Creating Autonomous Vehicle Systems

Second Edition

Synthesis Lectures on Computer Science

The Synthesis Lectures on Computer Science publishes 75–150 page publications on general computer science topics that may appeal to researchers and practitioners in a variety of areas within computer science.

Creating Autonomous Vehicle Systems, Second Edition
Shaoshan Liu, Liyun Li, Jie Tang, Shuang Wu, and Jean-Luc Gaudiot
September 2020

Blockchain Platforms: A Look at the Underbelly of Distributed Platforms
Stijn Van Hijfte
July 2020

Analytical Performance Modeling for Computer Systems, Third Edition
Y.C. Tay
July 2018

Creating Autonomous Vehicle Systems
Shaoshan Liu, Liyun Li, Jie Tang, Shuang Wu, and Jean-Luc Gaudiot
October 2017

Introduction to Logic, Third Edition
Michael Genesereth and Eric J. Kao
November 2016

Analytical Performance Modeling for Computer Systems, Second Edition
Y.C. Tay
October 2013

Introduction to Logic, Second Editio
Michael Genesereth and Eric Kao
August 2013

Introduction to Logic
Michael Genesereth and Eric Kao
January 2013

Science Fiction Prototyping: Designing the Future with Science Fiction
Brian David Johnson
April 2011

Storing Clocked Programs Inside DNA: A Simplifying Framework for Nanocomputing
Jessica P. Chang, Dennis E. Shasha
April 2011

Analytical Performance Modeling for Computer Systems
Y.C. Tay
April 2010

The Theory of Timed I/O Automata
Dilsun K. Kaynar, Nancy Lynch, Roberto Segala, and Frits Vaandrager
2006

Creating Autonomous Vehicle Systems, Second Edition
Shaoshan Liu, Liyun Li, Jie Tang, Shuang Wu, and Jean-Luc Gaudiot

ISBN: 978-3-031-00677-7 paperback
ISBN: 978-3-031-01805-3 ebook
ISBN: 978-3-031-00073-7 hardcover

DOI 10.1007/978-3-031-01805-3

A Publication in the Springer series
SYNTHESIS LECTURES ON COMPUTER SCIENCE, #12

Series ISSN: 1932-1228 Print 1932-1686 Electronic

Creating Autonomous Vehicle Systems

Second Edition

Shaoshan Liu
PerceptIn
Liyun Li
Xpeng Motors
Jie Tang
South China University of Technology
Shuang Wu
YiTu
Jean-Luc Gaudiot
University of California, Irvine

SYNTHESIS LECTURES ON COMPUTER SCIENCE #12

ABSTRACT

This book is one of the first technical overviews of autonomous vehicles written for a general computing and engineering audience. The authors share their practical experiences designing autonomous vehicle systems. These systems are complex, consisting of three major subsystems: (1) algorithms for localization, perception, and planning and control; (2) client systems, such as the robotics operating system and hardware platform; and (3) the cloud platform, which includes data storage, simulation, high-definition (HD) mapping, and deep learning model training. The algorithm subsystem extracts meaningful information from sensor raw data to understand its environment and make decisions as to its future actions. The client subsystem integrates these algorithms to meet real-time and reliability requirements. The cloud platform provides offline computing and storage capabilities for autonomous vehicles. Using the cloud platform, new algorithms can be tested so as to update the HD map—in addition to training better recognition, tracking, and decision models.

Since the first edition of this book was released, many universities have adopted it in their autonomous driving classes, and the authors received many helpful comments and feedback from readers. Based on this, the second edition was improved by extending and rewriting multiple chapters and adding two commercial test case studies. In addition, a new section entitled "Teaching and Learning from this Book" was added to help instructors better utilize this book in their classes. The second edition captures the latest advances in autonomous driving and that it also presents usable real-world case studies to help readers better understand how to utilize their lessons in commercial autonomous driving projects.

This book should be useful to students, researchers, and practitioners alike. Whether you are an undergraduate or a graduate student interested in autonomous driving, you will find herein a comprehensive overview of the whole autonomous vehicle technology stack. If you are an autonomous driving practitioner, the many practical techniques introduced in this book will be of interest to you. Researchers will also find extensive references for an effective, deeper exploration of the various technologies.

KEYWORDS

autonomous driving, driverless cars, perception, vehicle localization, planning and control, autonomous driving hardware platform, autonomous driving cloud infrastructures, low-speed autonomous vehicle, autonomous last-mile delivery vehicle

Contents

Preface to the Second Edition . . . . **xv**

Teaching and Learning from This Book . . . . **xvii**

1 Introduction to Autonomous Driving **1**
- 1.1 Autonomous Driving Technologies Overview 1
- 1.2 Autonomous Driving Algorithms 2
  - 1.2.1 Sensing 2
  - 1.2.2 Perception 3
  - 1.2.3 Object Recognition and Tracking 5
  - 1.2.4 Action 6
- 1.3 Autonomous Driving Client System 8
  - 1.3.1 Robot Operating System (ROS) 8
  - 1.3.2 Hardware Platform 10
- 1.4 Autonomous Driving Cloud Platform 11
  - 1.4.1 Simulation 11
  - 1.4.2 HD Map Production 12
  - 1.4.3 Deep Learning Model Training 13
- 1.5 It Is Just the Beginning 13

2 Autonomous Vehicle Localization **15**
- 2.1 Localization with GNSS 15
  - 2.1.1 GNSS Overview 15
  - 2.1.2 GNSS Error Analysis 16
  - 2.1.3 Satellite-Based Augmentation Systems 17
  - 2.1.4 Real-Time Kinematic and Differential GPS 18
  - 2.1.5 Precise Point Positioning 20
  - 2.1.6 GNSS INS Integration 21
- 2.2 Localization with LiDAR and High-Definition Maps 22
  - 2.2.1 LiDAR Overview 23
  - 2.2.2 High-Definition Maps Overview 25
  - 2.2.3 Localization with LiDAR and HD Map 29
- 2.3 Visual Odometry 33

2.3.1 Stereo Visual Odometry 34
2.3.2 Monocular Visual Odometry 34
2.3.3 Visual Inertial Odometry 35
2.4 Dead Reckoning and Wheel Odometry 36
2.4.1 Wheel Encoders 37
2.4.2 Wheel Odometry Errors 38
2.4.3 Reduction of Wheel Odometry Errors 39
2.5 Sensor Fusion 41
2.5.1 CMU Boss for Urban Challenge 41
2.5.2 Stanford Junior for Urban Challenge 43
2.5.3 Bertha from Mercedes Benz 44
2.6 References 46

3 Perception in Autonomous Driving 51
3.1 Introduction 51
3.2 Datasets 51
3.3 Detection 54
3.4 Segmentation 56
3.5 Stereo, Optical Flow, and Scene Flow 57
3.5.1 Stereo and Depth 57
3.5.2 Optical Flow 59
3.5.3 Scene Flow 59
3.6 Tracking 61
3.7 Conclusion 63
3.8 References 64

4 Deep Learning in Autonomous Driving Perception 69
4.1 Convolutional Neural Networks 69
4.2 Detection 70
4.3 Semantic Segmentation 74
4.4 Stereo and Optical Flow 76
4.4.1 Stereo 76
4.4.2 Optical Flow 78
4.4.3 Unsupervised Learning for Dense Correspondence 81
4.5 Conclusion 83
4.6 References 84

5 Prediction and Routing **87**
5.1 Planning and Control Overview 87
5.1.1 Architecture: Planning and Control in a Broader Sense 87
5.1.2 Scope of Each Module: Solve the Problem with Modules 89
5.2 Traffic Prediction 92
5.2.1 Behavior Prediction as Classification 93
5.2.2 Vehicle Trajectory Generation 97
5.3 Lane Level Routing 100
5.3.1 Constructing a Weighted Directed Graph for Routing 101
5.3.2 Typical Routing Algorithms 103
5.3.3 Routing Graph Cost: Weak or Strong Routing 107
5.4 Conclusions 108
5.5 References 108

6 Decision, Planning, and Control **111**
6.1 Behavioral Decisions 111
6.1.1 Markov Decision Process Approach 113
6.1.2 Scenario-Based Divide and Conquer Approach 115
6.2 Motion Planning 123
6.2.1 Vehicle Model, Road Model, and SL-Coordination System 124
6.2.2 Motion Planning with Path Planning and Speed Planning 126
6.2.3 Motion Planning with Longitudinal Planning and Lateral Planning 133
6.3 Feedback Control 137
6.3.1 Bicycle Model 137
6.3.2 PID Control 139
6.4 Conclusions 140
6.5 References 141

7 Reinforcement Learning-Based Planning and Control **143**
7.1 Introduction 143
7.2 Reinforcement Learning 145
7.2.1 Q-Learning 147
7.2.2 Actor-Critic Methods 151
7.3 Learning-Based Planning and Control in Autonomous Driving 153
7.3.1 Reinforcement Learning on Behavioral Decision 154
7.3.2 Reinforcement Learning on Planning and Control 154

7.4 Conclusions 157
7.5 References 157

8 Client Systems for Autonomous Driving 159
8.1 Autonomous Driving: A Complex System 159
8.2 Operating System for Autonomous Driving 161
  8.2.1 ROS Overview 161
  8.2.2 System Reliability 163
  8.2.3 Performance Improvement 164
  8.2.4 Resource Management and Security 165
8.3 Computing Platform 165
  8.3.1 Computing Platform Implementation 166
  8.3.2 Existing Computing Solutions 166
  8.3.3 Computer Architecture Design Exploration 167
8.4 References 171

9 Cloud Platform for Autonomous Driving 173
9.1 Introduction 173
9.2 Infrastructure 173
  9.2.1 Distributed Computing Framework 175
  9.2.2 Distributed Storage 175
  9.2.3 Heterogeneous Computing 176
9.3 Simulation 177
  9.3.1 BinPipeRDD 178
  9.3.2 Connecting Spark and ROS 179
  9.3.3 Performance 180
9.4 Model Training 180
  9.4.1 Why Use Spark? 181
  9.4.2 Training Platform Architecture 182
  9.4.3 Heterogeneous Computing 183
9.5 HD Map Generation 183
  9.5.1 HD Map 184
  9.5.2 Map Generation in the Cloud 185
9.6 Conclusions 186
9.7 References 187

10 Autonomous Last-Mile Delivery Vehicles in Complex Traffic Environments . . . 189
10.1 Background and Motivations 189

10.2 Autonomous Delivery Technologies in Complex Traffic Conditions 190
10.3 JD.com: An Autonomous Driving Solution 192
10.3.1 Autonomous Driving Architecture 192
10.3.2 Localization and HD Map 193
10.3.3 Perception 195
10.3.4 Prediction, Decision, and Planning 196
10.4 Safety and Security Strategies 199
10.4.1 Simulation-Level Verification 200
10.4.2 Vehicle-End Monitoring 200
10.4.3 Remote Monitoring 200
10.5 Production Deployments 200
10.6 Lessons Learned 201
10.7 References 202

11 PerceptIn's Autonomous Vehicles Lite 203
11.1 Introduction 203
11.2 Expensive Autonomous Driving Technologies 204
11.2.1 Sensing 204
11.2.2 Localization 205
11.2.3 Perception 205
11.2.4 Decision Making 206
11.2.5 HD Map Creation and Maintenance 206
11.2.6 System Integration 207
11.3 Achieving Affordability and Reliability 207
11.3.1 Sensor Fusion 208
11.3.2 Modular Design 209
11.3.3 High-Precision Visual Map 210
11.4 Deploying Autonomous LSEV for Mobility as a Service 210
11.5 Conclusions 212
11.6 References 212

Author Biographies 215

Preface to the Second Edition

Autonomous vehicles—be they on land, on water, or in the air—are upon us and are finding a myriad of new applications, from driverless taxi services to automatic airborne surveillance of sensitive remote areas. Continued technological advances in the past few decades have made these innovations possible, but the design problems that must be surmounted in order to provide useful, efficient, and, supremely importantly, safe operations of these independent units are equally numerous and daunting.

It is thus the purpose of this book to provide an overview of these problems and lead the reader through some common design solutions. High technological capabilities, complete integration of hardware and software, and deep synergy with resident platforms (such as cloud servers) are a must for an eventual successful deployment. The focus of the book is on land vehicles and, more specifically, cars in urban or country road environments, as well as off-road operations. The aim of this book is to address an audience of engineers, be they from the academic or the industrial side, with a survey of the problems, solutions, and future research issues they will encounter in the development of autonomous vehicles, from sensing, perception, all the way to action, and including support from cloud-based servers. A copious amount of bibliographic references completes the picture and will help the reader navigate through a jungle of past work.

The first edition of this book was very well received, as many universities adopted it in their autonomous driving classes, and many companies utilized it for their internal trainings (more details on this can be found in the new chapter "Teaching and Learning from this Book"). Also, we are very fortunate to have received a lot of helpful comments and feedbacks from the first edition readers, which greatly helped improve the content and the quality of the second edition. Specifically, based on these comments and feedback, we made the following improvements in the second edition. First, we extended Chapters 3 and 4 to provide the latest updates on perception techniques; second, we rewrote Chapters 5, 6, and 7 to make them easier to understand; third, we added two commercial case studies in Chapters 10 and 11, so that readers can easily understand how to apply their learnings in real-world environments. In addition, we added a section titled "Teaching and Learning from this Book" to help instructors better utilize this book in their classes. We believe that the second edition captures the up-to-date advances in autonomous driving, and also presents practical real-world case studies to help readers utilize their learnings in commercial autonomous driving projects.

STRUCTURE OF THE BOOK

A brief history of information technology and an overview of the algorithms behind autonomous driving systems, of the architecture of the systems, and of the support infrastructure needed is provided in Chapter 1. Localization, being one of the most important tasks in autonomous driving, is covered in Chapter 2, where the most common approaches are introduced. The principles, advantages, and drawbacks of GNSS, INS, LiDAR, and wheel odometry are described in detail, and the integration of various versions of these strategies are discussed. As for detection, i.e., "understanding" the environment based on sensory data, it is described in Chapter 3, with an exploration of the various algorithms in use, including scene understanding, image flow, tracking, etc. The large datasets, highly complex computations required by image classification, object detection, semantic segmentation, etc., are best handled by the deep learning approaches to perception advocated for in Chapter 4, where applications to detection, semantic segmentation, and image flow are described in detail. Once the environment is understood by the autonomous vehicle, it must somehow predict future events (e.g., the motion of another vehicle in its vicinity) and plan its own route; this is the purpose of Chapter 5. Next, in Chapter 6, an even more detailed level of decision making, planning, and control is presented. Feedback between modules with possibly orthogonal decisions as well as conflict resolution (e.g., one module could recommend a lane change, but another one has detected an obstacle in the lane in question) are covered with an emphasis on describing algorithms for behavioral decision making (e.g., Markov decision processes, scenario-based divide and conquer), and for motion planning. This leads into Chapter 7 for a demonstration of the need to supplement the design with Reinforcement Learning-based Planning and Control for a complete integration of situational scenarios in the development of an autonomous system. Underneath it all, the on-board computing platform is the topic of Chapter 8. It includes an introductory description of the Robot Operating System, followed by an actual summary of the real hardware employed. The need for heterogeneous computing is introduced with a strong emphasis on meeting real-time computing requirements as well as on-board considerations (power consumption and heat dissipation). This means that a variety of processing units (general-purpose CPU, GPUs, FPGAs, etc.) must be used. Chapter 9 covers the infrastructure for the cloud platform used to "tie it all together" (i.e., provide services for distributed simulation tests for new algorithm deployment, offline deep learning model training, and High-Definition (HD) map generation). Chapter 10 presents a case study of a commercial autonomous last-mile delivery vehicle operating in complex traffic environments. Finally, Chapter 11 presents a case study of affordable autonomous vehicles for microtransit services.

Teaching and Learning from This Book

1. INTRODUCTION

In recent years, autonomous driving has become quite a popular topic in the research community as well as in industry. However, the biggest barrier to the rapid development of this field is a very limited talent supply. This is due to several problems: first, autonomous driving is the complex integration of many technologies, making it extremely challenging to teach; second, most existing autonomous driving classes focus on one technology of the complex autonomous driving technology stack, thus failing to provide a comprehensive introduction; third, without good integration experiments, it is very difficult for the students to understand the interaction between different technology pieces.

To address these problems, we have developed a modular and integrated approach to teach autonomous driving. For students interested in autonomous driving, this book provides a comprehensive overview of the whole autonomous vehicle technology stack. For practitioners, this book presents many practical techniques and a number of references to aid them performing an effective, deeper exploration of particular modules. In addition, to help the students understand the interaction between different modules, we developed platforms for hands-on integration experiments. Our teaching methodology starts with an overview of autonomous driving technologies, followed by different technology modules, and ends with integration experiments. Note that the order of the modules can be flexibly adjusted based on the students' background and interest level. We have successfully applied this methodology to three different scenarios: an introduction to an autonomous driving class for undergraduate students with limited technology background; a graduate level embedded systems class, in which we added a session on autonomous driving; as well as a two-week professional training for seasoned engineers.

The rest of this overview is organized as follows. Section 2 reviews existing classes on autonomous driving; Section 3 presents the details of the proposed modular and integrated teaching methodology; Section 4 presents the three case studies where we applied the proposed methodology; and we draw the conclusions in Section 5.

2. EXISTING CLASSES ON AUTONOMOUS DRIVING

Autonomous driving has been attracting attention from academia as well as industry. However, comprehensive and complex autonomous driving systems involve a very diverse set of technologies including sensing, perception, localization, decision making, real-time operating system, heterogeneous computing, graphic/video processing, cloud computing, etc. This sets high requirements for lecturers to master all aspects of relative technologies. It is even more of a challenge for students to understand the interactions between these technologies.

Problem-based learning (PBL) is a feasible and practical way to teach relative knowledge and technologies of autonomous driving [20]. Costa et al. built a simulator by integrating Gazebo 3D simulator for students to design and understand autonomous driving systems [13]. In terms of the educational methodology on Learning from Demonstration, Arnaldi et al. proposed an affordable setup for machine learning applications of autonomous driving by implementing embedded programming for small autonomous driving cars [14]. However, these approaches usually cover only one or two technologies, such as machine learning, and fail to provide a comprehensive understanding of the whole system.

Several major universities already offer autonomous-driving-related classes. For instance, MIT offers two courses on autonomous driving. The first one focuses on Artificial Intelligence and is available online to the public and registered students. This course invites several guest speakers on the topic of deep learning, reinforcement learning, robotics, psychology, etc. [15]. The other concentrates on deep learning for self-driving cars and teaches deep learning knowledge by building a self-driving car [16]. Stanford also offers a course for introducing the key artificial intelligence technologies that could be used for autonomous driving [17]. However, these classes all focus on machine learning and do not provide good coverage of the different technologies involved in autonomous driving. It is thus hard for students to obtain a comprehensive understanding of autonomous driving systems.

In terms of experimental platforms and competition developments, Paull et al. proposed Duckietown, an open-source and inexpensive platform for autonomy education and research [18]. The autonomous vehicles are equipped with a Raspberry Pi 2 and a monocular camera for sensing. In addition, there are several competitions of autonomous driving for a broad range of students to stimulate their passion and encourage learning about key technologies in autonomous cars and related traffic systems [19, 21–23]. However, to use these platforms and to enter these competitions, students need to first gain a basic understanding of the technologies involved as well as their interactions; this type of education is currently lacking.

As seasoned autonomous driving researchers and practitioners, we think the best way to learn how to create autonomous vehicle systems is to first grasp the basic concepts in each technology module, then integrate these modules to understand how they interact. Existing autonomous driving classes either focus on only one or two technologies, or have the students directly build a working autonomous vehicle. As a result, this disconnection between individual technology module and system integration creates a high entry barrier for students interested in autonomous driving and often intimidates interested students away from entering this exciting field. To address this exact problem, we designed our modular and integrated approach, which we will present in this chapter, along with sharing our experiences with autonomous driving education.

3. A MODULAR AND INTEGRATED TEACHING APPROACH

Autonomous driving involves many technologies, thus making autonomous driving education extremely challenging. To address this problem, we propose a modular and integrated teaching methodology for autonomous driving. To implement this methodology, we have developed modular teaching materials including a textbook and a series of multimedia online lectures, as well as integration experimental platforms to enable a comprehensive autonomous driving education.

3.1 TEACHING METHODOLOGY

In the past several years, we have taught undergraduate- and graduate-level classes as well as initiated new engineers to the concepts of autonomous driving. A common problem that we found was that the first encounter on this subject usually intimidated many students because of its perceived complexity. Similarly, even experienced engineers newly exposed to the field of autonomous driving felt it was extremely stressful (in other words, it took them outside their comfort zone), especially since the subject touches upon many new areas.

On the other hand, we found that a modular and integrated approach is an effective way of teaching autonomous driving. This means that we first break the complex autonomous driving technology stack into modules and have the students start with the module with which they are most familiar, then have them move on to other modules. This allows the students to maintain a high level of interest and make satisfactory progress throughout the learning process. Once the students have gone through all the modules, they are challenged to perform a few integration experiments to understand the interactions between these modules. Aside from being an effective teaching method, this approach allows the instructors to flexibly adapt the class curriculum to the needs of students with different technology backgrounds, including undergraduate students with little technology background, graduate students with a general computer science technology background, and seasoned engineers who are experts in a particular field.

Figure 1 illustrates the proposed modular and integrated teaching approach: the class is divided into nine modules and integration experiments. Undergraduate and graduate students can both start with a general overview of autonomous driving technologies, but the undergraduate students may need more time to understand the basics of the technologies involved. They can then move on to localization, followed by traditional perception and perception with deep learning. Next, they can learn about the decision-making pipeline, including planning and control, motion planning, as well as end-to-end planning. Once the students are done with these, they can delve into client systems and cloud platforms. Finally, they can perform integration experiments to understand the interactions between these modules.

On the other hand, seasoned engineers with embedded system backgrounds can start with the general overview and then directly move on to the client systems module and get familiar with the new materials from the perspective of their comfort zone, thus allowing them to maintain a high interest level. They can then move on to the cloud platform, which also focuses on system design, and still stay within their comfort zone. Once they master these modules, they are equipped with enough background knowledge and confidence to learn the rest of the modules.

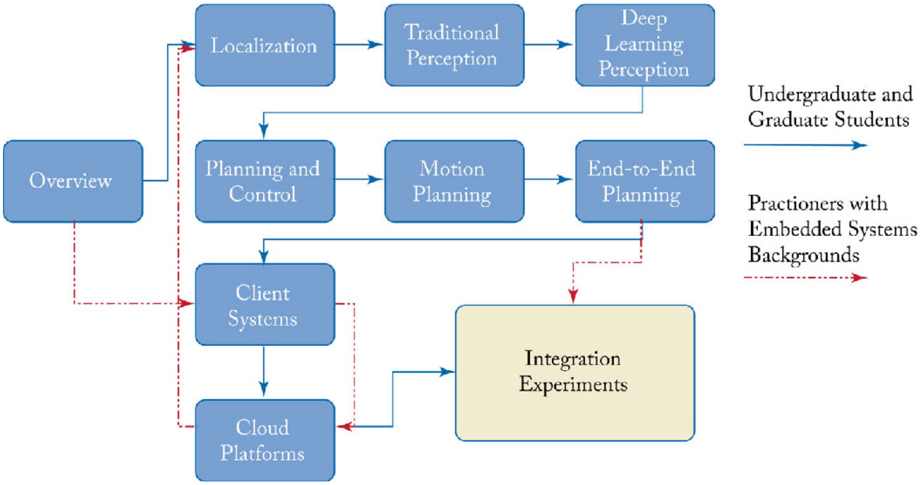

Figure 1: Example of our modular and integrated teaching approach.

3.2 MODULAR TEACHING MATERIALS

First, to cover all the major modules in the autonomous driving technology stack, we developed the current book, *Creating Autonomous Vehicle Systems*, now in its second edition. This is one of the first technical overviews of autonomous vehicles where we share our practical experiences creating autonomous vehicle systems. This book consists of nine chapters that provide an overview of autonomous vehicle systems followed by descriptions of localization technologies, the traditional techniques used for perception, deep learning-based techniques for perception, the planning and control sub-system (especially prediction and routing technologies) motion planning and feedback control for the planning and control subsystem, the reinforcement learning-based planning and control, the details of client systems design, and the details of cloud platforms for autonomous driving.

This book is aimed at students, researchers, and practitioners alike. For undergraduate or graduate students interested in autonomous driving, this book provides a comprehensive overview of the whole autonomous vehicle technology stack. For autonomous driving practitioners, it presents many practical techniques in implementing autonomous driving systems. It provides researchers with many references for an effective, deeper exploration of the various technologies.

Along with this book, in cooperation with IEEE Computer Society and O'Reilly, we have developed a series of online lectures to introduce each module [5, 6]. Along with the multimedia presentations, this allows students to easily acquire an in-depth understanding of a specific technology.

3.3 INTEGRATION EXPERIMENTAL PLATFORMS

A common problem of autonomous driving education is the lack of experimental platforms. In most autonomous driving classes, people use simulators to verify the performance of newly developed algorithms [11]. Nonetheless, the simulation approach fails to provide an environment for students to understand the interaction between different modules. On the other hand, using an autonomous driving vehicle as an experimental platform is not practical due to the cost, as a demonstration autonomous vehicle can easily cost over $800,000.

A straightforward way to perform integration experiments is to use mobile platforms, such as mobile phones. Nowadays, mobile phones usually consist of many sensors (GPS, IMU, cameras, etc.) as well as powerful heterogeneous computing platforms (with CPU, GPU, and DSP). Therefore, mobile phones can be used as a great integration experimental platform for localization and perception tasks. For instance, as demonstrated in the video "Perceptin Robot System Running on a Cell Phone [8], we have successfully implemented real-time localization, obstacle detection and avoidance, as well as planning and control functions on a Samsung Galaxy 7 mobile phone to drive a mobile robot at 5 miles per hour.

4. CASE STUDIES

Several institutes and universities already have adopted or are in the process of adopting the teaching methodology as well as the experimental platforms described in this chapter. In this section, we present three case studies of applying the aforementioned teaching methodology and materials: the first one is an introduction to an autonomous driving class for undergraduate students with limited technology backgrounds; the second is a graduate-level embedded system class, in which we added a session on autonomous driving; and the last is a two-week professional training for seasoned engineers. These three case studies were carefully selected to demonstrate the flexibility of using the proposed approach to teach autonomous driving.

4.1 INTRODUCTION TO AUTONOMOUS DRIVING CLASS

The introduction to autonomous driving class for undergraduate and graduate students we have developed consists of 15–20 lectures depending on the length of the quarter or semester. Also, a 20-hour experiment session is required for integration experiments. Its overall purpose is to provide a technical overview on autonomous driving for students with basic experience in programming, algorithms, and operating systems.

Due to their limited background, we do not expect students to fully master all the modules, but we intend to maintain their interest level and equip them with the basic knowledge to delve into the modules in which they are particularly interested. To achieve this, we follow the approach shown in Figure 1 and divide the class into nine modules. To maintain a high level of interest, at the beginning of each session, we play a short video, such as "Creating Autonomous Vehicle Systems" [5], to provide a summary and demo of the technologies discussed. Then we move on to the details regarding of implementation of each technology. Also, throughout the class, we have the students use mobile phones to perform integration experiments. Specifically, for localization experiments, we first have the students extract real-time GPS localization data; then we have them improve their localization data by fusing IMU data with GPS data. For perception, we have the students install a deep learning framework, such as MXNET [12], onto their mobile phones and run simple object detection networks.

In addition, we developed multiple integration experiments for the students to gain a deep understanding of the interaction between different modules. Since it is an introductory class, for the advanced integration experiments, such as fusing GPS, IMU, and camera data to provide accurate location updates in real time, we did not provide enough technical background in the lectures. Thus, in order to accomplish these tasks, students not only need to perform their own research to get enough technical background but also spend significant time and effort to perform the experiments. Indeed, we did not expect any student to be able to accomplish the advanced tasks. To our surprise, 8% of the students were able to successfully accomplish these advanced tasks. These observations demonstrate that with a modular and integrated teaching approach, students not only obtain a comprehensive overview of the technologies but are also able to delve into the modules of their interests and become experts.

4.2 ADDING AUTONOMOUS DRIVING MATERIALS TO EMBEDDED SYSTEMS CLASS

We have also added a session in an existing graduate-level embedded systems class to explore how embedded systems technologies can be integrated in autonomous driving systems. The embedded systems class runs over a 20-week semester with 60 hours of lectures and 20 hours of experiments, in which we allocated 6 hours of lectures and 10 hours of experiments for autonomous driving contents. Interestingly, before we started this class, we checked with the students on whether they would start their engineering career in autonomous driving. Most students were highly interested but at the same time feared that autonomous driving was too complicated for them.

With the hope of easing the students' apprehension toward the design of autonomous driving systems, we placed the autonomous driving session at the end of the class, after the students grasped the basic skills of designing and implementing embedded systems with different software and hardware optimization techniques, such as heterogeneous computing. Prior to starting this case study session, all students' understanding of autonomous driving was limited to the conceptual level. Out of the 56 students enrolled, only 10 students were able to list some technologies involved in autonomous driving, such as localization and perception, but none understood the details of these technologies.

Due to the limited time available in the session, we first presented an overview on autonomous driving and then focused on two modules: localization and perception. We delved into two simple algorithm implementations, ORB-SLAM [9] for localization and SqueezeNet [10] for object detection. Next, we placed the students into groups of four to perform integration experiments with these algorithms on their Android cell phones and had them compare the performance of using CPU only vs. the performance of using heterogeneous computing components such as GPU and DSP. After completing the project, the students were asked to summarize their design choices and present their results in class. The presentations helped them understand the techniques used by other groups and they could learn from each other through the presentations.

The results were encouraging. First, it was very interesting to see different optimization strategies from different groups. Some groups prioritized computing resources for localization tasks to guarantee frequent position updates, whereas other groups prioritized computing resources for perception to guarantee real-time obstacle avoidance. Second, through this session, the apprehension toward autonomous driving went away, and the after-class survey indicated that 85% of the students would like to continue learning autonomous driving.

4.3 PROFESSIONAL TRAINING

For autonomous driving companies, one of the biggest challenges is the difficulty of recruiting autonomous driving engineers since there is only a very limited talent pool with autonomous driving experience. Therefore, it is crucial to develop a professional training session to quickly equip seasoned engineers with the technical knowledge to delve into one module of autonomous driving.

We worked closely with an autonomous driving company to quickly bring their engineers, most of whom have embedded systems and general software engineering backgrounds, up to speed. The challenges were three-fold: first, the training session was only two weeks long, not enough time to delve into the technology details; second, in this time span, we needed to place the engineers into different engineering roles, although they came from similar technology background; and third, confidence was a big issue for these engineers as they were concerned as to whether they could get a handle on the complexity of autonomous driving in a short amount of time.

To address these challenges, following the methodology presented in Figure 1, we had the engineers all start with the technology overview, followed by the client systems and the cloud platforms modules during the first week. Since the engineers came from embedded systems and general software engineering backgrounds, they were quite comfortable starting with these modules. Through system modules, they learned about the characteristics of different workloads as well as how to integrate them on embedded and cloud systems. In the second week, based on the engineers' performance in the first week as well as their interest levels toward different technologies, we assigned them to dig deeper into a specific module, such as perception, localization, or decision making.

For integration experiments, unlike in the undergraduate or graduate classes, the engineers were given the chance to work on an actual product after two weeks of training. In this training session, seven engineers were successfully added to the team; one was assigned to the sensing team, two were assigned to the perception team, two were added to the localization team, and two were assigned to the decision-making team. A demo of the development process is shown in the video "Perceptin Autonomous Vehicle Development" [7].

5. CONCLUSION

We are often asked what the most important technology in autonomous driving is. Our answer is always integration. As mentioned above and stressed here again, autonomous driving is not one single technology but rather a complex system integrating many technologies. However, before integration happens, one has to understand each technology module involved. Existing autonomous driving classes often focus on one or two technology modules, or directly have the students build a working autonomous vehicle, thus creating high entry barriers for students. Surprisingly, most students are indeed highly interested in autonomous driving, but it is the fear that they cannot handle the complexities involved that often drives them away.

To address this problem, we developed this modular and integrated approach to teaching autonomous driving. This approach breaks the complex autonomous driving system into different technology modules and first has the students understand each module. After the students grasp the basic concepts in each technology module, they are asked to perform integration experiments to help them understand the interactions between these modules.

We have successfully applied this methodology to three pilot case studies: an undergraduate-level introduction to autonomous driving class, a graduate-level embedded systems class with a session on autonomous driving, as well as a professional training session at an autonomous driving company. Although the students in these three pilot case studies have very diverse backgrounds, the modular teaching approach allowed us to adjust the order of modules flexibly to fit the needs of different students, and the integration experiments enabled the students to understand the interactions between different modules. This gave the students a comprehensive understanding of the modules as well as their interactions. In addition, our experiences showed that the proposed approach allowed students to start with their comfortable modules and then move on to other modules, therefore enabling the students to maintain a high interest level and good performance.

REFERENCES

[1] Liu, S., Peng, J., and Gaudiot, J-L. 2017. Computer, drive my car! *Computer*, 50(1), pp. 8–8.

[2] Liu, S., Tang, J., Zhang, Z., and Gaudiot, J-L. 2017. Computer architectures for autonomous driving. *Computer*, 50(8), pp. 18–25.

[3] Liu, S., Tang, J., Wang, C., Wang, Q., and Gaudiot, J-L. 2017. A unified cloud platform for autonomous driving. *Computer*, (12), pp. 42–49.

[4] Liu, S., Li, L., Tang, J., Wu, S., and Gaudiot, J-L. 2017. Creating autonomous vehicle systems. *Synthesis Lectures on Computer Science*, 6(1), pp. i–186. xxi

[5] IEEE Computer Society, Creating Autonomous Vehicle Systems, accessed 1 Feb 2018, https://www.youtube.com/watch?v=B8A6BiRkNUw&t=93s. xxi, xxii

[6] OReilly, Enabling Computer-Vision-Based Autonomous Vehicles, accessed 1 Feb 2018, https://www.youtube.com/watch?v=89giovpaTUE&t=434s. xxi

[7] PerceptIn Autonomous Vehicle Development, accessed 1 Feb 2018, https://www.youtube.com/watch?v=rzRC57IXtRY. xxiv

[8] PerceptIn, PerceptIn Robot System Running on a Cell Phone, accessed 1 Feb 2018, https://www.youtube.com/watch?v=Mib8SXacKEE. xxi

[9] ORB-SLAM, accessed 1 Feb 2018, http://webdiis.unizar.es/~raulmur/orbslam/. xxiii

[10] SqueezeNet, accessed 1 Feb 2018, https://github.com/DeepScale/SqueezeNet. xxiii

[11] Tang, J., Liu, S., Wang, C., and Liu, C. 2017. Distributed simulation platform for autonomous driving. *International Conference on Internet of Vehicles (IOV)* 2017: pp. 190–200. xxi

[12] Apache MXNET, accessed 1 Feb 2018, https://mxnet.apache.org/. xxii

[13] Costa, V., Rossetti, R., and Sousa, A. 2017. Simulator for teaching robotics, ROS and autonomous driving in a competitive mindset. *International Journal of Technology and Human Interaction*, 13(4), p. 14. xviii

[14] Arnaldi, N., Barone, C., Fusco, F., Leofante, F., and Tacchella, A. 2016. Autonomous driving and undergraduates: An affordable setup for teaching robotics, *Proceedings of the 3rd Italian Workshop on Artificial Intelligence and Robotics*, pp.,5–9, Genova, Italy, November 28, 2016. xviii

[15] Artificial General Intelligence, accessed 1 Feb 2018, https://agi.mit.edu/. xviii

[16] Deep Learning for Self-Driving Cars, accessed 1 Feb 2018, https://selfdrivingcars.mit.edu/. xviii

[17] Artificial Intelligence: Principles and Techniques, accessed 1 Feb 2018, http://web.stanford.edu/class/cs221/. xviii

[18] Paull, L., Tani, J., Zuber, M.T., Rus, D., How, J., Leonard, J., and Censi, A. 2016. Duckietown: An open, inexpensive and flexible platform for autonomy education and research, *IEEE International Conference on Robotics and Automation*, Singapore, May. 2017, pp. 1–8. DOI: 10.1007/978-3-319-55553-9_8. xviii

[19] Karaman, S., Anders, A., Boulet, M., Connor, M.T., Abbott, J., Gregson, K.L., Guerra, W. J., Guldner, O.R., Mohamoud, M.M., Plancher, B.K., Robert, T-I., and Vivilecchia, J.R. 2017. Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at MIT. *IEEE Integrated STEM Education Conference*, pp. 195–203, 2017. xviii

[20] Tan, S. and Shen, Z., Hybrid problem-based learning in digital image processing: A case study, *IEEE Transactions on Education*, 2017, (99): pp. 1–9. xviii

[21] Robotica 2017, accessed 1 Feb 2018, http://robotica2017.isr.uc.pt/index.php/en/competitions/major/autonomous-driving. xviii

[22] Autonomous Driving Challenge, accessed 1 Feb 2018, http://www.autodrivechallenge.org/. xviii

[23] NXP CUP Intelligent Car Racing, accessed 1 Feb 2018, https://community.nxp.com/groups/tfc-emea. xviii