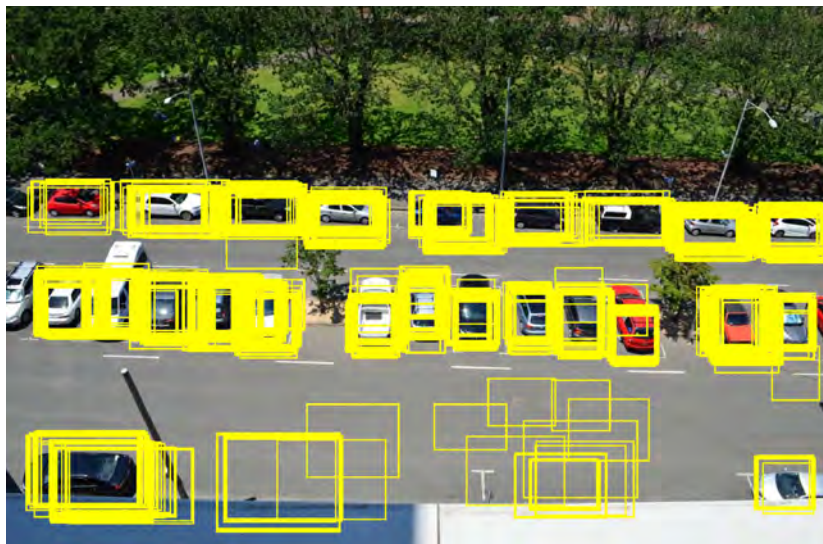
```

1 % Preprocess the Barry Street frame (663 x 1000) to original ...
   training image size (720 x 1280)
2 BarryStreetImageProcessed = uint8(zeros(720,1280,3));
3 BarryStreetImageProcessed(58:720, 141:1140,:) = BarryStreetImage;
4
5 % Run the trained detector on Barry Street image. This step takes ...
   1 minute on \acrshort{cpu} and a second on \acrshort{gpu}.
6 [bboxes,scores] = detect(rcnn,BarryStreetImageProcessed);

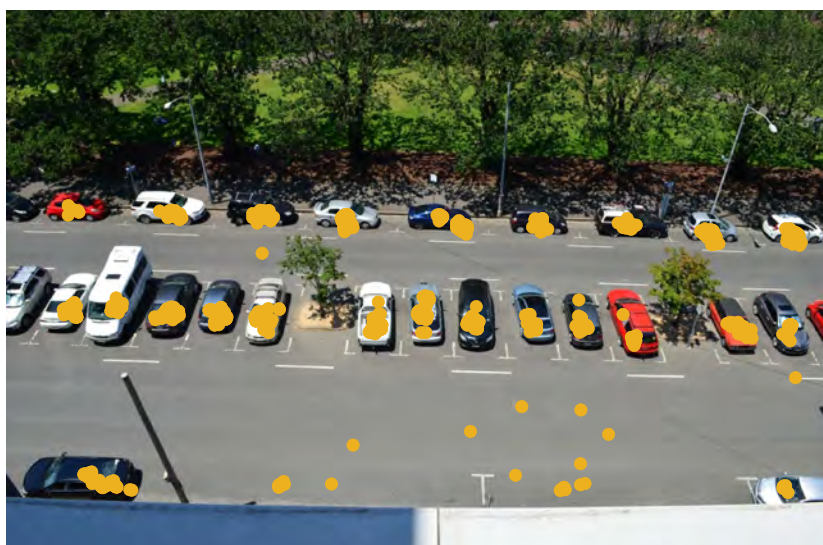
```

The outputs of the detector are the bounding boxes and their respective scores. From Figure 11.11 we observe that 17 vehicles are detected. However, we notice that not all the vehicles are detected. Therefore, in the next step we run the fine-tuned network for all the 100 Barry Street images. Figure 11.12 shows all the detections, and Figure 11.13 shows the centres of the detections. Subsequently, we save all the bounding boxes and scores of the detections for post processing.





**Figure 11.12:** The detections of all Barry Street images with Faster-RCNN fine-tuned with the PKLot dataset.



**Figure 11.13:** The centre of detections of all Barry Street images with Faster-RCNN fine-tuned with the PKLot dataset.

Next, we use a robust density-based clustering algorithm (Ester et al., 1996) to estimate the number of clusters based on the spatial distance between the neighboring points and the number of occurrences.

```
1 % Use density-based algorithm
2 idx = dbscan(bboxesTotal,20,4);
3 Classes = unique(idx,'rows');
```

Ideally the spatial distance should be equal to the standard deviation of the detections, and is a function of the image size and the size of the detections. We found this experimentally to be 20 pixels. Also, the threshold for the number of neighborhood occurrences is set to 4. This value can be set based on the total number of images the detection is made, which in this case is 100 images. The output of the function is the number of clusters, and it identifies the number of parking slots. A total number of 26 clusters were identified by the algorithm. In the next step we estimate the final bounding boxes of each class by weighing them with their respective scores. This estimation is achieved by the following equation, where  $*$  represents element-wise multiplications:

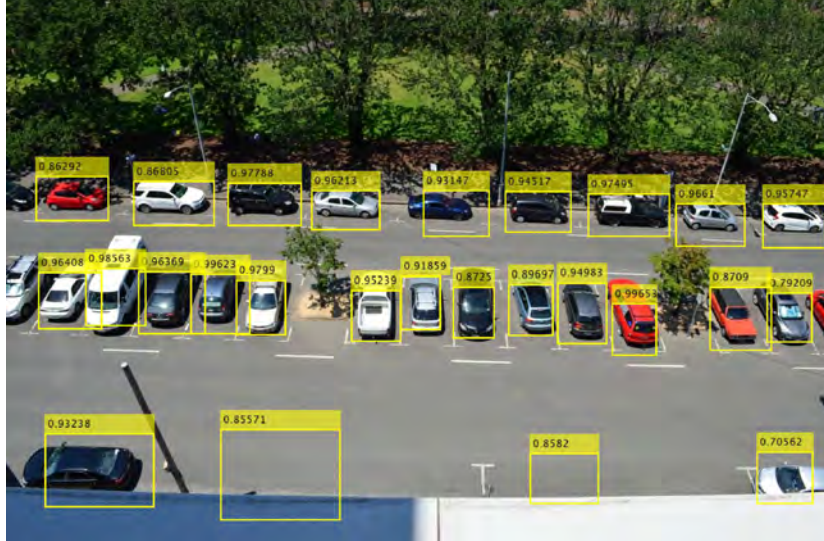
$$Bbox^{class} = \frac{Bbox * BboxScore}{\sum BboxScore} \quad (11.1)$$

where  $Bbox^{class}$  represents the bounding box of a particular class (parking slot),  $Bbox$  represents the array containing all the bounding boxes of the particular class, and  $BboxScore$  is the matrix containing the respective scores of the  $Bbox$ . This is achieved in the following lines of code:

```
1 % Estimate the bounding boxes of each class as the parking slot
2 classifiedMean(n,:) = [(classified{n}(:,1))*classifiedScore{n}) ...
3 .
4 .
5 (classified{n}(:,4))*classifiedScore{n})/(sum(classifiedScore{n}))];
```

Figure 11.14 shows the bounding boxes of each classes with their respective average scores, and Figure 11.15 shows the average precision of the detections. The average precision with 50% IoU (also refereed as AP50) is 37%. This low average precision is due to a couple of problems. Firstly, the sizes of the detections are not uniform, for instance see the third row of Figure 11.14. Secondly, the lengths of the detected parking slots are smaller as compared to the depicted parking slots, as the detector detects vehicles that are smaller than the parking slots. Therefore, to improve the delineations of the slots we further post-process the detections.

We start by calculating the average length and width of the parking slots in the above lines of code. We interchange the length and the width for the slots whose



**Figure 11.14:** Bounding boxes derived from each cluster representing individual parking slots.

aspect ratio was less than 1. Figure 11.16 shows the visualization of the parking slots. Later, we calculate the average length and width of the parking slots to be 80 x 60 pixels, and therefore, the average aspect ratio to be 1.33. Next, as per our assumption (in Section 11.4.1), we increase the length of the parking slots by 20%, and therefore, resize all the parking slots to 96 x 60 pixels.

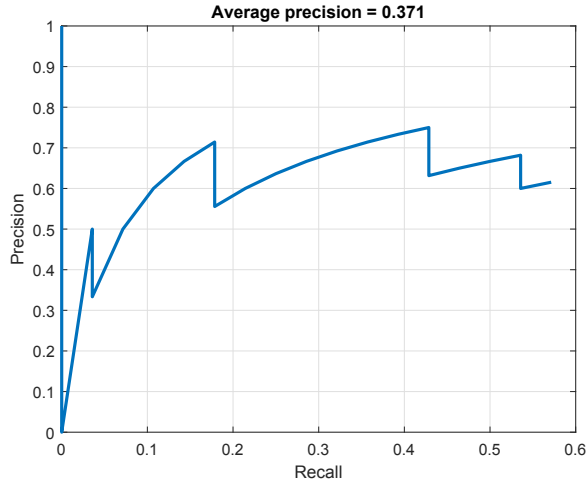
```

1 for n=1:length(classifiedMean2)
2 if (classifiedMean2(n,3)/classifiedMean2(n,4) > (1/w)) % aspect ...
   ratio constraint
3 differenceX = classifiedMean2(n,3) - 1*1.2; % 80% assumption
4 .
5 else
6 .
7 classifiedMean2(n,4) = classifiedMean2(n,4) - differenceY;
8 end
9 end

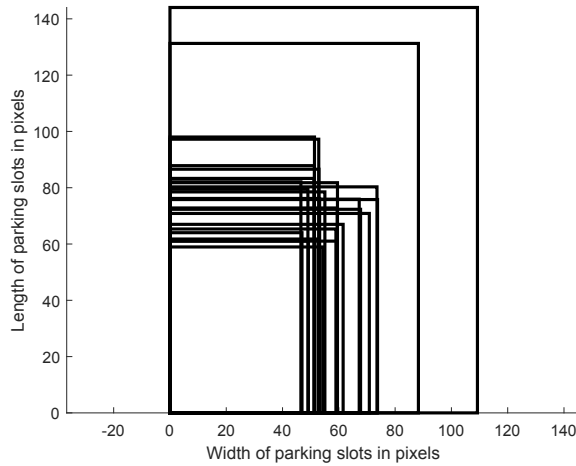
```

Next we plot Figure 11.17 that shows the average precision of the detections after post-processing the bounding boxes. We observe an excellent improvement in the average precision AP50 from 37.1% to 80.4%, which is around 116% improvement. For reference, the mAP50 (mean average precision for multi-class objects) of Faster-RCNN is approximately 59% (Redmon and Farhadi, 2018) with state-of-the-art feature extractor ResNet-101-FPN.



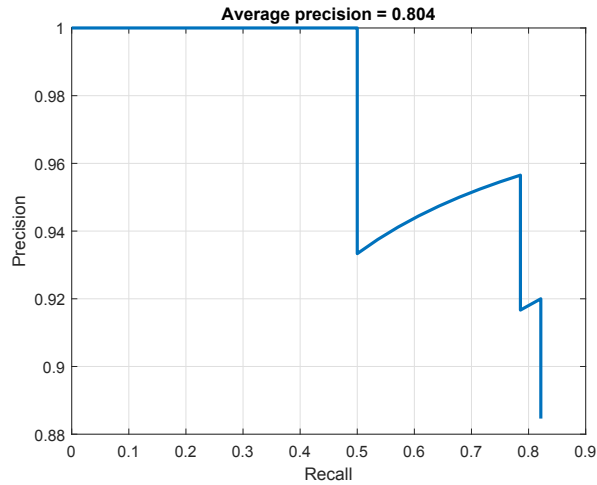


**Figure 11.15:** The average precision of all the bounding boxes delineated by clustering is 0.37.

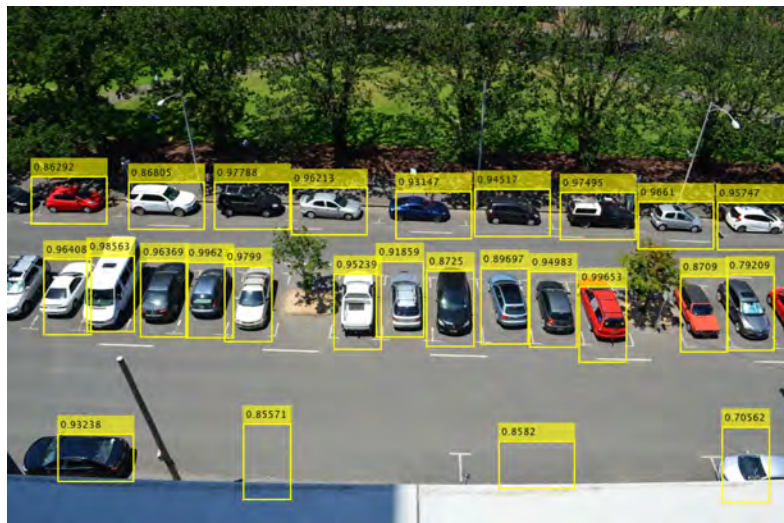


**Figure 11.16:** Visualization of the delineated parking slots after clustering. The length and width of the bounding boxes having an aspect ratio greater than 1 were interchanged.

In the next steps, we visualize the delineated parking slots of Barry Street in Figure 11.18, and we visualize them along with the ground truth bounding boxes in Figure 11.19. In these figures we see that all of the detections in the top and middle row are performed correctly. However, two parking slots in the bottom row of the parking area are missed. These missed detections can be explained on the basis of less parked vehicles in those parking slots that resulted in the

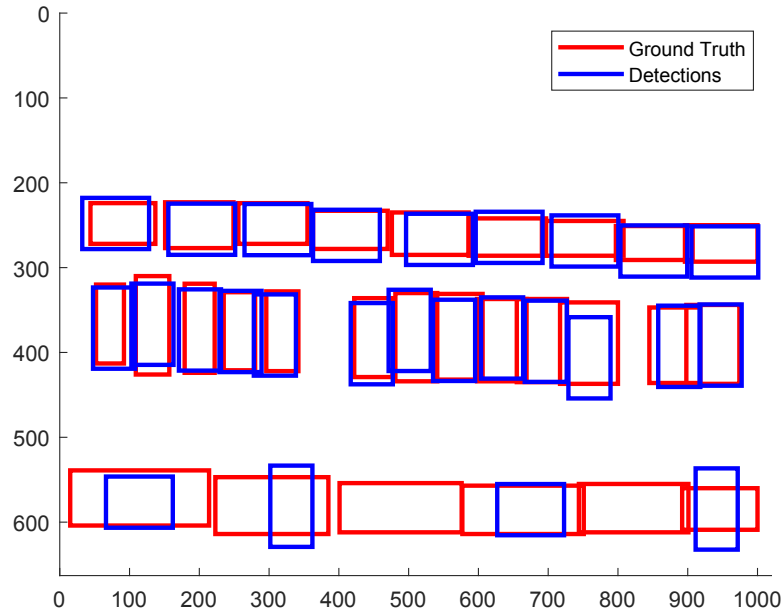


**Figure 11.17:** The average precision of the bounding boxes after post-processing.



**Figure 11.18:** The final delineations of the parking slots after post-processing.

lack of detections in those areas (Figures 11.12 and 11.13). Also, the bounding boxes of two of the detections in the bottom row have inverted dimensions. This anomaly can be explained based on the presence of the wall that occludes the vehicles, and hence the change of the dimensions (and hence the aspect ratios) of the bounding boxes for the detections. The change in the aspect ratio results in the inversion of the bounding box dimensions during the post-processing.



**Figure 11.19:** Visualization of the delineated parking slots and the actual ground truth for the Barry Street dataset.

Therefore, once the parking slots are delineated, we no longer need the computationally expensive Faster-RCNN for the detections. We can directly use the delineations to perform classifications with the method explained in the tutorial in Section 11.3. Therefore, the object detection problem reduces to the problem of image classification only. This is an advantage of the proposed method as compared to the methods that perform parking occupancy detection using Faster-RCNN directly.

### 11.4.3 Applications to Unmarked Open Parking Spaces

The methodology demonstrated in Tutorial 2 can be extended to unmarked open parking spaces as well, or in areas where the parking delineations do not exist (for instance in India). Instead of allocating individual bounding boxes to the parking slots, we could perhaps allocate continuous “parking zones” where the vehicles are most likely to park. We can also have information on the orientation of parking of the vehicles. This continuous parking zone can be broken down according to the standard sizes of the vehicles plus a buffer zone to calculate the number of empty parking slots. Figure 11.20 shows a visualization where 4.5 parking spaces could be identified in parking zone 5 using standard parking slot sizes. Therefore, more exploration in this context is needed and is a promising research direction.



**Figure 11.20:** Visualization of the parking zones for unmarked open parking spaces, showing the areas where the vehicles are likely to park. Using standard vehicle sizes the parking occupancy can be estimated.

#### 11.4.4 Training and Testing Time of Faster-RCNN with ResNet50 Backbone

The fine-tuning process took approximately 5 hours on a NVIDIA Tesla P100 GPU. It is infeasible to perform the fine-tuning process on CPU. However, once the model is fine-tuned it can operate with CPU in approximately 55 seconds (or 0.6 seconds on GPU). Therefore, for the automatic delineation of the parking slots GPU is not mandatory, and the system can run for a couple of hours on CPU to produce the results. This computational overhead can be reduced by using smaller networks like SqueezeNet, however, the accuracy of the detector might be compromised.

### 11.5 Conclusions

This chapter presents two tutorials, one for detecting image-based parking occupancy and the other for automatic delineation of the parking slots. In the first tutorial we fine-tune a pre-trained network on a subset of the publicly available PKLot dataset and checked the generalizing ability of the network by testing with the Barry Street dataset. Also, we provide insights on the training hyper-parameters, training and testing times, and accuracies, and visualize some wrong classifications.

In the second tutorial we demonstrate a novel method to automatically delineate the parking spaces using a state-of-the-art vehicle detector (Faster-RCNN). We fine-tuned Faster-RCNN with a subset of the PKLot data and detected vehicles in the Barry Street images. We combined the detections in multiple frames and performed spatio-temporal analysis of the parking slots to automatically delineate the parking slots. We used a robust density-based clustering algorithm to find the centre of the parking slots, and then weighted the bounding boxes according to the detection scores (confidence). We further post-processed the delineations to improve the detection accuracy and achieve better results than reported in the literature. We conclude that occlusions can effect the detections and can reduce the accuracy of automatic delineations of parking slots. Moreover, the results discussed in this tutorial can be extended to unmarked open parking spaces and points towards an interesting future direction.

## Bibliography

- Acharya, D., Yan, W., and Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. In *Proceedings of the 5th Annual Conference of Research@Locate*, volume 2087, pages 33–40.
- Ahmad, J., Lewis, Z., Duraisamy, P., and McDonald, T. (2019). Parking lot monitoring using mrcnn. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–4. IEEE.
- Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., and Vairo, C. (2017). Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, 72:327–334.
- Amato, G., Carrara, F., Falchi, F., Gennaro, C., and Vairo, C. (2016). Car parking occupancy detection using smart camera networks and deep learning. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 1212–1217. IEEE.
- Cai, B. Y., Alvarez, R., Sit, M., Duarte, F., and Ratti, C. (2019). Deep learning-based video system for accurate and real-time parking measurement. *IEEE Internet of Things Journal*, 6(5):7693–7701.
- de Almeida, P. R., Oliveira, L. S., Britto, A. S., Silva, E. J., and Koerich, A. L. (2015). PKLot – a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937–4949.
- del Postigo, C. G., Torres, J., and Menéndez, J. M. (2015). Vacant parking area estimation through background subtraction and transience map analysis. *IET Intelligent Transport Systems*, 9(9):835–841.

- Ding, X. and Yang, R. (2019). Vehicle and parking space detection based on improved yolo network model. In *Journal of Physics: Conference Series*, volume 1325, page 012084. IOP Publishing.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. *arXiv preprint arXiv:1602.07360*.
- Ichihashi, H., Notsu, A., Honda, K., Katada, T., and Fujiyoshi, M. (2009). Vacant parking space detector for outdoor parking lot by using surveillance camera and FCM classifier. In *2009 IEEE International Conference on Fuzzy Systems*, pages 127–134. IEEE.
- Jung, H. G., Lee, Y. H., and Kim, J. (2009). Uniform user interface for semi-automatic parking slot marking recognition. *IEEE Transactions on Vehicular Technology*, 59(2):616–626.
- Khan, G., Farooq, M. A., Tariq, Z., and Khan, M. U. G. (2019). Deep-learning based vehicle count and free parking slot detection system. In *2019 22nd International Multitopic Conference (INMIC)*, pages 1–7. IEEE.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988.
- Nieto, R. M., García-Martín, Á., Hauptmann, A. G., and Martínez, J. M. (2018). Automatic vacant parking places management system using multicamera vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):1069–1080.
- Paidi, V., Fleyeh, H., and Nyberg, R. G. (2020). Deep learning-based vehicle occupancy detection in an open parking lot using thermal camera. *IET Intelligent Transport Systems*.

- Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99.
- Sairam, B., Agrawal, A., Krishna, G., and Sahu, S. P. (2020). Automated vehicle parking slot detection system using deep learning. In *2020 Fourth International Conference on Computing Methodologies and Communication (IC-CMC)*, pages 750–755. IEEE.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Suhr, J. K. and Jung, H. G. (2013). Sensor fusion-based vacant parking slot detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):21–36.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.
- True, N. (2007). Vacant parking space detection in static images. *University of California, San Diego*, 17:659–662.
- Valipour, S., Siam, M., Stroulia, E., and Jagersand, M. (2016). Parking-stall vacancy indicator system, based on deep convolutional neural networks. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 655–660. IEEE.
- Vítek, S. and Melničuk, P. (2018). A distributed wireless camera system for the management of parking spaces. *Sensors*, 18(1):69.
- Zhang, L., Huang, J., Li, X., and Xiong, L. (2018). Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset. *IEEE Transactions on Image Processing*, 27(11):5350–5364.

## 12 Reducing Parking Pressure by Sharing Resources

YAOLI WANG

### Abstract

This chapter describes a scenario where ridesharing is introduced in urban parking to relieve the pressure of finding a parking site in the city center. A significant amount of time is wasted in cruising for a parking lot according to both life experience and research findings. Although a few policies and strategies have been tested, the middle ground between individual flexibility and reduced travel demand is not yet well accommodated. Therefore, I report of a joint model of ridesharing and parking: people drive from their front doors to a satellite parking site to share rides, and travel to a similar destination in the city center so that parking demand is reduced.

### Keywords

Ridesharing, smart parking, travel behavior

### 12.1 Current Situation and Issues of Urban Parking

Cities are confronted with serious parking problems. Especially in countries with high population density, for example, in the metropolises of China, India, and Japan parking is always a challenge. In the past decades, a couple of countries, especially in Asia, with a swift growth of economy, have gone through a surge of cars on the roads. In India, the amount of cars grew from 55 millions to 210 millions within the time period between 2001 and 2015 (Parmar et al., 2020). The policy to encourage car purchasing in Beijing after the SARS outbreak in 2003 has changed the travel behaviors of Beijing residents, which triggered the world-known challenge of Beijing's traffic congestion. People started to chase after private transport afterwards, which is also assumed as a label of higher social status and as higher travel flexibility.

At least two issues emerge with higher car ownership: limited parking space and wasted cost in searching for parking. Crowded parking space is normal in Chinese neighborhoods. Sometimes roads may be so occupied and narrow that a driver cannot park the car without scratching other cars if no second person is helping to guide the driver's operation. If coming back home late, residents may have to call their neighbors to make some space for their cars. To avoid



the trouble, some people place scooters, bikes, or miscellaneous stuff to take a parking space. Within neighborhoods of higher parking demand people have to pay about 200,000 CNY (30,000 USD) for private parking permits in addition to the parking fee. In Australia, parking space in a city center for a whole day can be as expensive as a parking fine<sup>1</sup>. Hence, globally households are taking increasing resources from public parking space.

For the other issue, cruising for a parking lot is time-consuming. The time searching for parking at the urban center can increase exponentially with the occupancy rate of parking spaces (Millard-Ball et al., 2014). The wasted time and fuel adds significantly to the costs of travel (Shoup, 2006). On the other hand, even though people are aware of the risk of prolonged cruising, they still cruise for on-street parking at the expense of marginal fuel cost to avoid the definitive high fees for off-street parking, which essentially is a trade-off between money and time (Shoup, 2006; Van Ommeren et al., 2012). If a driver is lucky enough to quickly find an empty parking spot, the driver saves. However, if parking space is saturated, even paying for parking cannot solve the issue.

## 12.2 Exploration for Parking Solutions

The battle against the parking issue continues worldwide. Amongst the mutually related parking management strategies mentioned by Litman (2008), three strategies are remote parking, mobility management (e.g., change of transportation mode), and parking pricing, which together yield 10%–30% parking demand reduction.

Mobility management aims to control the amount of cars on the road. To make up for overselling of cars as well as the related issues, the Chinese government carried out innovative policies. Beijing launched car plate restrictions based on the last digit of car plate in 2009, aiming to relieve its notorious traffic congestion<sup>2</sup>. Some households hence decided to buy two cars with different last digits so that they could drive everyday. Then Beijing rolled out a car plate lottery in 2011 which demands that car purchasers join a pool to win a chance of obtaining a car plate. The policy now is inclined to families with no car<sup>3</sup>. If a person buys a new energy car, the buyer does not need to join the lottery but has to queue for it. While the game between general public and policies to some extent retards the growth of car occupancy, the policies cannot solve the issue of parking, and still may deteriorate the situation.

Another strategy explores new technologies to solve the difficulty in sourcing for parking. For example, there are applications on the market that find and

<sup>1</sup><https://bit.ly/3vM12Xb> – Sydney Morning Herald, 8 September 2015

<sup>2</sup><https://bit.ly/2P1gZWj> – Global Times, 17 April 2018

<sup>3</sup><https://bit.ly/393jHBd> – China Daily, 3 June 2020

navigate to available parking spaces<sup>4</sup>, that track availability of parking lots on the fly<sup>5</sup> (Lin et al., 2017), that assist drivers to reserve or retrieve parking spots in real time (e.g., Yan et al., 2008), and that customize the price of parking to vehicles (Ayala et al., 2012). Crowd sensing offers another information source for real-time parking information (Bock and Di Martino, 2017). Dynamic dissemination of parking resources reduces the competition for parking spots (Chai et al., 2017). However, when parking demand is higher than supply, drivers have to wait or cruise for parking anyway.

Accordingly, strategies have been established to mitigate the demand for parking. The extreme case is to take public transport in lieu of driving. Public transport is typically well developed in metropolitan areas. However, if the trip origin and destination differ significantly in population density as well as parking space, e.g., traveling from peripheral suburbs with low public transport accessibility to city center with high parking fee, neither taking public transport nor driving is economic. Therefore, compromises such as Park-and-Ride (P&R) come to the modal mix, where parking is co-located with public transport nodes such as bus stops or railway stations. It saves time for cruising and relieves the over-saturated space for parking in the city center. P&R, however, is neither feasible where public transit is not conveniently developed, nor necessary where parking space is sufficient. Although P&R reduces parking demand effectively, it brings side effects in the meantime. Flexibility has to be compromised by waiting for public transport and driving to specified parking sites.

A second solution to reduce parking demand is ridesharing. Instead of individuals each driving their own car, they can share rides with each other, no matter whether this is by pre-arranged carpooling or real-time ridesharing. The benefit of ridesharing is that parking demand is reduced while it preserves certain flexibility compared with public transport (Carter and O'Connell, 1982). A couple of drawbacks exist in ridesharing as well. Ridesharing is far less popular in reality (Dubernet et al., 2013) than expected based on its proven potential (Tachet et al., 2017). Privacy and trust are significant factors that obstruct people from joining ridesharing. People are less willing to share rides or detour for strangers, nor would they like to disclose the specific location of their homes. Hence, Wang et al. (2017) argue that social acquaintance should be given sufficient consideration in the matchmaking for ridesharing. To protect privacy, launch pads are proposed by (Rigby and Winter, 2015) to disguise the accurate home location. Sharing rides from a public pickup spot is also driven by such motivation (Stiglic et al., 2015).

---

<sup>4</sup><https://bit.ly/3tJcLRS> – Chinese Government, 25 September 2017

<sup>5</sup><https://bit.ly/3vQmR1E> – China Daily, 1 September 2017

## 12.3 Vision of a New Mode: Park-and-Ridesharing

As a solution to the above-mentioned issues, this chapter suggests to share rides from a satellite parking lot, named *Park-and-rideSharing* (P&S). The P&S model incorporates remote parking that relieves parking pressure in the city center and preserves home location privacy, and mobility management in the last leg to the final destination for higher flexibility. There are a few advantages:

- Shared rides reduce parking demand in urban central areas;
- Sharing from a satellite parking protects privacy;
- Shared rides increase flexibility compared with P&R.

To validate the benefit of the model, three scenarios have been compared: driving from home to city center, driving to satellite parking and transferring to a public transit (P&R), and the proposed park-and-ridesharing (P&S). The second scenario is seen in daily life and has been studied (Karamychev and Van Reeve, 2011). This chapter focuses on the pros and cons of the third scenario, trying to provide a new idea to the general public and let the readers explore on their own.

### 12.3.1 The Framework of the Model

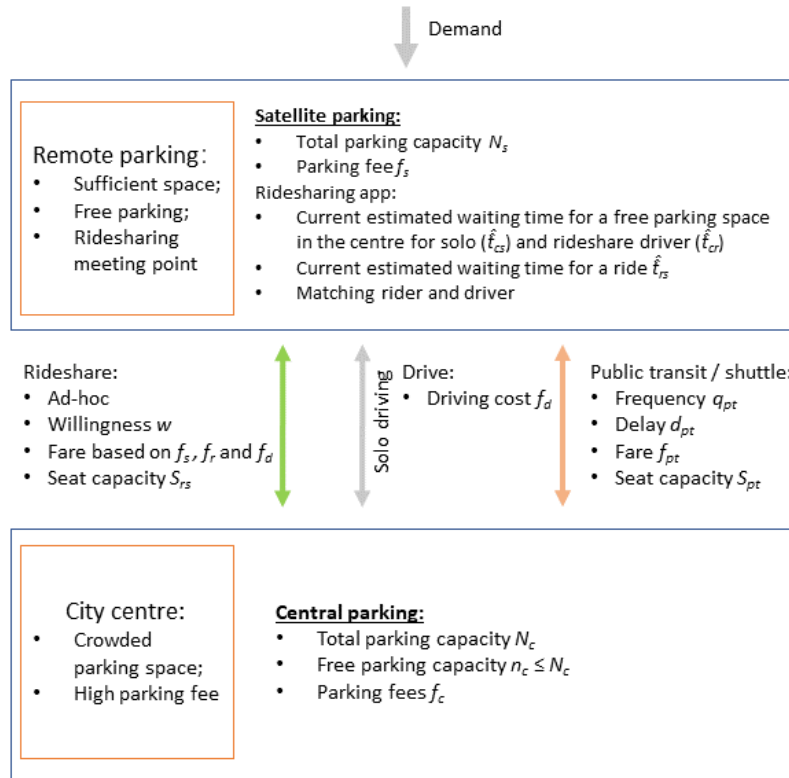
The conceptual model considers a simplified theoretical scenario as illustrated by Figure 12.1 with the factors as demonstrated in Table 12.1. The conceptual model can be extended to real-world scenarios. As a validation of the potential of P&S in Beijing, a study of Beijing with trajectories from all over the city to the Olympic Park is ongoing in the time of writing this chapter.

The scenario consists of a *city center* with a *central parking lot* and an inbound road with a *satellite parking lot*. People are assumed to travel to the city center only to stop at one location for a specific period, and leave the city back to the satellite parking. Not considered is trip chaining, i.e., a travel with a sequence of several destinations.

Travelers only make decisions from their egocentric perspective. Whether or not to share a ride is a trade-off between time and money, with influential factors of looking for parking, time of waiting for rides, money spent for parking, petrol, and so on. Some subjective factors also contribute to the decision as a transformed cost in time or money, e.g., the willingness to join ridesharing. There are four types of travelers in this model:

1. a solo driver with private car,
2. a passenger taking public transit,

3. a ridesharing driver with a private car,
4. a ridesharing passenger.



**Figure 12.1:** The conceptual scenario of the P&S system.

Each type of traveler is associated with a cost function as the utility of being that type. In the baseline scenario where ridesharing is not available, travelers can only choose between being a solo driver or using public transportation to get to the city center.

### 12.3.1.1 Scenario Setting

The *central parking lot* with a capacity  $N_c$  is open for both ridesharing drivers and solo drivers. However, ridesharing drivers are prioritized over solo drivers if queuing for a spot. Ridesharing drivers are rewarded by a reduced central parking fee. The *satellite parking lot* is assumed to be sufficient for the demand. In a real city, any number of satellite parking lots close to the center can be identified, e.g., roadside parking, commercial parking lots, or parking lots at railway or subway stations. Satellite parking lots not only allow people to park but also

**Table 12.1:** Input and output parameters of the model.

Type	Parameters
Input	Traffic flow distribution by hour of day
	Average parking duration
	Parking fees: central parking for solo-driving and ridesharing, and satellite parking
	Maximum capacity at satellite and central parking lots
	Available parking capacity at satellite and central parking
	Public transit travel fare
	Costs for driving (including gas, car usage, congestion charge)
	Maximum capacity of ridesharing vehicle or public transit
	Subjective willingness to do ridesharing
	Public transit frequency
	Public transit delay compared with driving
	Individual value of time, to convert time to monetary cost
Output	Total travel cost as defined by each case study
	Ratios of each traveler type

function as a meeting point for a ride in this scenario. The satellite parking fee, which is either cheaper than central parking or free of charge, costs the same for ridesharing and solo drivers.

The cost of cruising for parking is highly related to traffic flow volume, the time spent in the city, and the searching time for an empty space in the central parking are input parameters. To facilitate decision making, this model considers a smartphone application that retrieves the parking situation on the fly and broadcasts to travelers. Ridesharing vehicles can use the smartphone application to reserve a parking spot before they actually reach the central parking lot. People at the satellite parking are informed of current estimated waiting time for a ride as well as for the average waiting and searching time for parking at the central parking lot to support their decision making.

In case a traveler cannot find a ride outbound from city, the model assumes a public transit system between satellite and central parking lot. This assumption is pragmatic and is taking place in real life. The efficiency of the public transit system is a parameter that affects travelers' preference to ridesharing. The utility of public transit and also the baseline scenarios with no ridesharing available are studied.

The output of the system – the total travel cost – includes three parts: The total time duration for travel converted into monetary unit, the monetary cost of the travel, and the intrinsic willingness cost to ridesharing converted into monetary unit. The total travel time is calculated from making the mode choice at the satellite parking lot to getting back to the car at the satellite parking lot, excluding

the time for activities at the city center. This time includes waiting time for rides or for public transit, travel time for the last leg, and search time for a parking space in the central parking lot (as a solo-driver or as a ridesharing driver). Monetary costs include cost split for ridesharing, cost for solo driving, parking fee and public transit fee. Willingness cost approximates the inconvenience of ridesharing converted to monetary cost, such as psychological uneasiness or time and money spent for cleaning, which only occurs to ridesharing participants.

#### **12.3.1.2 Ridesharing Strategy**

Each driver arriving at the satellite parking calculates their utility and decides to be one of the four types of travelers. The drivers who choose to share rides will join the ridesharing population at the satellite parking lot. Ridesharing is assumed to introduce no extra waiting cost for ridesharing drivers at the satellite parking lot. They only pick up passengers if they are queuing at the satellite parking lot instead of waiting for passengers. The driver re-estimates the utility of each travel mode and picks the second best in the case there is no matched passenger. On the other hand, a passenger (ridesharing or public transport) is constantly recalculating the travel mode utility since the lapse of time affects. Hence they may switch mode and become another type of traveler.

#### **12.3.1.3 Deciding the Travel Mode**

Travelers' choices of travel modes are based on the estimated utility of each mode. Utility functions are decided according to prior knowledge or theories in behavioral economy or other related fields. As aforementioned, it is a comprehensive factor of the input parameters in Table 12.1 and beyond. A traveler in this scenario always chooses the travel mode of minimal cost. Inbound and outbound travel costs are different. Exact outbound costs are difficult to estimate because ridesharing is not guaranteed on the way back. However, travelers do consider the outbound costs when deciding the overall travel mode in the inbound travel. Public transit thus is provided as a backup choice when estimating the cost. This is not to deny the efficiency of public transit. Sometimes public transit can be more efficient than private driving. It is just applied as a baseline mode. On the way back, passengers split costs with drivers as well. Drivers are supposed to pick up passengers who are waiting in the queue. Solo drivers remain to drive alone on their way back for the sake of their subjective willingness.

#### **12.3.1.4 Results**

This P&S scenario has been tested with simulations and is currently being tested by real-world data analysis. A series of simulations with different parameter set-

tings yield indicative findings. Ridesharing has been seen to reduce travel time and travel cost, especially at peak hours when ridesharing is advantageous over other options. Higher volume of travel decreases waiting time for a ride, while it raises the time for cruising if driving alone. Human subjective value of time is converted to money as a factor of behavior utility. There is a sweet spot between spending more time and spending more money. Ostensibly ridesharing also provides resilience to an urban traffic system. The process is a self-adaptive system to mitigate traffic burden when travel demand is high, which brings environmental and social benefits in the meantime. Another issue is the relation between public transit and ridesharing. Although they seem to be competitive at the first glance, they can be supplementary to each other. The decision to do ridesharing is a comprehensive outcome instead of simply travel efficiency. Some people prefer ridesharing intrinsically maybe for a more flexible schedule and maybe for more private space. However, they may hesitate to give up solo driving in the extreme case when a ride cannot be found on the way back. Therefore, a good public transit system ensures travelers to do ridesharing even if the return trip does not have a matched ride.

## 12.4 Looking Forward to the Future

As I have discussed, the future of urban parking is challenging especially in global cities with high population density and high ratio of private car usage. A solution can solve one aspect of a problem, but sometimes may even make the flip side worse. For example, new energy vehicles indeed exude significantly less pollution and greenhouse gas, but the replacement of petrol cars misleads to a perception that a city can tolerate more cars. Consequently parking is likely to be put into a worse situation, or a city is forced to sprawl even more which in turn stimulates people to buy more cars.

Solving the inner-urban parking problem is essentially a change of travel mode and travel behaviors. This chapter reports a conceptual model called *Park-and-Ridesharing*, showing significant potential by simulations. However, its implementation still requires careful planning and testing. Investigation of real world trajectories is conducted as a proof of concept. The empirical analysis has confirmed a significant ratio of trajectories matched for ridesharing. A next step could be to carry out tests in partnership with local governments. For that purpose, many interwoven technical and social challenges should be addressed, e.g., dynamic ridesharing strategies and techniques, social adoption of ridesharing, and real-time parking information updates. Researchers, engineers, and policy makers from multiple fields need to sit together and work that through.

## Bibliography

- Ayala, D., Wolfson, O., Xu, B., DasGupta, B., and Lin, J. (2012). Pricing of parking for congestion reduction. In *Proceedings of the 20th ACM SIGSPATIAL*, page 43.
- Bock, F. and Di Martino, S. (2017). How many probe vehicles do we need to collect on-street parking information? In *International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE.
- Carter, E. C. and O'Connell, K. (1982). Ridesharing element of parking facilities for industrial employment centers. Report DTFH61-80-C-00057, Federal Highway Administration, U.S. Department of Transportation.
- Chai, H., Ma, R., and Zhang, H. M. (2017). Search for parking: A dynamic parking and route guidance system for efficient parking and traffic management. Technical report.
- Dubernet, T., Rieser-Schüssler, N., and Axhausen, K. W. (2013). Using a multi-agent simulation tool to estimate the car-pooling potential. In *92nd Annual Meeting of the Transportation Research Board*, pages 13–0866, Washington, DC.
- Karamychev, V. and Van Reeve, P. (2011). Park-and-ride: Good for the city, good for the region? *Regional Science and Urban Economics*, 41:455–464.
- Lin, T., Rivano, H., and Le Mouel, F. (2017). A survey of smart parking solutions. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–25.
- Litman, T. (2008). Parking management best practices. *ITE Journal on the Web*, 78(September):69–73.
- Millard-Ball, A., Weinberger, R. R., and Hampshire, R. C. (2014). Is the curb 80% full or 20% empty? Assessing the impacts of san francisco's parking pricing experiment. *Transportation Research Part A: Policy and Practice*, 63:76–92.
- Parmar, J., Das, P., and Dave, S. M. (2020). Study on demand and characteristics of parking system in urban areas: A review. *Journal of Traffic and Transportation Engineering (English Edition)*, 7(1):111–124.
- Rigby, M. and Winter, S. (2015). Enhancing launch pads for decision-making in intelligent mobility on-demand. *Journal of Location Based Services*, 9:77–92.
- Shoup, D. C. (2006). Cruising for parking. *Transport Policy*, 13(6):479–486.
- Stiglic, M., Agatz, N., Savelsbergh, M., and Gradisar, M. (2015). The benefits of meeting points in ride-sharing systems. *Transportation Research Part B: Methodological*, 82:36–53.



- Tachet, R., Sagarra, O., Santi, P., Resta, G., Szell, M., Strogatz, S. H., and Ratti, C. (2017). Scaling law of urban ride sharing. *Scientific Reports*, 7:42868.
- Van Ommeren, J. N., Wentink, D., and Rietveld, P. (2012). Empirical evidence on cruising for parking. *Transportation Research Part A: Policy and Practice*, 46:123–130.
- Wang, Y., Winter, S., and Ronald, N. (2017). How much is trust: The cost and benefit of ridesharing with friends. *Computers, Environment and Urban Systems*, 65:103–112.
- Yan, G., Olariu, S., Weigle, M. C., and Abuelela, M. (2008). SmartParking: A secure and intelligent parking system using NOTICE. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*.

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

This chapter is licensed under a Creative Commons Attribution-ShareAlike 4.0 International licence.

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<sup>1</sup> 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<sup>2</sup> 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

---

<sup>1</sup><https://data.melbourne.vic.gov.au/> – City of Melbourne

<sup>2</sup><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 (Kopecký 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  $\lambda_i$  and latitude  $\phi_i$ , 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  $\langle x_j^i, y_j^i, t_j^i, tr_j^i, m_j^i \rangle$ , where  $x_j^i$  is (usually) the longitude,  $y_j^i$  is (usually) the latitude,  $t_j^i$  is the timestamp of the location recording,  $tr_j^i$  is the trip id ( $i$  in this case), and  $m_j^i$  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  $m_j^i \neq m_{j+1}^i$ , 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,  $C = C^1 \cup \dots \cup C^i \cup \dots \cup C^I$  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  $i$ :

$$v_j^i = \frac{\sqrt{(x_j^i - x_{j-1}^i)^2 + (y_j^i - y_{j-1}^i)^2}}{t_j^i - t_{j-1}^i} \quad (13.1)$$

$$a_j^i = \frac{v_j^i - v_{j-1}^i}{t_j^i - t_{j-1}^i} \quad (13.2)$$

$$dr_j^i = \tan^{-1} \frac{y_j^i - y_{j-1}^i}{x_j^i - x_{j-1}^i} - \tan^{-1} \frac{y_{j+1}^i - y_j^i}{x_{j+1}^i - x_j^i} \quad (13.3)$$

where  $2 \leq j \leq J^i$  for  $v_j^i$ ,  $2 \leq j \leq J^i$  for  $dr_j^i$ , and  $3 \leq j \leq J^i$  for  $a_j^i$ .

## 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  $\{C_E \cup C_S\} \subset C$ .

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  $C_S^k$ :

$$\{C_E^i \cup C_S^k\} \subset \mathbf{P}. \quad (13.4)$$



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:

$$C_{CW}^l = C_{WC}^l. \quad (13.5)$$



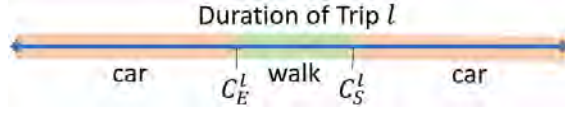


Figure 13.2: Sub-trips and change points of a multiple change point trip  $l$ .

### 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  $\Sigma$  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  $3\sigma$ . 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.

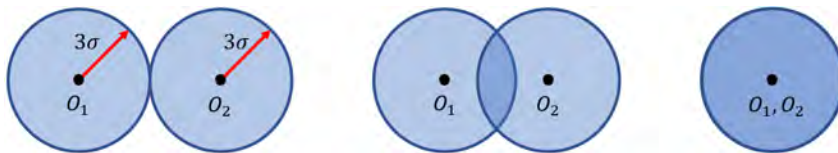


Figure 13.3: Three possible case scenarios for two recorded points with same true latitude and longitude.

### 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<sup>3</sup>: 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  $\sigma$  is not provided in the GeoLife dataset, hence, let us assume this  $\sigma$  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  $\mathbf{P}$  bigger than a point, and  $\mathbf{P}$  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  $\mathbf{P}$  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:

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

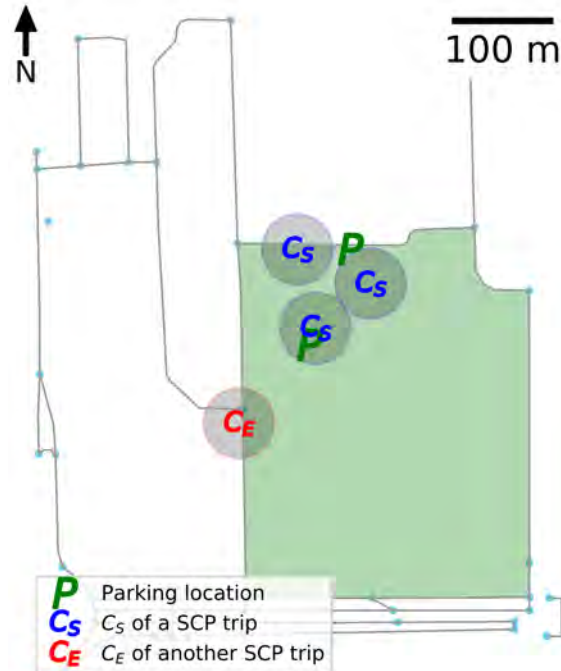
where  $\mathbf{C}(C, 3\sigma)$  represents a circle with center  $C$  and radius  $3\sigma$ .

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  $\sigma^2$  be the measurement error variance for trip  $l$ . Hence,  $\sigma$  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

<sup>3</sup><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  $\sigma$  with  $\approx 98\%$  probability. Thus we get the constraints for validating the same parking spaces recorded twice:

$$C_{WC}^l - C_{CW}^l \leq 6\sigma. \quad (13.7)$$

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

A map is presented in Figure 13.5 to show the true locations of parking spaces as extracted from Google Maps<sup>4</sup> 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.

<sup>4</sup><https://developers.google.com/maps/documentation/api-picker>

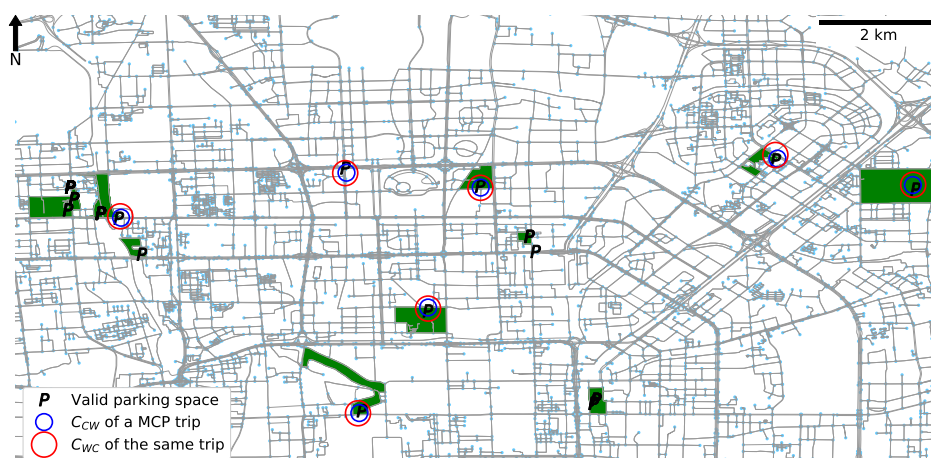


Figure 13.5: Mapped parking spaces using MCP trips of GeoLife trajectory data.

### 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.

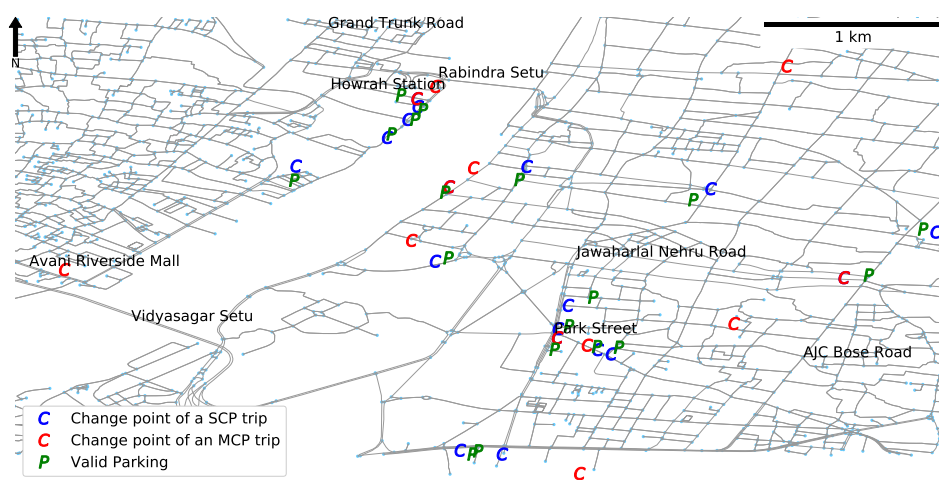


Figure 13.6: Reflection of the model in Kolkata, India.

## 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., Schlenker, 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 (JTPE)*, 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 crowd-sensing 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.
- Kopecký, 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.
- Shoab, 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.



Parking is a challenge for cities everywhere, but especially for cities in low- and middle-income countries. There, cities are experiencing rapid urbanization and increasing motorization, while investment capacity for parking infrastructure is limited, and despite the availability of free on-street parking, it is not used in an efficient and coordinated way.

This book is meant to act as a resource for those managing urban parking challenges, particularly in low- and

middle-income countries. This open-access book can provide immediate guidance to city authorities, engineering firms, and urban planners worldwide and help develop data-driven solutions for smarter cities. The first part of this book portrays geospatial technologies in the context of urban mobility in smart cities. The second part focuses on implementing those technologies in parking management in low and middle-income countries.

ISBN 978-3-85448-045-7



[www.tuwien.at/academicpress](http://www.tuwien.at/academicpress)