EXHIBIT 1043

Scheduled Maintenance: On Thursday, April 4, IEEE *Xplore* will undergo scheduled maintenance from 1:00-3:00 PM ET (5:00-7:00 PM UTC). During this time, there may be intermittent impact on performance. We apologize for any inconvenience.

A Survey of Autonomous Driving: *Common Practices and Emerging Technologies*

Publisher: IEEE

Cite This PDF

Ekim Yurtsever ; Jacob Lambert ; Alexander Carballo ; Kazuya Takeda **All Authors**

756 Cites in Papers

53196 Full Text Views

Alerts

Manage Content Alerts

Add to Citation Alerts

Open Access Comment(s)

Under a Creative Commons License

Abstract

Document Sections

I. Introduction

II. Prospects and Challenges

III. System Components and Architecture

IV. Localization and Mapping

V. Perception

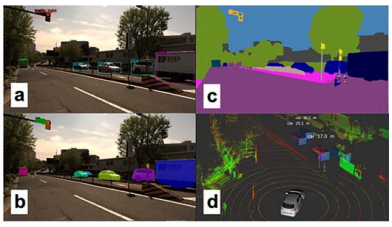
Show Full Outline

Authors

Figures

References

Downl PDF



This figure illustrates outputs of state-of-the-art perception algorithms in an urban scene near Nagoya University, with camera and lidar data collected by our experiment... View more

Abstract: Automated driving systems (ADSs) promise a safe, comfortable and efficient driving experience. However, fatalities involving vehicles equipped with ADSs are on the rise. ... **View more**

Citations

Keywords

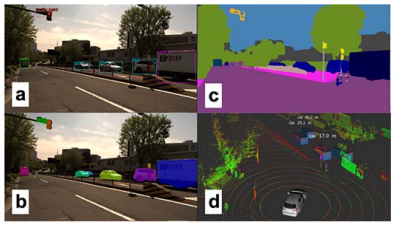
Metrics

More Like This

Footnotes

Metadata

Abstract:
Automated driving systems (ADSs) promise a safe, comfortable and efficient driving experience. However, fatalities involving vehicles equipped with ADSs are on the rise. The full potential of ADSs cannot be realized unless the robustness of state-of-the-art is improved further. This paper discusses unsolved problems and surveys the technical aspect of automated driving. Studies regarding present challenges, high-level system architectures, emerging methodologies and core functions including localization, mapping, perception, planning, and human machine interfaces, were thoroughly reviewed. Furthermore, many state-of-the-art algorithms were implemented and compared on our own platform in a real-world driving setting. The paper concludes with an overview of available datasets and tools for ADS development.



This figure illustrates outputs of state-of-the-art perception algorithms in an urban scene near Nagoya University, with camera and lidar data collected by our experiment... View more

Published in: IEEE Access (Volume: 8)

Page(s): 58443 - 58469

DOI: 10.1109/ACCESS.2020.2983149

Date of Publication: 25 March 2020

Publisher: IEEE

Electronic ISSN: 2169-3536

Contents

CCBY - IEEE is not the copyright holder of this material. Please follow the instructions via https :// creativecommons . org / licenses / by / 4 . 0 / to obtain full-text articles and stipulations in the API documentation.

SECTION I.
Introduction

According to a recent technical report by the National Highway Traffic Safety Administration (NHTSA), 94% of road accidents are caused by human errors [1]. Against this backdrop, Automated Driving Systems (ADSs) are being developed with the promise of preventing accidents, reducing emissions, transporting the mobility-impaired and reducing driving related stress [2]. If widespread deployment can be realized, annual social benefits of ADSs are projected to reach nearly $800 billion by 2050 through congestion mitigation, road casualty reduction, decreased energy consumption and increased productivity caused by the reallocation of driving time [3].

The accumulated knowledge in vehicle dynamics, breakthroughs in computer vision caused by the advent of deep learning [4] and availability of new sensor modalities, such as lidar [5], catalyzed ADS research and industrial implementation. Furthermore, an increase in public interest and market potential precipitated the emergence of ADSs with varying degrees of automation. However, robust automated driving in urban environments has not been achieved yet [6]. Accidents caused

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 29, 2020, accepted March 22, 2020, date of publication March 25, 2020, date of current version April 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983149

A Survey of Autonomous Driving: *Common Practices and Emerging Technologies*

EKIM YURTSEVER[1], (Member, IEEE), JACOB LAMBERT[1], (Student Member, IEEE), ALEXANDER CARBALLO[1], (Member, IEEE), AND KAZUYA TAKEDA[1,2], (Senior Member, IEEE)

[1]Graduate School of Informatics, Nagoya University, Nagoya 464-8603, Japan
[2]Tier IV Inc., Nagoya 113-0033, Japan

Corresponding author: Ekim Yurtsever (ekimyurtsever@gmail.com)

ABSTRACT Automated driving systems (ADSs) promise a safe, comfortable and efficient driving experience. However, fatalities involving vehicles equipped with ADSs are on the rise. The full potential of ADSs cannot be realized unless the robustness of state-of-the-art is improved further. This paper discusses unsolved problems and surveys the technical aspect of automated driving. Studies regarding present challenges, high-level system architectures, emerging methodologies and core functions including localization, mapping, perception, planning, and human machine interfaces, were thoroughly reviewed. Furthermore, many state-of-the-art algorithms were implemented and compared on our own platform in a real-world driving setting. The paper concludes with an overview of available datasets and tools for ADS development.

INDEX TERMS Autonomous vehicles, control, robotics, automation, intelligent vehicles, intelligent transportation systems.

I. INTRODUCTION

According to a recent technical report by the National Highway Traffic Safety Administration (NHTSA), 94% of road accidents are caused by human errors [1]. Against this backdrop, Automated Driving Systems (ADSs) are being developed with the promise of preventing accidents, reducing emissions, transporting the mobility-impaired and reducing driving related stress [2]. If widespread deployment can be realized, annual social benefits of ADSs are projected to reach nearly $800 billion by 2050 through congestion mitigation, road casualty reduction, decreased energy consumption and increased productivity caused by the reallocation of driving time [3].

The accumulated knowledge in vehicle dynamics, breakthroughs in computer vision caused by the advent of deep learning [4] and availability of new sensor modalities, such as lidar [5], catalyzed ADS research and industrial implementation. Furthermore, an increase in public interest and market potential precipitated the emergence of ADSs with varying degrees of automation. However, robust automated driving in urban environments has not been achieved yet [6]. Accidents caused by immature systems [7]–[10] undermine trust, and furthermore, cost lives. As such, a thorough investigation of unsolved challenges and the state-of-the-art is deemed necessary here.

Eureka Project PROMETHEUS [11] was carried out in Europe between 1987-1995, and it was one of the earliest major automated driving studies. The project led to the development of VITA II by Daimler-Benz, which succeeded in automatically driving on highways [12]. DARPA Grand Challenge, organized by the US Department of Defense in 2004, was the first major automated driving competition where all of the attendees failed to finish the 150-mile off-road parkour. The difficulty of the challenge was in the rule that no human intervention at any level was allowed during the finals. Another similar DARPA Grand Challenge was held in 2005. This time five teams managed to complete the off-road track without any human interference [13].

Fully automated driving in urban scenes was seen as the biggest challenge of the field since the earliest attempts. During DARPA Urban Challenge [26], held in 2007, many different research groups around the globe tried their ADSs in a test environment that was modeled after a typical urban scene. Six teams managed to complete the event. Even though this competition was the biggest and most significant event up to that time, the test environment lacked certain aspects of a real-world urban driving scene such as pedestrians and cyclists. Nevertheless, the fact that six teams managed to complete the challenge attracted significant attention. After

The associate editor coordinating the review of this manuscript and approving it for publication was Liang Hu.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 4

TABLE 1. Comparison of ADS related survey papers.

Related work	Survey coverage									
	Connected systems	End-to-end	Localization	Perception	Assessment	Planning	Control	HMI	Datasets & software	Implementation
[14]	-	-	-	✓	-	-	-	-	-	-
[15]	-	-	✓	✓	-	✓	✓	-	-	✓
[16]	-	-	✓	✓	-	✓	-	-	-	-
[17]	-	-	✓	-	-	-	-	-	-	-
[18]	-	-	-	-	-	✓	✓	-	-	-
[19]	-	-	-	-	-	✓	-	-	-	-
[20]	✓	✓	✓	✓	-	-	-	-	-	-
[21]	✓	-	✓	-	-	-	-	-	✓	-
[22]	✓	-	-	-	-	-	-	-	-	-
[23]	-	-	✓	✓	-	✓	✓	-	-	✓
[24]	-	✓	-	-	✓	✓	✓	-	-	-
[25]	-	-	-	-	✓	✓	-	-	-	-
Ours	✓	✓	✓	✓	✓	✓	-	✓	✓	✓

DARPA Urban Challenge, several more automated driving competitions such as [27]–[30] were held in different countries.

Common practices in system architecture have been established over the years. Most of the ADSs divide the massive task of automated driving into subcategories and employ an array of sensors and algorithms on various modules. More recently, end-to-end driving started to emerge as an alternative to modular approaches. Deep learning models have become dominant in many of these tasks [31].

The Society of Automotive Engineers (SAE) refers to hardware-software systems that can execute dynamic driving tasks (DDT) on a sustainable basis as ADS [32]. There are also vernacular alternative terms such as "autonomous driving" and "self-driving car" in use. Nonetheless, despite being commonly used, SAE advices not to use them as these terms are unclear and misleading. In this paper we follow SAE's convention.

The present paper attempts to provide a structured and comprehensive overview of state-of-the-art automated driving related hardware-software practices. Moreover, emerging trends such as end-to-end driving and connected systems are discussed in detail. There are overview papers on the subject, which covered several core functions [15], [16], and which concentrated only on the motion planning aspect [18], [19]. However, a survey that covers: present challenges, available and emerging high-level system architectures, individual core functions such as localization, mapping, perception, planning, vehicle control, and human-machine interface altogether does not exist. The aim of this paper is to fill this gap in the literature with a thorough survey. In addition, a detailed summary of available datasets, software stacks, and simulation tools is presented here. Another contribution of this paper is the detailed comparison and analysis of alternative approaches through implementation. We implemented some state-of-the-art algorithms in our platform using open-source software. Comparison of existing overview papers and our work is shown in Table 1.

The remainder of this paper is written in eight sections. Section II is an overview of present challenges. Details of automated driving system components and architectures are given in Section III. Section IV presents a summary of state-of-the-art localization techniques followed by Section V, an in-depth review of perception models. Assessment of the driving situation and planning are discussed in Section VI and VII respectively. In Section VIII, current trends and shortcomings of human machine interface are introduced. Datasets and available tools for developing automated driving systems are given in Section IX.

II. PROSPECTS AND CHALLENGES

A. SOCIAL IMPACT

Widespread usage of ADSs is not imminent. Yet it is still possible to foresee its potential impact and benefits to a certain degree:

1) *Problems that can be solved*: preventing traffic accidents, mitigating traffic congestions, reducing emissions
2) *Arising opportunities*: reallocation of driving time, transporting the mobility impaired
3) *New trends*: consuming Mobility as a Service (MaaS), logistics revolution

Widespread deployment of ADSs can reduce the societal loss caused by erroneous human behavior such as distraction, driving under influence and speeding [3].

Globally, the elder group (over 60 years old) is growing faster than the younger groups [33]. Increasing the mobility of elderly with ADSs can have a huge impact on the quality of life and productivity of a large portion of the population.

A shift from personal vehicle-ownership towards consuming Mobility as a Service (MaaS) is an emerging trend. Currently, ride-sharing has lower costs compared to vehicle-ownership under 1000 km annual mileage [34]. The ratio of owned to shared vehicles is expected to be 50:50 by 2030 [35]. Large scale deployment of ADSs can accelerate this trend.

B. CHALLENGES

ADSs are complicated robotic systems that operate in indeterministic environments. As such, there are myriad scenarios

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 5

with unsolved issues. This section discusses the high level challenges of driving automation in general. More minute, task-specific details are discussed in corresponding sections.

The Society of Automotive Engineers (SAE) defined five levels of driving automation in [32]. In this taxonomy, *level zero* stands for no automation at all. Primitive driver assistance systems such as adaptive cruise control, anti-lock braking systems and stability control start with *level one* [36]. *Level two* is partial automation to which advanced assistance systems such as emergency braking or collision avoidance [37], [38] are integrated. With the accumulated knowledge in the vehicle control field and the experience of the industry, level two automation became a feasible technology. The real challenge starts above this level.

Level three is conditional automation; the driver could focus on tasks other than driving during normal operation, however, s/he has to quickly respond to an emergency alert from the vehicle and be ready to take over. In addition, level three ADS operate only in limited operational design domains (ODDs) such as highways. Audi claims to be the first production car to achieve level 3 automation in limited highway conditions [39]. However, taking over the control manually from the automated mode by the driver raises another issue. Recent studies [40], [41] investigated this problem and found that the takeover situation increases the collision risk with surrounding vehicles. The increased likelihood of an accident during a takeover is a problem that is yet to be solved.

Human attention is not needed in any degree at level four and five. However, *level four* can only operate in limited ODDs where special infrastructure or detailed maps exist. In the case of departure from these areas, the vehicle must stop the trip by automatically parking itself. The fully automated system, *level five*, can operate in any road network and any weather condition. No production vehicle is capable of level four or level five driving automation yet. Moreover, Toyota Research Institute stated that no one in the industry is even close to attaining level five automation [42].

Level four and above driving automation in urban road networks is an open and challenging problem. The environmental variables, from weather conditions to surrounding human behavior, are highly indeterministic and difficult to predict. Furthermore, system failures lead to accidents: in the Hyundai competition one of the ADSs crashed because of rain [7], Google's ADS hit a bus while lane changing because it failed to estimate the speed of a bus [8], and Tesla's Autopilot failed to recognize a white truck and collided with it, killing the driver [9].

Fatalities [9], [10] caused by immature technology undermine public acceptance of ADSs. According to a recent survey [34], the majority of consumers question the safety of the technology, and want a significant amount of control over the development and use of ADS. On the other hand, extremely cautious ADSs are also making a negative impression [43].

Ethical dilemmas pose another set of challenges. In an inevitable accident situation, how should the system behave [44]? Experimental ethics were proposed regarding this issue [45].

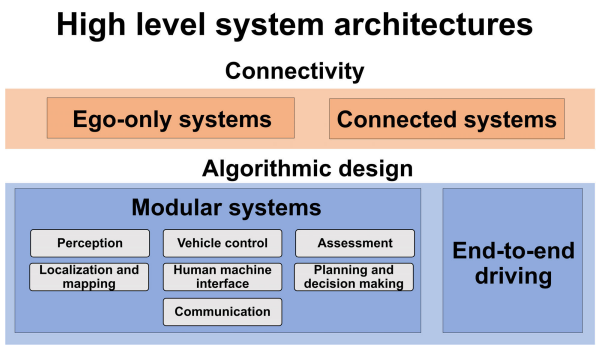


FIGURE 1. A high level classification of automated driving system architectures.

Risk and reliability certification is another task yet to be solved. Like in aircraft, ADSs need to be designed with high redundancies that will minimize the chance of a catastrophic failure. Even though there is promising projects in this regard such as DeepTest [46], the design-simulation-test-redesign-certification procedure is still not established by the industry nor the rule-makers.

Finally, various optimization goals such as time to reach the destination, fuel efficiency, comfort, and ride-sharing optimization increases the complexity of an already difficult to solve problem. As such, carrying all of the dynamic driving tasks safely under strict conditions outside a well defined, geofenced area remains as an open problem.

III. SYSTEM COMPONENTS AND ARCHITECTURE

A. SYSTEM ARCHITECTURE

Classification of system architectures is shown in Figure 1. ADSs are designed either as standalone, ego-only systems [15], [47] or connected multi-agent systems [48]–[50]. Furthermore, these design philosophies are realized with two alternative approaches: modular [15], [47], [51]–[58] or end-to-end driving [59]–[67].

1) EGO-ONLY SYSTEMS

The ego-only approach is to carry all of the necessary automated driving operations on a single self-sufficient vehicle at all times, whereas a connected ADS may or may not depend on other vehicles and infrastructure elements given the situation. Ego-only is the most common approach amongst the state-of-the-art ADSs [15], [47], [51]–[56], [56]–[58]. We believe this is due to the practicality of having a self-sufficient platform for development and the additional challenges of connected systems.

2) MODULAR SYSTEMS

Modular systems, referred as the mediated approach in some works [59], are structured as a pipeline of separate components linking sensory inputs to actuator outputs [31].

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 6

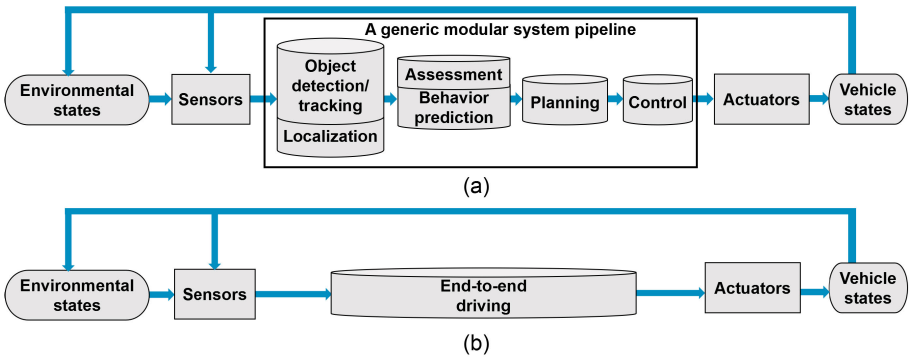


FIGURE 2. Information flow diagrams of: (a) a generic modular system, and (b) an end-to-end driving system.

Core functions of a modular ADS can be summarized as: localization and mapping, perception, assessment, planning and decision making, vehicle control, and human-machine interface. Typical pipelines [15], [47], [51]–[56], [56]–[58] start with feeding raw sensor inputs to localization and object detection modules, followed by scene prediction and decision making. Finally, motor commands are generated at the end of the stream by the control module [31], [68].

Developing individual modules separately divides the challenging task of automated driving into an easier-to-solve set of problems [69]. These sub-tasks have their corresponding literature in robotics [70], computer vision [71] and vehicle dynamics [36], which makes the accumulated know-how and expertise directly transferable. This is a major advantage of modular systems. In addition, functions and algorithms can be integrated or built upon each other in a modular design. E.g, a safety constraint [72] can be implemented on top of a sophisticated planning module to force some hard-coded emergency rules without modifying the inner workings of the planner. This enables designing redundant but reliable architectures.

The major disadvantages of modular systems are being prone to error propagation [31] and over-complexity. In the unfortunate Tesla accident, an error in the perception module in the form of a misclassification of a white trailer as sky, propagated down the pipeline until failure, causing the first ADS related fatality [46].

3) END-TO-END DRIVING

End-to-end driving, referred as direct perception in some studies [59], generate ego-motion directly from sensory inputs. Ego-motion can be either the continuous operation of steering wheel and pedals or a discrete set of actions, e.g, acceleration and turning left. There are three main approaches for end-to-end driving: direct supervised deep learning [59]–[63], neuroevolution [66], [67] and the more recent deep reinforcement learning [64], [65]. The flow diagram of a generic end-to-end driving system is shown in Figure 2 and comparison of the approaches is given in Table 2.

TABLE 2. Common end-to-end driving approaches.

Related works	Learning/training strategy	Pros/cons
[59]–[63]	Direct supervised deep learning	Imitates the target data: usually a human driver. Can be trained offline. Poor generalization performance.
[64], [65]	Deep reinforcement learning	Learns the optimum way of driving. Requires online interaction. Urban driving has not been achieved yet
[66], [67]	Neuroevolution	No backpropagation. Requires online interaction. Real world driving has not been achieved yet.

The earliest end-to-end driving attempt dates back to ALVINN [60], where a 3-layer fully connected network was trained to output the direction that the vehicle should follow. An end-to-end driving system for off-road driving was introduced in [61]. With the advances in artificial neural network research, deep convolutional and temporal networks became feasible for automated driving tasks. A deep convolutional neural network that takes image as input and outputs steering was proposed in [62]. A spatiotemporal network, an FCN-LSTM architecture, was developed for predicting ego-vehicle motion in [63]. DeepDriving is another convolutional model that tries to learn a set of discrete perception indicators from the image input [59]. This approach is not entirely end-to-end though, the proper driving actions in the perception indicators have to be generated by another module. All of the mentioned methods follow direct supervised training strategies. As such, ground truth is required for training. Usually, the ground truth is the ego-action sequence of an expert human driver and the network learns to imitate the driver. This raises an import design question: should the ADS drive like a human?

A novel deep reinforcement learning model, Deep Q Networks (DQN), combined reinforcement learning with deep learning [73]. In summary, the goal of the network is to select a set of actions that maximize cumulative future

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 7

rewards. A deep convolutional neural network was used to approximate the optimal action reward function. Actions are generated first with random initialization. Then, the network adjust its parameters with experience instead of direct supervised learning. An automated driving framework using DQN was introduced in [64], where the network was tested in a simulation environment. The first real world run with DQN was achieved in a countryside road without traffic [65]. DQN based systems do not imitate the human driver, instead, they learn the optimum way of driving.

Neuroevolution refers to using evolutionary algorithms to train artificial neural networks [74]. End-to-end driving with neuroevolution is not popular as DQN and direct supervised learning. To the best of our knowledge, real world end-to-end driving with neuroevolution is not achieved yet. However, some promising simulation results were obtained [66], [67]. ALVINN was trained with neuroevolution and outperformed the direct supervised learning version [66]. A RNN was trained with neuroevolution in [67] using a driving simulator. The biggest advantage of neuroevolution is the removal of backpropagation, hence, the need for direct supervision.

End-to-end driving is promising, however it has not been implemented in real-world urban scenes yet, except limited demonstrations. The biggest shortcomings of end-to-end driving in general are the lack of hard coded safety measures and interpretability [69]. In addition, DQN and neuroevolution has one major disadvantage over direct supervised learning: these networks must interact with the environment online and fail repeatedly to learn the desired behavior. On the contrary, direct supervised networks can be trained offline with human driving data, and once the training is done, the system is not expected to fail during operation.

4) CONNECTED SYSTEMS

There is no operational connected ADS in use yet, however, some researchers believe this emerging technology will be the future of driving automation [48]–[50]. With the use of Vehicular Ad hoc NETwork (VANETs), the basic operations of automated driving can be distributed amongst agents. V2X is a term that stands for ''vehicle to everything.'' From mobile devices of pedestrians to stationary sensors on a traffic light, an immense amount of data can be accessed by the vehicle with V2X [22]. By sharing detailed information of the traffic network amongst peers [75], shortcomings of the ego-only platforms such as sensing range, blind spots, and computational limits may be eliminated. More V2X applications that will increase safety and traffic efficiency are expected to emerge in the foreseeable future [76].

VANETs can be realized in two different ways: conventional IP based networking and Information-Centric Networking (ICN) [48]. For vehicular applications, lots of data have to be distributed amongst agents with intermittent and in less than ideal connections while maintaining high mobility [50]. Conventional IP-host based Internet protocol cannot function properly under these conditions. On the other hand, in information-centric networking, vehicles stream query messages to an area instead of a direct address and they accept corresponding responses from any sender [49]. Since vehicles are highly mobile and dispersed on the road network, the identity of the information source becomes less relevant. In addition, local data often carries more crucial information for immediate driving tasks such as avoiding a rapidly approaching vehicle on a blind spot.

Early works, such as the CarSpeak system [82], proved that vehicles can utilize each other's sensors and use the shared information to execute some dynamic driving tasks. However, without reducing huge amounts of continuous driving data, sharing information between hundreds of thousand vehicles in a city could not become feasible. A semiotic framework that integrates different sources of information and converts raw sensor data into meaningful descriptions was introduced in [83] for this purpose. In [84], the term Vehicular Cloud Computing (VCC) was coined and the main advantages of it over conventional Internet cloud applications was introduced. Sensors are the primary cause of the difference. In VCC, sensor information is kept on the vehicle and only shared if there is a local query from another vehicle. This potentially saves the cost of uploading/downloading a constant stream of sensor data to the web. Besides, the high relevance of local data increases the feasibility of VCC. Regular cloud computing was compared to vehicular cloud computing and it was reported that VCC is technologically feasible [85]. The term ''Internet of Vehicles'' (IoV) was proposed for describing a connected ADS [48] and the term ''vehicular fog'' was introduced in [49].

Establishing an efficient VANET with thousands of vehicles in a city is a huge challenge. For an ICN based VANET, some of the challenging topics are security, mobility, routing, naming, caching, reliability and multi-access computing [86]. In summary, even though the potential benefits of a connected system is huge, the additional challenges increase the complexity of the problem to a significant degree. As such, there is no operational connected system yet.

B. SENSORS AND HARDWARE

State-of-the-art ADSs employ a wide selection of onboard sensors. High sensor redundancy is needed in most of the tasks for robustness and reliability. Hardware units can be categorized into five: exteroceptive sensors for perception, proprioceptive sensors for internal vehicle state monitoring tasks, communication arrays, actuators, and computational units.

Exteroceptive sensors are mainly used for perceiving the environment, which includes dynamic and static objects, e.g., drivable areas, buildings, pedestrian crossings. Camera, lidar, radar and ultrasonic sensors are the most commonly used modalities for this task. A detailed comparison of exteroceptive sensors is given in Table 3.

1) MONOCULAR CAMERAS

Cameras can sense color and are passive, i.e. they do not emit any signal for measurements. Sensing color is extremely

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 8

TABLE 3. Exteroceptive sensors.

Modality	Affected by Illumination	Affected by weather	Color	Depth	Range	Accuracy	Size	Cost
Lidar	-	✓	-	✓	medium (< 200m)	high	large*	high*
Radar	-	-	-	✓	high	medium	small	medium
Ultrasonic	-	-	-	✓	short	low	small	low
Camera	✓	✓	✓	-	-	-	smallest	lowest
Stereo Camera	✓	✓	✓	✓	medium (< 100m)	low	medium	low
Flash Camera [77]	✓	✓	✓	✓	medium (< 100m)	low	medium	low
Event Camera [78]	limited	✓	-	-	-	-	smallest	low
Thermal Camera [79], [80]	-	✓	-	-	-	-	smallest	low

* Cost, size and weight of lidars started to decrease recently [81]

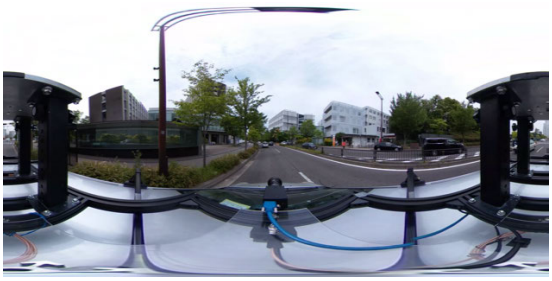


FIGURE 3. Ricoh Tetha V panoramic images collected using our data collection platform, in Nagoya University campus. Note some distortion still remains on the periphery of the image.

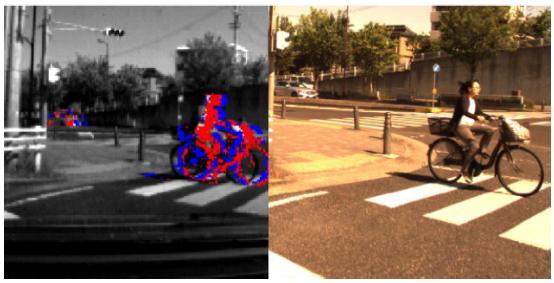


FIGURE 4. DAVIS240 events, overlayed on the image (left) and corresponding RBG image from a different camera (right), collected by our data collection platform, at a road crossing near Nagoya University. The motion of the cyclist and vehicle causes brightness changes which trigger events.

important for tasks such as traffic light recognition. Furthermore, 2D computer vision is an established field with remarkable state-of-the-art algorithms. Moreover, a passive sensor does not interfere with other systems since it does not emit any signals. However, cameras have certain shortcomings. Illumination conditions affect their performance drastically, and depth information is difficult to obtain from a single camera. There are promising studies [87] to improve monocular camera based depth perception, but modalities that are not negatively affected by illumination and weather conditions are still necessary for dynamic driving tasks. Other camera types gaining interest for ADS include flash cameras [77], thermal cameras [79], [80], and event cameras [78].

2) OMNIDIRECTIONAL CAMERA

For 360° 2D vision, omnidirectional cameras are used as an alternative to camera arrays. They have seen widespread use, with increasingly compact and high performance hardware being constantly released. Panoramic view is particularly desirable for applications such as navigation, localization and mapping [88]. An example panoramic image is shown in Figure 3.

3) EVENT CAMERAS

Event cameras are among the newer sensing modalities that have seen use in ADS [89]. Event cameras record data asynchronously for individual pixels with respect to visual stimulus. The output is therefore an irregular sequence of data points, or *events* triggered by changes in brightness. The response time is in the order of microseconds [90]. The main limitation of current event cameras is pixel size and image resolution. For example, the DAVIS40 image shown in Figure 4 has a pixel size of 18.5×18.5 μm and a resolution of 240×180. Recently, a driving dataset with event camera data has been published [89].

4) RADAR

Radar, lidar and ultrasonic sensors are very useful in covering the shortcomings of cameras. Depth information, i.e. distance to objects, can be measured effectively to retrieve 3D information with these sensors, and they are not affected by illumination conditions. However, they are active sensors. Radars emit radio waves that bounce back from objects and measure the time of each bounce. Emissions from active sensors can interfere with other systems. Radar is a well-established technology that is both lightweight and cost-effective. For example, radars can fit inside side-mirrors. Radars are cheaper and can detect objects at longer distances than lidars, but the latter are more accurate.

5) LIDAR

Lidar operates with a similar principle that of radar but it emits infrared light waves instead of radio waves. It has

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 9

much higher accuracy than radar under 200 meters. Weather conditions such as fog or snow have a negative impact on the performance of lidar. Another aspect is the sensor size: smaller sensors are preferred on the vehicle because of limited space and aerodynamic restraints and lidars are generally larger than radars.

In [91], human sensing performance is compared to ADS. One of the key findings of this study is that even though human drivers are still better at reasoning in general, the perception capability of ADSs with sensor-fusion can exceed humans, especially in degraded conditions such as insufficient illumination.

6) PROPRIOCEPTIVE SENSORS

Proprioceptive sensing is another crucial category. Vehicle states such as speed, acceleration and yaw must be continuously measured in order to operate the platform safely with feedback. Almost all of the modern production cars are equipped with proprioceptive sensors. Wheel encoders are mainly used for odometry, Inertial Measurement Units (IMU) are employed for monitoring the velocity and position changes, tachometers are utilized for measuring speed and altimeters for altitude. These signals can be accessed through the CAN protocol of modern cars.

Besides sensors, an ADS needs actuators to manipulate the vehicle and advanced computational units for processing and storing sensor data.

7) FULL SIZE CARS

There are numerous instrumented vehicles introduced by different research groups, such as Stanford's Junior [15], which employs an array of sensors with different modalities for perceiving external and internal variables. Boss won the DARPA Urban Challenge with an abundance of sensors [47]. RobotCar [53] is a cheaper research platform aimed for data collection. In addition, different levels of driving automation have been introduced by the industry; Tesla's Autopilot [92] and Google's self driving car [93] are some examples. Bertha [57] is developed by Daimler and has 4 120° short-range radars, two long-range range radar on the sides, stereo camera, wide angle-monocular color camera on the dashboard, another wide-angle camera for the back. Our vehicle is shown in Figure 5. A detailed comparison of sensor setups of 10 different full-size ADSs is given in Table 4.

8) LARGE VEHICLES AND TRAILERS

Earliest intelligent trucks were developed for the PATH program in California [102], which utilized magnetic markers on the road. Fuel economy is an essential topic in freight transportation and methods such as platooning has been developed for this purpose. Platooning is a well-studied phenomenon; it reduces drag and therefore fuel consumption [103]. In semi-autonomous truck platooning, the lead truck is driven by a human driver, and several automated trucks follow it; forming a semi-autonomous road-train as defined in [104]. Sartre European Union project [105] introduced such a system that

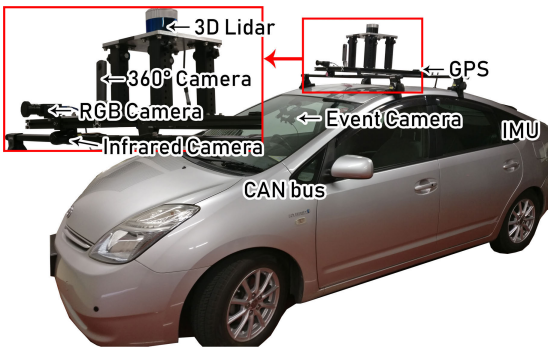


FIGURE 5. The ADS equipped Prius of Nagoya University. We have used this vehicle to perform core automated driving operations.

TABLE 4. Onboard sensor setup of ADS equipped vehicles.

Platform	# 360° rotating lidars	# stationary lidars	# Radars	# Cameras
Ours	1	-	-	4
Boss [47]	1	9	5	2
Junior [15]	1	2	6	4
BRAiVE [52]	-	5	1	10
RobotCar [53]	-	3	-	4
Google car (prius) [55]	1	-	4	1
Uber car (XC90) [56]	1	-	10	7
Uber car (Fusion) [56]	1	7	7	20
Bertha [57]	-	-	6	3
Apollo Auto [58]	1	3	2	2

satisfies three core conditions: using the already existing public road network, sharing the traffic with non-automated vehicles and not modifying the road infrastructure. A platoon consisting of three automated trucks was formed in [103] and significant fuel savings were reported.

Tractor-trailer setup poses an additional challenge for automated freight transport. Conventional control methods such as feedback linearization [106] and fuzzy control [107] were used for path tracking without considering the jackknifing constraint. The possibility of jackknifing, the collision of the truck and the trailer with each other, increases the difficulty of the task [108]. A control safety governor design was proposed in [108] to prevent jackknifing while reversing.

IV. LOCALIZATION AND MAPPING

Localization is the task of finding ego-position relative to a reference frame in an environment [17], and it is fundamental to any mobile robot. It is especially crucial for ADSs [21]; the vehicle must use the correct lane and position itself in it accurately. Furthermore, localization is an elemental requirement for global navigation.

The reminder of this section details the three most common approaches that use solely on-board sensors: Global Positioning System and Inertial Measurement Unit (GPS-IMU) fusion, Simultaneous Localization And Mapping (SLAM), and state-of-the-art a priori map-based localization. Readers are referred to [17] for a broader localization overview. A comparison of localization methods is given in Table 5.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 10

TABLE 5. Localization techniques.

Methods	Robustness	Cost	Accuracy	Size	Computational requirements	Related works
Absolute positioning sensors	low	low	low	small	lowest	[94]
Odometry/dead reckoning	low	low	low	smallest	low	[95]
GPS-IMU fusion	medium	medium	low	small	low	[96]
SLAM	medium-high	medium	high	large	very high	[97]
A priori Map-based						
Landmark search	high	medium	high	large	medium	[98], [99]
Point cloud matching	highest	highest	highest	largest	high	[100], [101]

A. GPS-IMU FUSION

The main principle of GPS-IMU fusion is correcting accumulated errors of dead reckoning in intervals with absolute position readings [109]. In a GPS-IMU system, changes in position and orientation are measured by IMU, and this information is processed for localizing the vehicle with dead reckoning. There is a significant drawback of IMU, and in general dead reckoning: errors accumulate with time and they often lead to failure in long-term operations [110]. With the integration of GPS readings, the accumulated errors of the IMU can be corrected in intervals.

GPS-IMU systems by themselves cannot be used for vehicle localization as they do not meet the performance criteria [111]. In the 2004 DARPA Grand Challenge, the red team from Carnegie Mellon University [96] failed the race because of a GPS error. The accuracy required for urban automated driving is too high for the current GPS-IMU systems used in production cars. Moreover, in dense urban environments, the accuracy drops further, and the GPS stops functioning from time to time because of tunnels [109] and high buildings.

Even though GPS-IMU systems by themselves do not meet the performance requirements and can only be utilized for high-level route planning, they are used for initial pose estimation in tandem with lidar and other sensors in state-of-the-art localization systems [111].

B. SIMULTANEOUS LOCALIZATION AND MAPPING

Simultaneous localization and mapping (SLAM) is the act of online map making and localizing the vehicle in it at the same time. A priori information about the environment is not required in SLAM. It is a common practice in robotics, especially in indoor environments. However, due to the high computational requirements and environmental challenges, running SLAM algorithms outdoors, which is the operational domain of ADSs, is less efficient than localization with a pre-built map [112].

Team MIT used a SLAM approach in DARPA urban challenge [113] and finished it in the 4th place. Whereas, the winner, Carnegie Mellon's Boss [47] and the runner-up, Stanford's Junior [15], both utilized a priori information. In spite of not having the same level of accuracy and efficiency, SLAM techniques have one major advantage over a priori methods: they can work anywhere.

SLAM based methods have the potential to replace a priori techniques if their performances can be increased further [20]. We refer the readers to [21] for a detailed SLAM survey in the intelligent vehicle domain.

C. A PRIORI MAP-BASED LOCALIZATION

The core idea of a priori map-based localization techniques is matching: localization is achieved through the comparison of online readings to the information on a detailed pre-built map and finding the location of the best possible match [111]. Often an initial pose estimation, for example with a GPS, is used at the beginning of the matching process. There are various approaches to map building and preferred modalities.

Changes in the environment affect the performance of map-based methods negatively. This effect is prevalent especially in rural areas where past information of the map can deviate from the actual environment because of changes in roadside vegetation and constructions [114]. Moreover, this method requires an additional step of map making.

There are two different map-based approaches; landmark search and matching.

1) LANDMARK SEARCH

Landmark search is computationally less expensive in comparison to point cloud matching. It is a robust localization technique as long as a sufficient amount of landmarks exists. In an urban environment, poles, curbs, signs and road markers can be used as landmarks.

A road marking detection method using lidar and Monte Carlo Localization (MCL) was used in [98]. In this method, road markers and curbs were matched to a 3D map to find the location of the vehicle. A vision based road marking detection method was introduced in [115]. Road markings detected by a single front camera were compared and matched to a low-volume digital marker map with global coordinates. Then, a particle filter was employed to update the position and heading of the vehicle with the detected road markings and GPS-IMU output. A road marking detection based localization technique using; two cameras directed towards the ground, GPS-IMU dead reckoning, odometry, and a precise marker location map was proposed in [116]. Another vision based method with a single camera and geo-referenced traffic signs was presented in [117].

This approach has one major disadvantage; landmark dependency makes the system prone to fail where landmark amount is insufficient.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 11

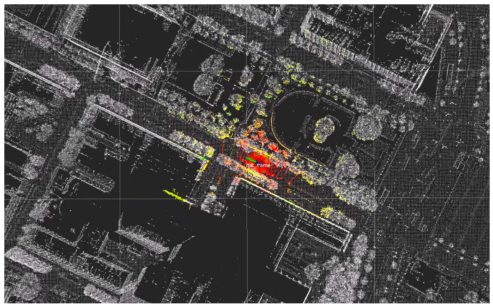


FIGURE 6. We used NDT matching [101], [118] to localize our vehicle in the Nagoya University campus. White points belong to the offline pre-built map and the colored ones were obtained from online scans. The objective is to find the best match between colored points and white points, thus localizing the vehicle.

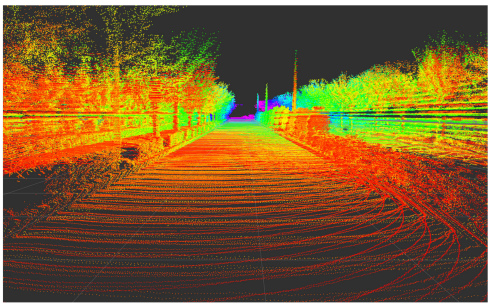


FIGURE 7. Creating a 3D pointcloud map with congregation of scans. We used Autoware [122] for mapping.

2) POINT CLOUD MATCHING

The state-of-the-art localization systems use multi-modal point cloud matching based approaches. In summary, the online-scanned point cloud, which covers a smaller area, is translated and rotated around its center iteratively to be compared against the larger a priori point cloud map. The position and orientation that gives the best match between the two point clouds give the localized position of the sensor relative to the map. For initial pose estimation, GPS is used commonly along dead reckoning. We used this approach to localize our vehicle. The matching process is shown in Figure 6 and the map-making in Figure 7.

In the seminal work of [111], a point cloud map collected with lidar was used to augment inertial navigation and localization. A particle filter maintained a three-dimensional vector of 2D coordinates and the yaw angle. A multi-modal approach with probabilistic maps was utilized in [100] to achieve localization in urban environments with less than 10 cm RMS error. Instead of comparing two point clouds point by point and discarding the mismatched reads, the variance of all observed data was modeled and used for the matching task. A matching algorithm for lidar scans using multi-resolution Gaussian Mixture Maps (GMM) was proposed in [119]. Iterative Closest Point (ICP) was compared against Normal Distribution Transform (NDT) in [118], [120]. In NDT, accumulated sensor readings are transformed into a grid that is represented by the mean and covariance obtained from the scanned points that fall into its' cells/voxels. NDT proved to be more robust than point-to-point ICP matching. An improved version of 3D NDT matching was proposed in [101], and [114] augmented NDT with road marker matching. An NDT-based Monte Carlo Localization (MCL) method that utilizes an offline static map and a constantly updated short-term map was developed by [121]. In this method, NDT occupancy grid was used for the short-term map and it was utilized only when and where the static map failed to give sufficient explanations.

Map-making and maintaining is time and resource consuming. Therefore some researchers such as [99] argue that methods with a priori maps are not feasible given the size of road networks and rapid changes.

3) 2D TO 3D MATCHING

Matching online 2D readings to a 3D a priori map is an emerging technology. This approach requires only a camera on the ADS equipped vehicle instead of the more expensive lidar. The a priori map still needs to be created with a lidar.

A monocular camera was used to localize the vehicle in a point cloud map in [123]. With an initial pose estimation, 2D synthetic images were created from the offline 3D point cloud map and they were compared with normalized mutual information to the online images received from the camera. This method increases the computational load of the localization task. Another vision matching algorithm was introduced in [124] where a stereo camera setup was utilized to compare online readings to synthetic depth images generated from 3D prior.

Camera based localization approaches could become popular in the future as the hardware requirement is cheaper than lidar based systems.

V. PERCEPTION

Perceiving the surrounding environment and extracting information which may be critical for safe navigation is a critical objective for ADS. A variety of tasks, using different sensing modalities, fall under the category of perception. Building on decades of computer vision research, cameras are the most commonly used sensor for perception, with 3D vision becoming a strong alternative/supplement.

The reminder of this section is divided into core perception tasks. We discuss image-based object detection in Section V-A1, semantic segmentation in Section V-A2, 3D object detection in Section V-A3, road and lane detection in Section V-C and object tracking in Section V-B.

A. DETECTION

1) IMAGE-BASED OBJECT DETECTION

Object detection refers to identifying the location and size of objects of interest. Both static objects, from traffic lights and signs to road crossings, and dynamic objects such as other vehicles, pedestrians or cyclists are of concern to ADSs.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 12

Generalized object detection has a long-standing history as a central problem in computer vision, where the goal is to determine if objects of specific classes are present in an image, then to determine their size via a rectangular bounding box. This section mainly discusses state-of-the-art object detection methods, as they represent the starting point of several other tasks in an ADS pipe, such as object tracking and scene understanding.

Object recognition research started more than 50 years ago, but only recently, in the late 1990s and early 2000s, has algorithm performance reached a level of relevance for driving automation. In 2012, the deep convolutional neural network (DCNN) AlexNet [4] shattered the ImageNet image recognition challenge [131]. This resulted in a near complete shift of focus to supervised learning and in particular deep learning for object detection. There exists a number of extensive surveys on general image-based object detection [132]–[134]. Here, the focus is on the state-of-the-art methods that could be applied to ADS.

While state-of-the-art methods all rely on DCNNs, there currently exist a clear distinction between them:

1) Single stage detection frameworks use a single network to produce object detection locations and class prediction simultaneously.
2) Region proposal detection frameworks use two distinct stages, where general regions of interest are first proposed, then categorized by separate classifier networks.

Region proposal methods are currently leading detection benchmarks, but at the cost requiring high computation power, and generally being difficult to implement, train and fine-tune. Meanwhile, single stage detection algorithms tend to have fast inference time and low memory cost, which is well-suited for real-time driving automation. YOLO (You Only Look Once) [135] is a popular single stage detector, which has been improved continuously [128], [136]. Their network uses a DCNN to extract image features on a coarse grid, significantly reducing the resolution of the input image. A fully-connected neural network then predicts class probabilities and bounding box parameters for each grid cell and class. This design makes YOLO very fast, the full model operating at 45 FPS and a smaller model operating at 155 FPS for a small accuracy trade-off. More recent versions of this method, YOLOv2, YOLO9000 [136] and YOLOv3 [128] briefly took over the PASCAL VOC and MS COCO benchmarks while maintaining low computation and memory cost. Another widely used algorithm, even faster than YOLO, is the Single Shot Detector (SSD) [137], which uses standard DCNN architectures such as VGG [130] to achieve competitive results on public benchmarks. SSD performs detection on a coarse grid similar to YOLO, but also uses higher resolution features obtained early in the DCNN to improve detection and localization of small objects.

Considering both accuracy and computational cost is essential for detection in ADS; the detection needs to be reliable, but also operate better than real-time, to allow as much time as possible for the planning and control modules to react to those objects. As such, single stage detectors are often the detection algorithms of choice for ADSs. However, as shown in Table 6, region proposal networks (RPN), used in two-stage detection frameworks, have proven to be unmatched in terms of object recognition and localization accuracy, and computational cost has improved greatly in recent years. They are also better suited for other tasks related to detection, such as semantic segmentation as discussed in Section V-A2. Through transfer learning, RPNs achieving multiple perception tasks simultaneously are become increasingly feasible for online applications [138]. RPNs can replace single stage detection networks for ADS applications in the near future.

TABLE 6. Comparison of 2D bounding box estimation architectures on the test set of ImageNet1K, ordered by Top 5% error. Number of parameters (Num. Params) and number of layers (Num. Layers), hints at the computational cost of the algorithm.

Architecture	Num. Params ($\times 10^6$)	Num. Layers	ImageNet1K Top 5 Error %
Incept.ResNet v2 [125]	30	95	4.9
Inception v4 [125]	41	75	5
ResNet101 [126]	45	100	6.05
DenseNet201 [127]	18	200	6.34
YOLOv3-608 [128]	63	53+1	6.2
ResNet50 [126]	26	49	6.7
GoogLeNet [129]	6	22	6.7
VGGNet16 [130]	134	13+2	6.8
AlexNet [4]	57	5+2	15.3

a: OMNIDIRECTIONAL AND EVENT CAMERA-BASED PERCEPTION

360 degree vision, or at least panoramic vision, is necessary for higher levels of automation. This can be achieved through camera arrays, though precise extrinsic calibration between each camera is then necessary to make image stitching possible. Alternatively, omnidirectional cameras can be used, or a smaller array of cameras with very wide angle *fisheye* lenses. These are however difficult to intrinsically calibrate; the spherical images are highly distorted and the camera model used must account for mirror reflections or fisheye lens distortions, depending on the camera model producing the panoramic images [141], [142]. The accuracy of the model and calibration dictates the quality of undistorted images produced, on which the aforementioned 2D vision algorithms are used. An example of fisheye lenses producing two spherical images then combined into one panoramic image is shown in Figure 3. Some distortions inevitably remain, but despite these challenges in calibration, omnidirectional cameras have been used for many applications such as SLAM [143] and 3D reconstruction [144].

Event cameras are a fairly new modality which output asynchronous *events* usually caused by movement in the observed scene, as shown in Figure 4. This makes the sensing modality interesting for dynamic object detection. The

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 13

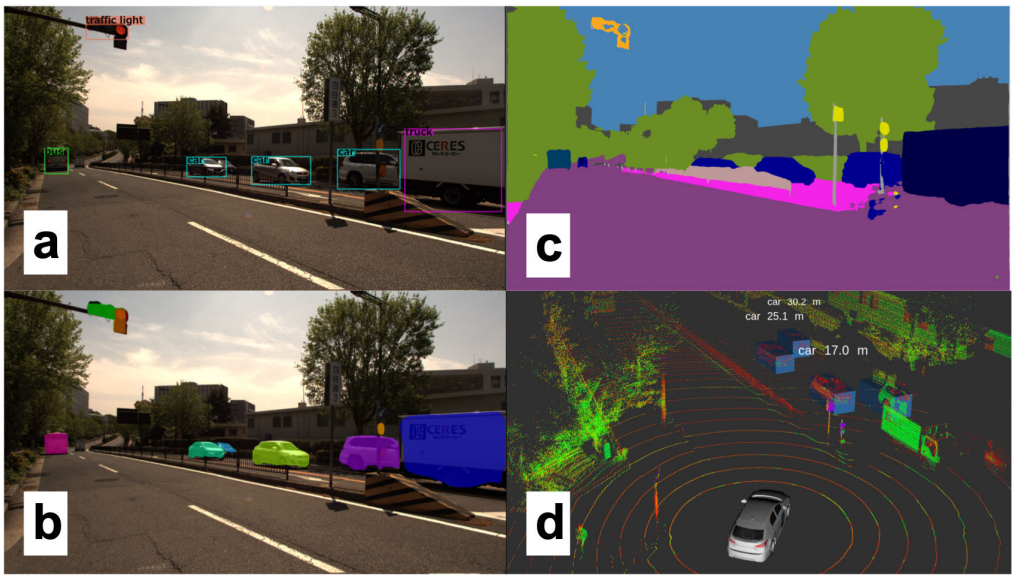


FIGURE 8. An urban scene near Nagoya University, with camera and lidar data collected by our experimental vehicle and object detection outputs from state-of-the-art perception algorithms. (a) A front facing camera's view, with bounding box results from YOLOv3 [128] and (b) instance segmentation results from MaskRCNN [138]. (c) Semantic segmentation masks produced by DeepLabv3 [139]. (d) The 3D Lidar data with object detection results from SECOND [140]. Amongst the four, only the 3D perception algorithm outputs range to detected objects.

other appealing factor is their response time on the order of microseconds [90], as frame rate is a significant limitation for high-speed driving. The sensor resolution remains an issue, but new models are rapidly improving. They have been used for a variety of applications closely related to ADS. A recent survey outlines progress in pose estimation and SLAM, visual-inertial odometry and 3D reconstruction, as well as other applications [145]. Most notably, a dataset for end-to-end driving with event cameras was recently published, with preliminary experiments showing that the output of an event camera can, to some extent, be used to predict car steering angle [89].

b: POOR ILLUMINATION AND CHANGING APPEARANCE

The main drawback with using camera is that changes in lighting conditions can significantly affect their performance. Low light conditions are inherently difficult to deal with, while changes in illumination due to shifting shadows, intemperate weather, or seasonal changes, can cause algorithms to fail, in particular supervised learning methods. For example, snow drastically alters the appearance of scenes and hides potentially key features such as lane markings. An easy alternative is to use an alternate sensing modalities for perception, but lidar also has difficulties with some weather conditions like fog and snow [146], and radars lack the necessary resolution for many perception tasks [51]. A sensor fusion strategy is often employed to avoid any single point of failure [147].

Thermal imaging through infrared sensors are also used for object detection in low light conditions, which is particularly effective for pedestrian detection [148]. Camera-only methods which attempt to deal with dynamic lighting conditions directly have also been developed. Both attempting to extract lighting invariant features [149] and assessing the quality of features [150] have been proposed. Pre-processed, *illumination invariant* images have applied to ADS [151] and were shown to improve localization, mapping and scene classification capabilities over long periods of time. Still, dealing with the unpredictable conditions brought forth by inadequate or changing illumination remains a central challenge preventing the widespread implementation of ADS.

2) SEMANTIC SEGMENTATION

Beyond image classification and object detection, computer vision research has also tackled the task of image segmentation. This consists of classifying each pixel of an image with a class label. This task is of particular importance to driving automation as some objects of interest are poorly defined by bounding boxes, in particular roads, traffic lines, sidewalks and buildings. A segmented scene in an urban area can be seen in Figure 8. As opposed to semantic segmentation, which labels pixels based on a class, instance segmentation algorithms further separates instances of the same class, which is important in the context of driving automation. In other words, objects which may have different trajectories and behaviors must be differentiated from each other. We used the COCO dataset [152] to train the instance segmentation algorithm Mask R-CNN [138] with the sample result shown in Figure 8.

Segmentation has recently become feasible for real-time applications. Generally, developments in this field progress

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 14

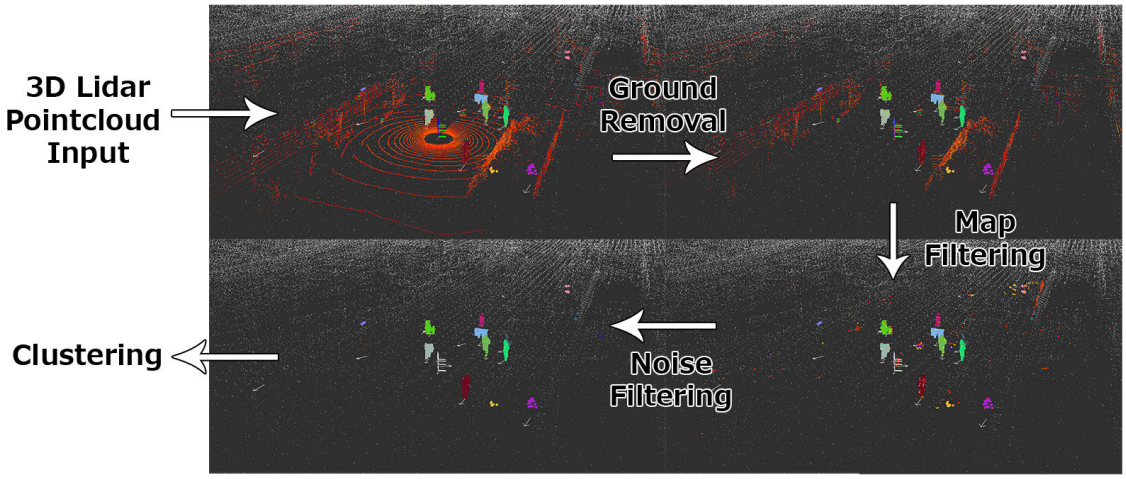


FIGURE 9. Outline of a traditional method for object detection from 3D pointcloud data. Various filtering and data reduction methods are used first, followed by clustering. The resulting clusters are shown by the different colored points in the 3D lidar data of pedestrians collected by our data collection platform.

in parallel with image-based object detection. The aforementioned Mask R-CNN [138] is a generalization of Faster R-CNN [153]. The multi-task R-CNN network can achieve accurate bounding box estimation and instance segmentation simultaneously and can also be generalized to other tasks like pedestrian pose estimation with minimal domain knowledge. Running at 5 fps means it is approaching the area of real-time use for ADS.

Unlike Mask-RCNN's architecture which is more akin to those used for object detection through its use of region proposal networks, segmentation networks usually employ a combination of convolutions for feature extraction. Those are followed by deconvolutions, also called transposed convolutions, to obtain pixel resolution labels [154], [155]. Feature pyramid networks are also commonly used, for example in PSPNet [156], which also introduced dilated convolutions for segmentation. This idea of sparse convolutions was then used to develop DeepLab [157], with the most recent version being the current state-of-the-art for object segmentation [139]. We employed DeepLab with our ADS and a segmented frame is shown in Figure 8.

While most segmentation networks are as of yet too slow and computationally expensive to be used in ADS, it is important to notice that many of these segmentations networks are initially trained for different tasks, such as bounding box estimation, then generalized to segmentation networks. Furthermore, these networks were shown to learn universal feature representations of images, and can be generalized for many tasks. This suggests the possibility that single, generalized perception networks may be able to tackle all perception tasks required for an ADS.

3) 3D OBJECT DETECTION

Given their affordability, availability and widespread research, cameras are used by nearly all algorithms presented so far as the primary perception modality. However, cameras have limitations that are critical to ADS. Aside from illumination which was previously discussed, camera-based object detection occurs in the projected image space and therefore the scale of the scene is unknown. To make use of this information for dynamic driving tasks like obstacle avoidance, it is necessary to bridge the gap from 2D image-based detection to the 3D, metric space. Depth estimation is therefore necessary, which is in fact possible with a single camera [158] though stereo or multi-view systems are more robust [159]. These algorithms necessarily need to solve an expensive image matching problem, which adds a significant amount of processing cost to an already complex perception pipeline.

A relatively new sensing modality, the 3D lidar, offers an alternative for 3D perception. The 3D data collected inherently solves the scale problem, and since they have their own emission source, they are far less dependable on lighting condition, and less susceptible to intemperate weather. The sensing modality collects sparse 3D points representing the surfaces of the scene, as shown in Figure 9, which are challenging to use for object detection and classification. The appearance of objects change with range, and after some distance, very few data points per objects are available to detect an object. This poses some challenges for detection, but since the data is a direct representation of the world, it is more easily separable. Traditional methods often use euclidean clustering [160] or region-growing methods [161] for grouping points into objects. This approach has been made much more robust through various filtering techniques, such as ground filtering [162] and map-based filtering [163]. We implemented a 3D object detection pipeline to get clustered objects from raw point cloud input. An example of this process is shown in Figure 9.

As with image-based methods, machine learning has also recently taken over 3D detection methods. These methods

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 15

FIGURE 10. A depth image produced from synthetic lidar data, generated in the CARLA [164] simulator.

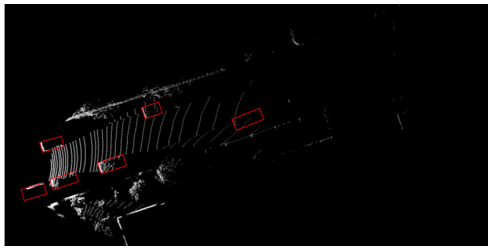


FIGURE 11. Bird's eye view perspective of 3D lidar data, a sample from the KITTI dataset [177].

TABLE 7. Average Precision (AP) in % on the KITTI 3D object detection test set *car* class, ordered based on *moderate* category accuracy. These algorithms only use pointcloud data.

Algorithm	T [s]	Easy	Moderate	Hard
PointRCNN [179]	0.10	85.9	75.8	68.3
PointPillars [180]	0.02	79.1	75.0	68.3
SECOND [140]	0.04	83.1	73.7	66.2
IPOD [181]	0.20	82.1	72.6	66.3
F-PointNet [182]	0.17	81.2	70.4	62.2
VoxelNet (Lidar) [167]	0.23	77.5	65.1	57.7
MV3D (Lidar) [172]	0.24	66.8	52.8	51.3

have also notably been applied to RGB-D [165], which produce similar, but colored, point clouds; with their limited range and unreliability outdoors, RGB-D have not been used for ADS applications. A 3D representation of point data, through a 3D occupancy grid called voxel grids, was first applied for object detection in RGB-D data [165]. Shortly thereafter, a similar approach was used on point clouds created by lidars [166]. Inspired by image-based methods, 3D CNNs are used, despite being computationally very expensive.

The first convincing results for point cloud-only 3D bounding box estimation were produced by VoxelNet [167]. Instead of hand-crafting input features computed during the discretization process, VoxelNet learned an encoding from raw point cloud data to voxel grid. Their voxel feature encoder (VFE) uses a fully connected neural network to convert the variable number of points in each occupied voxel to a feature vector of fixed size. The voxel grid encoded with feature vectors was then used as input to an aforementioned RPN for multi-class object detection. This work was then improved both in terms of accuracy and computational efficiency by SECOND [140] by exploiting the natural sparsity of lidar data. We employed SECOND and a sample result is shown in Figure 8. Several algorithms have been produced recently, with accuracy constantly improving as shown in Table 7, yet the computational complexity of 3D convolutions remains an issue for real-time use.

Another option for lidar-based perception is 2D projection of point cloud data. There are two main representations of point cloud data in 2D, the first being a so-called depth image shown in Figure 10, largely inspired by camera-based methods that perform 3D object detection through depth estimation [168] and methods that operate on RGB-D data [169]. The VeloFCN network [170] proposed to use single-channel depth image as input to a shallow, single-stage convolutional neural network which produced 3D vehicle proposals, with many other algorithms adopting this approach. Another use of depth image was shown for semantic classification of lidar points [171].

The other 2D projection that has seen increasing popularity, in part due to the new KITTI benchmark, is projection to bird's eye view (BV) image. This is a top-view image of point clouds as shown in Figure 11. Bird's eye view images discretize space purely in 2D, so lidar points which vary in height alone occlude each other. The MV3D algorithm [172] used camera images, depth images, as well as multi-channel BV images; each channel corresponding to a different range of heights, so as to minimize these occlusions. Several other works have reused camera-based algorithms and trained efficient networks for 3D object detection on 2D BV images [173]–[176]. State-of-the-art algorithms are currently being evaluated on the KITTI dataset [177] and nuScenes dataset [178] as they offer labeled 3D scenes. Table 7 shows the leading methods on the KITTI benchmark, alongside detection times. 2D methods are far less computationally expensive, but recent methods that take point sparsity into account [140] are real-time viable and rapidly approaching the accuracy necessary for integration in ADSs.

Radar: Radar sensors have already been used for various perception applications, in various types of vehicles, with different models operating at complementary ranges. While not as accurate as the lidar, it can detect at object at high range and estimate their velocity [113]. The lack of precision for estimating shape of objects is a major drawback when it is used in perception systems [51], the resolution is simply too low. As such, it can be used for range estimation to large objects like vehicles, but it is challenging for pedestrians or static objects. Another issue is the very limited field of view of most radars, forcing a complicated array of radar sensors to cover the full field of view. Nevertheless, radar have seen widespread use as an ADAS component, for applications including proximity warning and adaptive cruise control [146]. While radar and lidar are often seen as competing sensing modalities, they will likely be used in tandem in fully automated driving systems. Radars are very long range,

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 16

have low cost and are robust to poor weather, while lidar offer precise object localization capabilities, as discussed in Section IV.

Another similar sensor to the radar are sonar devices, though their extremely short range of $<2m$ and poor angular resolution makes their use limited to very near obstacle detection [146].

B. OBJECT TRACKING

Object tracking is also often referred to as multiple object tracking (MOT) [183] and detection and tracking of multiple objects (DATMO) [184]. For fully automated driving in complex and high speed scenarios, estimating location alone is insufficient. It is necessary to estimate dynamic objects' heading and velocity so that a motion model can be applied to track the object over time and predict future trajectory to avoid collisions. These trajectories must be estimated in the vehicle frame to be used by planning, so range information must be obtained through multiple camera systems, lidars or radar sensors. 3D lidars are often used for their precise range information and large field of view, allowing tracking over longer periods of time. To better cope with the limitations and uncertainties of different sensing modalities, a sensor fusion strategy is often use for tracking [47].

Commonly used object trackers rely on simple data association techniques followed by traditional filtering methods. When objects are tracked in 3D space at high frame rate, nearest neighbor methods are often sufficient for establishing associations between objects. Image-based methods, however, need to establish some appearance model, which may consider the use of color histograms, gradients and other features such as KLT to evaluate the similarity [185]. Point cloud based methods may also use similarity metrics such as point density and Hausdorff distance [163], [186]. Since association errors are always a possibility, multiple hypothesis tracking algorithms [187] are often employed, which ensures tracking algorithms can recover from poor data association at any single time step. Using occupancy maps as a frame for all sensors to contribute to and then doing data association in that frame is common, especially when using multiple sensors [188]. To obtain smooth dynamics, the detection results are filtered by traditional Bayes filters. Kalman filtering is sufficient for simple linear models, while the extended and and unscented Kalman filters [189] are used to handle nonlinear dynamic models [190]. We implemented a basic particle filter based object-tracking algorithm, and an example of tracked pedestrians in contrasting camera and 3D lidar perspective is shown in Figure 12.

Physical models for the object being tracked are also often used for more robust tracking. In that case, non-parametric methods such as particle filters are used, and physical parameters such as the size of the object are tracked alongside dynamics [191]. More involved filtering methods such as Rao-Blackwellized particle filters have also been used to keep track of both dynamic variables and vehicle geometry variables for an L-shape vehicle model [192]. Various models

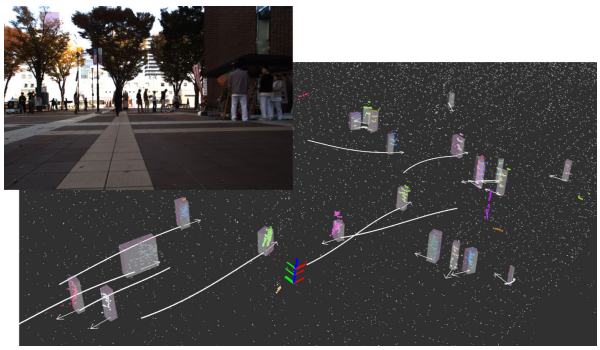


FIGURE 12. A scene with several tracked pedestrians and cyclist with a basic particle filter on an urban road intersection. Past trajectories are shown in white with current heading and speed shown by the direction and magnitude of the arrow, sample collected by our data collection platform.

have been proposed for vehicles and pedestrians, while some models generalize to any dynamic object [193].

Finally, deep learning has also been applied to the problem of tracking, particularly for images. Tracking in monocular images was achieved in real-time through a CNN-based method [194], [195]. Multi-task network which estimate object dynamics are also emerging [196] which further suggests that generalized networks tackling multiple perception tasks may be the future of ADS perception.

C. ROAD AND LANE DETECTION

Bounding box estimation methods previously covered are useful for defining some objects of interest but are inadequate for continuous surfaces like roads. Determining the drivable surface is critical for ADSs and has been specifically researched as a subset of the detection problem. While drivable surfaces can be determined through semantic segmentation, automated vehicles need to understand road semantics to properly negotiate the road. An understanding of lanes, and how they are connected through merges and intersections remains a challenge from the perspective of perception. In this section, we provide an overview of current methods used for road and lane detection, and refer the reader to in-depth surveys of traditional methods [198] and the state-of-the-art methods [199], [200].

This problem is usually subdivided in several tasks, each unlocking some level of automation. The simplest is determining the drivable area from the perspective of the ego-vehicle. The road can then be divided into lanes, and the vehicles' host lane can be determined. Host lane estimation over a reasonable distance allows ADAS technology such as lane departure warning, lane keeping and adaptive cruise control [198], [201]. Even more challenging is determining other lanes and their direction [202], and finally understanding complex semantics, as in their current and future direction, or merging and turning lanes [47]. These ADAS or ADS technologies have different criteria both in terms of task, detection distance and reliability rates, but fully automated driving will require a complete, semantic understanding of

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 17

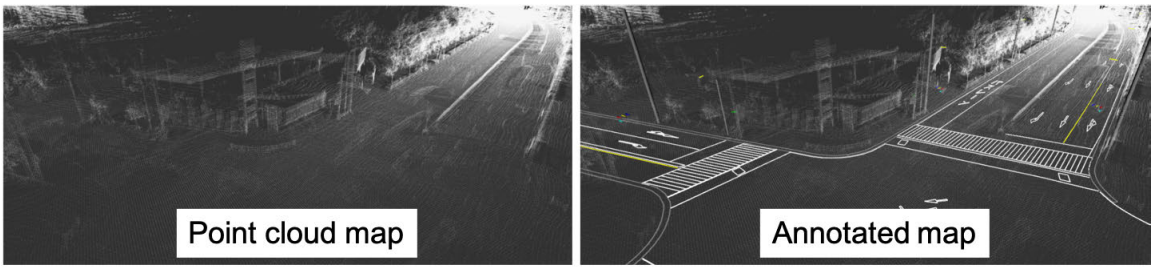


FIGURE 13. Annotating a 3D point cloud map with topological information. A large number of annotators were employed to build the map shown on the right-hand side. The point-cloud and annotated maps are available on [197].

road structures and the ability to detect several lanes at long ranges [199]. Annotated maps as shown in Figure 13 are extremely useful for understanding lane semantics.

Current methods on road understanding typically first rely on exteroceptive data preprocessing. When cameras are used, this usually means performing image color corrections to normalize lighting conditions [203]. For lidar, several filtering methods can be used to reduce clutter in the data such as ground extraction [162] or map-based filtering [163]. For any sensing modality, identifying dynamic objects which conflicts with the static road scene is an important pre-processing step. Then, road and lane feature extraction is performed on the corrected data. Color statistics and intensity information [204], gradient information [205], and various other filters have been used to detect lane markings. Similar methods have been used for road estimation, where the usual uniformity of roads and elevation gap at the edge allows for region growing methods to be applied [206]. Stereo camera systems [207], as well as 3D lidars [204], have been used determine the 3D structure of roads directly. More recently, machine learning-based methods which either fuse maps with vision [200] or use fully appearance-based segmentation [208] have been used.

Once surfaces are estimated, model fitting is used to establish the continuity of the road and lanes. Geometric fitting through parametric models such as lines [209] and splines [204] have been used, as well as non-parametric continuous models [210]. Models that assume parallel lanes have been used [201], and more recently models integrating topological elements such as lane splitting and merging were proposed [204].

Temporal integration completes the road and lane segmentation pipeline. Here, vehicle dynamics are used in combination with a road tracking system to achieve smooth results. Dynamic information can also be used alongside Kalman filtering [201] or particle filtering [207] to achieve smoother results.

Road and lane estimation is a well-researched field and many methods have already been integrated successfully for lane keeping assistance systems. However, most methods remain riddled with assumptions and limitations, and truly general systems which can handle complex road topologies have yet to be developed. Through standardized road maps which encode topology and emerging machine learning-based road and lane classification methods, robust systems for driving automation are slowly taking shape.

VI. ASSESSMENT

A robust ADS should constantly evaluate the overall risk level of the situation and predict the intentions of surrounding human drivers and pedestrians. A lack of acute assessment mechanism can lead to accidents. This section discusses assessment under three subcategories: overall risk and uncertainty assessment, human driving behavior assessment, and driving style recognition.

A. RISK AND UNCERTAINTY ASSESSMENT

Overall assessment can be summarized as quantifying the uncertainties and the risk level of the driving scene. It is a promising methodology that can increase the safety of ADS pipelines [31].

Using Bayesian methods to quantify and measure uncertainties of deep neural networks was proposed in [212]. A Bayesian deep learning architecture was designed for propagating uncertainty throughout an ADS pipeline, and the advantage of it over conventional approaches was shown in a hypothetical scenario [31]. In summary, each module conveys and accepts probability distributions instead of exact outcomes throughout the pipeline, which increases the overall robustness of the system.

An alternative approach is to assess the overall risk level of the driving scene separately, i.e outside the pipeline. Sensory inputs were fed into a risk inference framework in [83], [213] to detect unsafe lane change events using Hidden Markov Models (HMMs) and language models. Recently, a deep spatiotemporal network that infers the overall risk level of a driving scene was introduced in [211]. Implementation of this method is available open-source.[1] We employed this method to assess the risk level of a lane change as shown in Figure 14.

B. SURROUNDING DRIVING BEHAVIOR ASSESSMENT

Understanding surrounding human driver intention is most relevant to medium to long term prediction and decision making. In order to increase the prediction horizon of

[1] https://github.com/Ekim-Yurtsever/DeepTL-Lane-Change-Classification

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 18

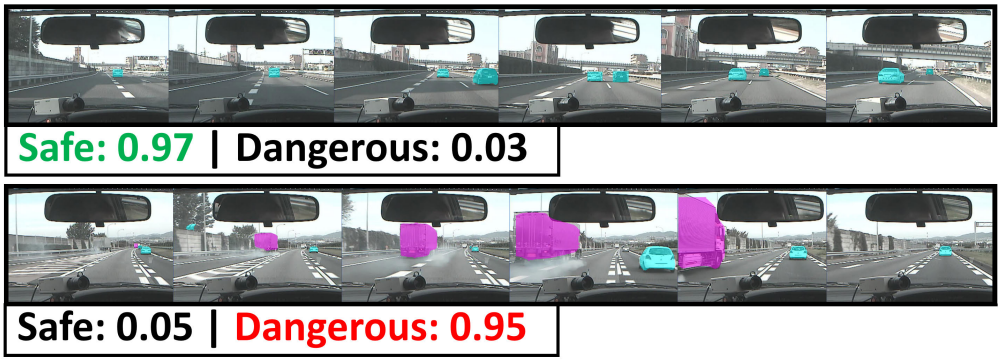


FIGURE 14. Assessing the overall risk level of driving scenes. We employed an open-source[1] deep spatiotemporal video-based risk detection framework [211] to assess the image sequences shown in this figure.

surrounding object behavior, human traits should be considered and incorporated into the prediction and evaluation steps. Understanding surrounding driver intention from the perspective of an ADS is not a common practice in the field, as such, state-of-the-art is not established yet.

In [214], a target vehicle's future behavior was predicted with a hidden Markov model (HMM) and the prediction time horizon was extended 56% by learning human driving traits. The proposed system tagged observations with predefined maneuvers. Then, the features of each type were learned in a data-centric manner with HMMs. Another learning based approach was proposed in [215], where a Bayesian network classifier was used to predict maneuvers of individual drivers on highways. A framework for long term driver behavior prediction using a combination of a hybrid state system and HMM was introduced in [216]. Surrounding vehicle information was integrated with ego-behavior through a symbolization framework in [83], [213]. Detecting dangerous cut in maneuvers was achieved with an HMM framework that was trained on safe and dangerous data in [217]. Lane change events were predicted 1.3 seconds in advance with support vector machines (SVM) and Bayesian filters [218].

The main challenges are the short observation window for understanding the intention of humans and real-time high-frequency computation requirements. ADSs can typically only observe a surrounding vehicle only for seconds. Complicated driving behavior models that require longer observation periods cannot be utilized under these circumstances.

C. DRIVING STYLE RECOGNITION

In 2016, Google's self-driving car collided with an oncoming bus [8] during a lane change where it assumed that the bus driver was going to yield. However, the bus driver accelerated instead. This accident may have been prevented if the ADS understood the bus driver's individual, unique driving style and predicted his behavior.

Driving style is a broad term without an established common definition. Furthermore, recognizing the surrounding human driving styles is a severely understudied topic. However, thorough reviews of driving style categorization of *human-driven* ego vehicles can be found in [220] and in [221]. Readers are referred to these papers for a complete review. The remainder of this subsection gives a brief overview of *human-driven* ego vehicle-based driving style recognition.

TABLE 8. Driving style categorization of human-driven ego vehicles.

Related work	# Classes	Methodology	Class details
[238]	5	PCA	non-aggressive to very aggressive
[239]	3	NN, SVM, DT	expert/typical/low-skill
[234]	3	FL	sporty/normal/comfortable
[235]	3	PCMLP	aggressive/moderate/calm
[240]	3	SAE & K-means	unidentified clusters
[83]	2	non-param. Bayesian	risky/safe
[222]	2	DTW	aggressive/non-aggressive
[223]	2	RB	sudden/safe
[237]	Continuous $[-1, 1]$	NN	mild to active

Typically, driving style is defined with respect to either aggressiveness [222]–[226] or fuel consumption [227]–[231]. For example, [232] introduced a rule-based model that classified driving styles with respect to jerk. This model decides whether a maneuver is aggressive or calm by a set of rules and jerk thresholds. Drivers were categorized with respect to their average speed in [233]. In conventional methods, total number and meaning of driving style classes are predefined beforehand. The vast majority of driving style recognition literature uses two [83], [213], [222], [223], [227] or three [234]–[236] classes. Representing driving style in a continuous domain is uncommon, but there are some studies. In [237], driving style was depicted as a continuous value between -1 and $+1$, which stands for mild and active respectively. Details of classification methods are given in Table 8.

More recently, machine learning based approaches have been utilized for driving style recognition. Principal component analysis was used and five distinct driving classes were detected in an unsupervised manner in [238] and a

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 19

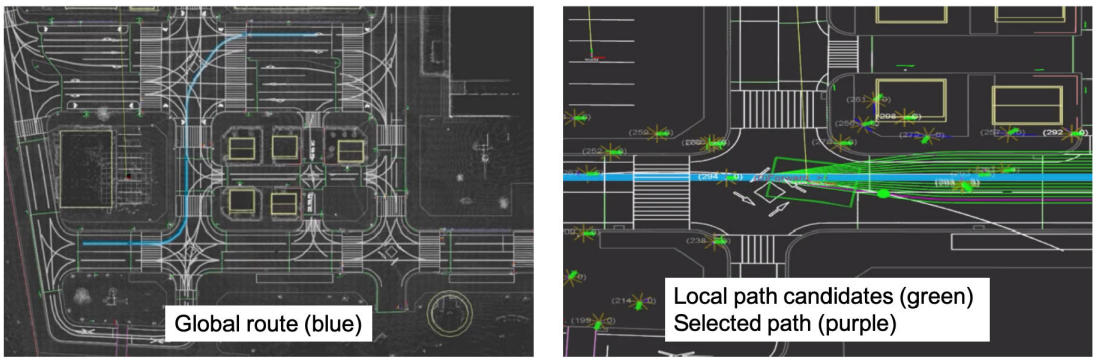


FIGURE 15. Global plan and the local paths. The annotated vector map shown in Figure 13 was utilized by the planner. We employed OpenPlanner [219], which is a graph-based planner, to illustrate a typical planning approach.

GMM based driver model was used to identify individual drivers with success in [241]. Car-following and pedal operation behavior was investigated separately in the latter study. Another GMM based driving style recognition model was proposed for electric vehicle range prediction in [242]. In [222], aggressive event detection with dynamic time warping was presented where the authors reported a high success score. Bayesian approaches were utilized in [243] for modeling driving style on roundabouts and in [244] to asses critical braking situations. Bag-of-words and K-means clustering was used to represent individual driving features in [245]. A stacked autoencoder was used to extract unique driving signatures from different drivers, and then macro driving style centroids were found with clustering [240]. Another autoencoder network was used to extract road-type specific driving features [246]. Similarly, driving behavior was encoded in a 3-channel RGB space with a deep sparse autoencoder to visualize individual driving styles [247].

A successful integration of driving style recognition into a real world ADS pipeline is not reported yet. However, these studies are promising and point to a possible new direction in ADS development.

VII. PLANNING AND DECISION MAKING

Planning can be divided into two sub-tasks: global route planning and local path planning. Figure 15 illustrates a typical planning approach in detail.

The remainder of this section gives a brief overview of the subject. For more information studies such as [18], [23], [248] can be referred.

A. GLOBAL PLANNING

The global planner is responsible for finding the route on the road network from origin to the final destination. The user usually defines the final destination. Global navigation is a well-studied subject, and high performance has become an industry standard for more than a decade. Almost all modern production cars are equipped with navigation systems that utilize GPS and offline maps to plan a global route.

Route planning is formulated as finding the point-to-point shortest path in a directed graph, and conventional methods are examined under four categories in [248]. These are; goal-directed, separator-based, hierarchical and bounded-hop techniques. A* search [249] is a standard goal-directed path planning algorithm and used extensively in various fields for almost 50 years.

The main idea of separator-based techniques is to remove a subset of vertices [250] or arcs from the graph and compute an overlay graph over it. Using the overlayed graph to calculate the shortest path results in faster queries.

Hierarchical techniques take advantage of the road hierarchy. For example, the road hierarchy in the US can be listed from top to bottom as freeways, arterials, collectors and local roads respectively. For a route query, the importance of hierarchy increases as the distance between origin and destination gets longer. The shortest path may not be the fastest nor the most desirable route anymore. Getting away from the destination thus making the route a bit longer to take the closest highway ramp may result in faster travel time in comparison to following the shortest path of local roads. Contraction Hierarchies (CH) method was proposed in [251] for exploiting road hierarchy.

Precomputing distances between selected vertexes and utilizing them on the query time is the basis of bounded-hop techniques. Precomputed shortcuts can be utilized partly or exclusively for navigation. However, the naive approach of precomputing all possible routes from every pair of vertices is impractical in most cases with large networks. One possible solution to this is to use hub labeling (HL) [252]. This approach requires preprocessing also. A label associated with a vertex consists of nearby *hub* vertices and the distance to them. These labels satisfy the condition that at least one shared hub vertex must exist between the labels of any given two vertices. HL is the fastest query time algorithm for route planning [248], in the expense of high storage usage.

A combination of the above algorithms are popular in state-of-the-art systems. For example, [253] combined a separator with a bounded-hop method and created the Transit Node

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 20

TABLE 9. Local planning techniques.

Approach	Methods	Pros and cons
Graph search	Dijkstra [254], A* [249], State lattice [255]	Slow and jerky
Sampling based	RPP [256], RRT [257], RRT* [258], PRM [259]	Fast solution but jerky
Curve interpolation	clothoids [260], polynomials [261], Bezier [262], splines [104]	Smooth but slow
Numerical optimization	num. non-linear opt. [263], Newton's method [264]	increases computational cost but improves quality
Deep learning	FCN [265], segmentation network [266]	high imitation performance, but no hard coded safety measures

Routing with Arc Flags (TNR + AF) algorithm. Modern route planners can make a query in milliseconds.

B. LOCAL PLANNING

The objective of the local planner is to execute a global plan without *failing*. In other words, in order to complete its trip, the ADS must find trajectories to avoid obstacles and satisfy optimization criteria in the configuration space (C-space), given a starting and destination point. A detailed local planning review is presented in [19] where the taxonomy of motion planning was divided into four groups; graph-based planners, sampling-based planners, interpolating curve planners and numerical optimization approaches. After a summary of these conventional planners, the emerging deep learning-based planners are introduced at the end of this section. Table 9 gives a brief summary of local planning methods.

Graph-based local planners use the same techniques as graph-based global planners such as Dijkstra [254] and A* [249], which output discrete paths rather than continuous ones. This can lead to jerky trajectories [19]. A more advanced graph-based planner is the state lattice algorithm. As all graph-based methods, state lattice discretizes the decision space. High dimensional lattice nodes, which typically encode 2D position, heading and curvature [255], are used to create a grid first. Then, the connections between the nodes are precomputed with an inverse path generator to build the state lattice. During the planning phase, a cost function, which usually considers proximity to obstacles and deviation from the goal, is utilized for finding the best path with the precomputed path primitives. State lattices can handle high dimensions and is good for local planning in dynamical environments, however, the computational load is high and the discretization resolution limits the planners' capacity [19].

A detailed overview of Sampling Based Planning (SBP) methods can be found in [267]. In summary, SBP tries to build the connectivity of the C-space by randomly sampling paths in it. Randomized Potential Planner (RPP) [256] is one of the earliest SBP approaches, where random walks are generated for escaping local minimums. Probabilistic roadmap method (PRM) [259] and rapidly-exploring random tree (RRT) [257] are the most commonly used SBP algorithms. PRM first samples the C-space during its learning phase and then makes a query with the predefined origin and destination points on the roadmap. RRT, on the other hand, is a single query planner. The path between start and goal configuration is incrementally built with random tree-like branches. RRT is faster than PRM and both are probabilistically complete [257], which means a path that satisfies the given conditions will be guaranteed to be found with *enough* runtime. RRT* [258], an extension of the RRT, provides more optimal paths instead of completely random ones while sacrificing computational efficiency. The main disadvantage of SBP in general is, again, the jerky trajectories [19].

Interpolating curve planners fit a curve to a known set of points [19], e.g. way-points generated from the global plan or a discrete set of future points from another local planner. The main obstacle avoidance strategy is to interpolate new collision-free paths that first deviate from, and then re-enter back to the initial planned trajectory. The new path is generated by fitting a curve to a new set of points: an exit point from the currently traversed trajectory, newly sampled collision free points, and a re-entry point on the initial trajectory. The resultant trajectory is smooth, however, the computational load is usually higher compared to other methods. There are various curve families that are used commonly such as clothoids [260], polynomials [261], Bezier curves [262] and splines [104].

Optimization based motion planners improve the quality of already existing paths with optimization functions. A* trajectories were optimized with numeric non-linear functions in [263]. Potential Field Method (PFM) was improved by solving the inherent oscillation problem using Newton's method with obtaining C_1 continuity in [264].

Recently, Deep Learning (DL) and reinforcement learning based local planners started to emerge as an alternative. Fully convolutional 3D neural networks can generate future paths from sensory input such as lidar point clouds [265]. An interesting take on the subject is to segment image data with path proposals using a deep segmentation network [266]. Planning a safe path in occluded intersections was achieved in a simulation environment using deep reinforcement learning in [268]. The main difference between end-to-end driving and deep learning based local planners is the output: the former outputs direct vehicle control signals such as steering and pedal operation, whereas the latter generates a trajectory. This enables DL planners to be integrated into conventional pipelines [24].

Deep learning based planners are promising, but they are not widely used in real-world systems yet. Lack of hard-coded safety measures, generalization issues, need for labeled data are some of the issues that need to be addressed.

VIII. HUMAN MACHINE INTERACTION

Vehicles communicate with their drivers/passengers through their HMI module. The nature of this communication greatly

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 21

depends on the objective, which can be divided into two: primary driving tasks and secondary tasks. The interaction intensity of these tasks depend on the automation level. Where a manually operated, level zero conventional car requires constant user input for operation, a level five ADS may need user input only at the beginning of the trip. Furthermore, the purpose of interaction may affect intensity. A shift from executing primary driving tasks to monitoring the automation process raises new HMI design requirements.

There are several investigations such as [269], [270] about automotive HMI technologies, mostly from the distraction point of view. Manual user interfaces for secondary tasks are more desired than their visual counterparts [269]. The main reason is vision is absolutely necessary and has no alternative for primary driving tasks. Visual interface interactions require glances with durations between 0.6 and 1.6 seconds with a mean of 1.2 seconds [269]. As such, secondary task interfaces that require vision is distracting and detrimental for driving.

Auditory User Interfaces (AUI) are good alternatives to visually taxing HMI designs. AUIs are omni-directional: even if the user is not attending, the auditory cues are hard to miss [271]. The main challenge of audio interaction is automatic speech recognition (ASR). ASR is a very mature field. However, in vehicle domain there are additional challenges; low performance caused by uncontrollable cabin conditions such as wind and road noise [272]. Beyond simple voice commands, conversational natural language interaction with an ADS is still an unrealized concept with many unsolved challenges [273].

The biggest HMI challenge is at automation level three and four. The user and the ADS need to have a mutual understanding, otherwise, they will not be able to grasp each other's intentions [270]. The transition from manual to automated driving and vice versa is prone to fail in the state-of-the-art. Recent research showed that drivers exhibit low cognitive load when monitoring automated driving compared to doing a secondary task [288]. Even though some experimental systems can recognize driver-activity with a driver facing camera based on head and eye-tracking [289], and prepare the driver for handover with visual and auditory cues [290] in simulation environments, a real world system with an efficient handover interaction module does not exist at the moment. This is an open problem [291] and future research should focus on delivering better methods to inform/prepare the driver for easing the transition [41].

IX. DATASETS AND AVAILABLE TOOLS

A. DATASETS AND BENCHMARKS

Datasets are crucial for researchers and developers because most of the algorithms and tools have to be tested and trained before going on road.

Typically, sensory inputs are fed into a stack of algorithms with various objectives. A common practice is to test and validate these functions separately on annotated datasets. For example, the output of cameras, 2D vision, can be fed into an object detection algorithm to detect surrounding vehicles and pedestrians. Then, this information can be used in another algorithm for planning purposes. Even though these two algorithms are connected in the stack of this example, the object detection part can be worked on and validated separately during the development process. Since computer vision is a well-studied field, there are annotated datasets for object detection and tracking specifically. The existence of these datasets increases the development process and enables interdisciplinary research teams to work with each other much more efficiently. For end-to-end driving, the dataset has to include additional ego-vehicle signals, chiefly steering and longitudinal control signals.

As learning approaches emerged, so did training datasets to support them. The PASCAL VOC dataset [292], which grew from 2005 to 2012, was one of the first dataset featuring a large amount of data with relevant classes for ADS. However, the images often featured single objects, in scenes and scales that are not representative of what is encountered in driving scenarios. In 2012, the KITTI Vision Benchmark [177] remedied this situation by providing a relatively large amount of labeled driving scenes. It remains as one of the most widely used datasets for applications related to driving automation. Yet in terms of quantity of data and number of labeled classes, it is far inferior to generic image databases such as ImageNet [131] and COCO [152]. While no doubt useful for training, these generic image databases lack the adequate context to test the capabilities of ADS. UC Berkeley DeepDrive [275] is a recent dataset with annotated image data. The Oxford RobotCar [53] was used to collect over 1000 km of driving data with six cameras, lidar, GPS and INS in the UK but is not annotated. ApolloScape is a very recent dataset that is not fully public yet [278]. Cityscapes [274] is commonly used for computer vision algorithms as a benchmark set. Mapillary Vistas is a big image dataset with annotations [276]. TorontoCity benchmark [286] is a very detailed dataset; however it is not public yet. The nuScenes dataset is the most recent urban driving dataset with lidar and image sensors [178]. Comma.ai has released a part of their dataset [293] which includes 7.25 hours of driving. In DDD17 [89] around 12 hours of driving data is recorded. The LiVi-Set [281] is a new dataset that has lidar, image and driving behavior. CommonRoad [294] is a new benchmark for motion-planning.

Naturalistic driving data is another type of dataset that concentrates on the individual element of the driving: the driver. SHRP2 [283] includes over 3000 volunteer participants' driving data over a 3-year collection period. Other naturalistic driving datasets are the 100-Car study [284], euroFOT [285] and NUDrive [282]. Table 10 shows the comparison of these datasets.

B. OPEN-SOURCE FRAMEWORKS AND SIMULATORS

Open source frameworks are very useful for both researchers and the industry. These frameworks can "democratize" ADS development. Autoware [122], Apollo [295], Nvidia DriveWorks [296] and openpilot [297] are amongst the most used

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 22

TABLE 10. Driving datasets.

Dataset	Image	LIDAR	2D annotation*	3D annotation*	ego signals	Naturalistic	POV	Multi trip	all weathers	day & night
Cityscapes [274]	✓	-	✓	-	-	-	Vehicle	-	-	-
Berkley DeepDrive [275]	✓	-	✓	-	-	-	Vehicle	-	✓	✓
Mapillary [276]	✓	-	✓	-	-	-	Vehicle	-	✓	✓
Oxford RobotCar [53]	✓	✓	-	-	-	-	Vehicle	✓	✓	✓
KITTI [177]	✓	✓	✓	✓	-	-	Vehicle	-	-	-
H3D [277]	✓	✓	-	✓	-	-	Vehicle	-	-	-
ApolloScape [278]	✓	✓	✓	✓	-	-	Vehicle	-	-	-
nuScenes [178]	✓	✓	✓	✓	-	-	Vehicle	-	✓	✓
Udacity [279]	✓	✓	✓	✓	-	-	Vehicle	-	-	-
DDD17 [89]	✓	-	✓	-	✓	-	Vehicle	-	✓	✓
Comma2k19 [280]	✓	-	-	-	✓	-	Vehicle	-	-	✓
LiVi-Set [281]	✓	✓	-	-	✓	-	Vehicle	-	-	-
NU-drive [282]	✓	-	-	-	✓	Semi	Vehicle	✓	-	-
SHRP2 [283]	✓	-	-	-	✓	✓	Vehicle	-	-	-
100-Car [284]	✓	-	-	-	✓	✓	Vehicle	-	✓	✓
euroFOT [285]	✓	-	-	-	✓	✓	Vehicle	-	-	-
TorontoCity [286]	✓	✓	✓	✓	-	-	Aerial, panorama, vehicle	-	-	-
KAIST multi-spectral [287]	✓	✓	✓	-	-	-	Vehicle	-	-	✓

*2D and 3D annotation can vary from bounding boxes to segmentation masks. Readers are referred to sources for details of the datasets.

software-stacks capable of running an ADS platform in real world. We utilized Autoware [122] to realize core automated driving functions in this study.

Simulations also have an important place for ADS development. Since the instrumentation of an experimental vehicle still has a high cost and conducting experiments on public road networks are highly regulated, a simulation environment is beneficial for developing certain algorithms/modules before road tests. Furthermore, highly dangerous scenarios such as a collision with pedestrian can be tested in simulations with ease. CARLA [164] is an urban driving simulator developed for this purpose. TORCS [298] was developed for race track simulation. Some researchers even used computer games such as Grand Theft Auto V [299]. Gazebo [300] is a common simulation environment for robotics. For traffic simulations, SUMO [301] is a widely used open-source platform. [302] proposed different concepts of integrating real-world measurements into the simulation environment.

X. CONCLUSIONS

In this survey on automated driving systems, we outlined some of the key innovations as well as existing systems. While the promise of automated driving is enticing and already marketed to consumers, this survey has shown there remains clear gaps in the research. Several architecture models have been proposed, from fully modular to completely end-to-end, each with their own shortcomings. The optimal sensing modality for localization, mapping and perception is still disagreed upon, algorithms still lack accuracy and efficiency, and the need for a proper online assessment has become apparent. Less than ideal road conditions are still an open problem, as well as dealing with intemperate weather. Vehicle-to-vehicle communication is still in its infancy, while centralized, cloud-based information management has yet to be implemented due to the complex infrastructure required. Human-machine interaction is an under-researched field with many open problems.

The development of automated driving systems relies on the advancements of both scientific disciplines and new technologies. As such, we discussed the recent research developments which are likely to have a significant impact on automated driving technology, either by overcoming the weakness of previous methods or by proposing an alternative. This survey has shown that through inter-disciplinary academic collaboration and support from industries and the general public, the remaining challenges can be addressed. With directed efforts towards ensuring robustness at all levels of automated driving systems, safe and efficient roads are just beyond the horizon. ∎

REFERENCES

[1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," Washington, DC, USA, Tech. Rep. DOT HS 812 115, 2015.

[2] T. J. Crayton and B. M. Meier, "Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy," *J. Transp. Health*, vol. 6, pp. 245–252, Sep. 2017.

[3] W. D. Montgomery, R. Mudge, E. L. Groshen, S. Helper, J. P. MacDuffie, and C. Carson, "America's workforce and the self-driving future: Realizing productivity gains and spurring economic growth," Securing America's Future Energy, Washington, DC, USA, Tech. Rep., 2018.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[5] B. Schwarz, "LIDAR: Mapping the world in 3D," *Nature Photon.*, vol. 4, no. 7, p. 429, 2010.

[6] S. Hecker, D. Dai, and L. Van Gool, "End-to-end learning of driving models with surround-view cameras and route planners," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 435–453.

[7] D. Lavrinc. *This is How Bad Self-Driving Cars Suck in Rain*. Accessed: Dec. 16, 2018. [Online]. Available: https://jalopnik.com/this-is-how-bad-self-driving-cars-suck-in-the-rain-1666268433

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 23

[8] A. Davies. *Google's Self-Driving Car Caused Its First Crash*. Accessed: Dec. 16, 2018. [Online]. Available: https://www.wired.com/2016/02/googles-self-driving-car-may-caused-first-crash/

[9] M. McFarland. *Who's Responsible When an Autonomous Car Crashes?* Accessed: Jun. 4, 2019. [Online]. Available: https://money.cnn.com/2016/07/07/technology/tesla-liability-risk/index.html

[10] T. B. Lee. *Autopilot Was Active When a Tesla Crashed Into a Truck, Killing Driver*. Accessed: May 19, 2019. [Online]. Available: https://arstechnica.com/cars/2019/05/feds-autopilot-was-active-during-deadly-march-tesla-crash/

[11] Eureka. *E! 45: Programme For a European Traffic System With Highest Efficiency and Unprecedented Safety*. Accessed: May 19, 2019. [Online]. Available: https://www.eurekanetwork.org/project/id/45

[12] B. Ulmer, "VITA II-active collision avoidance in real traffic," in *Proc. Intell. Vehicles Symp.*, 1994, pp. 1–6.

[13] M. Buehler, K. Iagnemma, and S. Singh, Eds., *The 2005 DARPA Grand Challenge: The Great Robot Race*, vol. 36. Berlin, Germany: Springer, 2007.

[14] D. Feng, C. Haase-Schuetz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," 2019, *arXiv:1902.07830*. [Online]. Available: http://arxiv.org/abs/1902.07830

[15] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 163–168.

[16] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: Approaches, lessons and challenges," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 368, no. 1928, pp. 4649–4672, Oct. 2010.

[17] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, Apr. 2018.

[18] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[19] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[20] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 384–406, Apr. 2018.

[21] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[22] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and cellular network technologies for V2X communications: A survey," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9457–9470, Dec. 2016.

[23] C. Badue, R. Guidolini, R. Vivacqua Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. Paixão, F. Mutz, L. Veronese, T. Oliveira-Santos, and A. F. De Souza, "Self-driving cars: A survey," 2019, *arXiv:1901.04407*. [Online]. Available: http://arxiv.org/abs/1901.04407

[24] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, no. 1, pp. 187–210, May 2018.

[25] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, p. 1, Dec. 2014.

[26] M. Buehler, K. Iagnemma, and S. Singh, Eds., *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56. Berlin, Germany: Springer, 2009.

[27] A. Broggi, P. Cerri, M. Felisa, M. C. Laghi, L. Mazzei, and P. P. Porta, "The VisLab intercontinental autonomous challenge: An extensive test for a platoon of intelligent vehicles," *Int. J. Vehicle Auto. Syst.*, vol. 10, no. 3, p. 147, 2012.

[28] A. Broggi, P. Cerri, S. Debattisti, M. Chiara Laghi, P. Medici, D. Molinari, M. Panciroli, and A. Prioletti, "PROUD—Public road urban driverless-car test," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3508–3519, Dec. 2015.

[29] P. Cerri, G. Soprani, P. Zani, J. Choi, J. Lee, D. Kim, K. Yi, and A. Broggi, "Computer vision at the Hyundai autonomous challenge," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 777–783.

[30] C. Englund, L. Chen, J. Ploeg, E. Semsar-Kazerooni, A. Voronov, H. H. Bengtsson, and J. Didoff, "The grand cooperative driving challenge 2016: Boosting the introduction of cooperative automated vehicles," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 146–152, Aug. 2016.

[31] R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla, and A. Weller, "Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1–9.

[32] SAE, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," SAE Tech. Paper J3016_201806, 2016.

[33] DESA, UN, "World population prospects the 2017 revision key findings and advance tables," in *Proc. World Popul. Prospect.*, 2017, pp. 1–46.

[34] Deloitte. *2019 Deloitte Global Automotive Consumer Study—Advanced Vehicle Technologies and Multimodal Transportation, Global Focus Countries*. Accessed: May 19, 2019. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/manufacturing/us-global-automotive-consumer-study-2019.pdf

[35] Federatione Internationale de l'Automobile (FiA) Region 1. *The Automotive Digital Transformation and the Economic Impacts of Existing Data Access Models*. Accessed: May 19, 2019. [Online]. Available: https://www.fiaregion1.com/wp-content/uploads/2019/03/The-Automotive-Digital-Transformation_Full-study.pdf

[36] R. Rajamani, *Vehicle Dynamics and Control*, vol. 25. Berlin, Germany: Springer, 2006, p. 471.

[37] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1162–1175, Sep. 2013.

[38] A. Colombo and D. Del Vecchio, "Efficient algorithms for collision avoidance at intersections," in *Proc. 15th ACM Int. Conf. Hybrid Syst., Comput. Control (HSCC)*, 2012, pp. 145–154.

[39] P. E. Ross, "The audi a8: The World's first production car to achieve level 3 autonomy," *IEEE Spectr.*, vol. 1, 2017. [Online]. Available: https://spectrum.ieee.org/cars-that-think/transportation/self-driving/the-audi-a8-the-worlds-first-production-car-to-achieve-level-3-autonomy

[40] C. Gold, M. Körber, D. Lechner, and K. Bengler, "Taking over control from highly automated vehicles in complex traffic situations: The role of traffic density," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 58, no. 4, pp. 642–652, Jun. 2016.

[41] N. Merat, A. H. Jamson, F. C. H. Lai, M. Daly, and O. M. J. Carsten, "Transition to manual: Driver behaviour when resuming control from a highly automated vehicle," *Transp. Res. F, Traffic Psychol. Behaviour*, vol. 27, pp. 274–282, Nov. 2014.

[42] E. Ackerman, "Toyota's gill pratt on self-driving cars and the reality of full autonomy," *IEEE Spectr.*, May 2017. [Online]. Available: https://spectrum.ieee.org/cars-that-think/transportation/self-driving/toyota-gill-pratt-on-the-reality-of-full-autonomy

[43] J. D'Onfro. *'I Hate Them': Locals Reportedly Are Frustrated With Alphabet's Self-Driving Cars*. Accessed: May 19, 2019. [Online]. Available: https://www.cnbc.com/2018/08/28/locals-reportedly-frustrated-with-alphabets-waymo-self-driving-cars.html

[44] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, Jun. 2016.

[45] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," 2015, *arXiv:1510.03346*. [Online]. Available: http://arxiv.org/abs/1510.03346

[46] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: Automated testing of deep-neural-network-driven autonomous cars," in *Proc. 40th Int. Conf. Softw. Eng. (ICSE)*, 2018, pp. 303–314.

[47] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[48] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 241–246.

[49] E.-K. Lee, M. Gerla, G. Pau, U. Lee, and J.-H. Lim, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular fogs," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 9, Sep. 2016, Art. no. 155014771666550.

[50] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: A survey and future perspectives," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 98–104, Feb. 2016.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 24

[51] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 763–770.
[52] A. Broggi, M. Buzzoni, S. Debattisti, P. Grisleri, M. C. Laghi, P. Medici, and P. Versari, "Extensive tests of autonomous driving technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1403–1415, Sep. 2013.
[53] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, Jan. 2017.
[54] N. Akai, L. Y. Morales, T. Yamaguchi, E. Takeuchi, Y. Yoshihara, H. Okuda, T. Suzuki, and Y. Ninomiya, "Autonomous driving based on accurate localization using multilayer LiDAR and dead reckoning," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.
[55] E. Guizzo, "How Google's self-driving car works," *IEEE Spectr.*, vol. 18, no. 7, pp. 1132–1141, Oct. 2011.
[56] H. Somerville, P. Lienert, and A. Sage. (Mar. 2018). *Uber's Use of Fewer Safety Sensors Prompts Questions After Arizona Crash. Business News, Reuters*. Accessed: Dec. 16, 2018. [Online]. Available: https://www.reuters.com/article/us-uber-selfdriving-sensors-insight/ubers-use-of-fewer-safety-sensors-prompts-questions-after-arizona-crash-idUSKBN1H337Q
[57] J. Ziegler *et al.*, "Making Bertha drive—An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Summer 2014.
[58] Baidu. *Apollo Auto*. Accessed: May 1, 2019. [Online]. Available: https://github.com/ApolloAuto/apollo
[59] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2722–2730.
[60] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1989, pp. 305–313.
[61] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun, "Off-road obstacle avoidance through end-to-end learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 739–746.
[62] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*. [Online]. Available: http://arxiv.org/abs/1604.07316
[63] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," 2017, *arXiv:1612.01079*. [Online]. Available: http://arxiv.org/abs/1612.01079
[64] A. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electron. Imag.*, vol. 2017, no. 19, pp. 70–76, Jan. 2017.
[65] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," 2018, *arXiv:1807.00412*. [Online]. Available: http://arxiv.org/abs/1807.00412
[66] S. Baluja, "Evolution of an artificial neural network based autonomous land vehicle controller," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 26, no. 3, pp. 450–463, Jun. 1996.
[67] J. Koutník, G. Cuccu, J. Schmidhuber, and F. Gomez, "Evolving large-scale neural networks for vision-based reinforcement learning," in *Proc. 15th Annu. Conf. Genetic Evol. Comput. Conf. (GECCO)*, 2013, pp. 1061–1068.
[68] S. Behere and M. Torngren, "A functional architecture for autonomous driving," in *Proc. 1st Int. Workshop Automot. Softw. Archit. (WASA)*, May 2015, pp. 3–10.
[69] L. Chi and Y. Mu, "Deep steering: Learning end-to-end driving model from spatial and temporal visual cues," 2017, *arXiv:1708.03798*. [Online]. Available: http://arxiv.org/abs/1708.03798
[70] J.-P. Laumond, *Robot Motion Planning and Control*, vol. 229. Berlin, Germany: Springer, 1998.
[71] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, vol. 5. New York, NY, USA: McGraw-Hill, 1995.
[72] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, "Constraint-based planning and control for safe, semi-autonomous operation of vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 383–388.
[73] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[74] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: From architectures to learning," *Evol. Intell.*, vol. 1, no. 1, pp. 47–62, Mar. 2008.
[75] H. T. Cheng, H. Shan, and W. Zhuang, "Infotainment and road safety service support in vehicular networking: From a communication perspective," *Mech. Syst. Signal Process.*, vol. 25, no. 6, pp. 2020–2038, Aug. 2011.
[76] J. Wang, Y. Shao, Y. Ge, and R. Yu, "A survey of vehicle to everything (V2X) testing," *Sensors*, vol. 19, no. 2, p. 334, 2019.
[77] C. H. Jang, C. S. Kim, K. C. Jo, and M. Sunwoo, "Design factor optimization of 3D flash lidar sensor based on geometrical model for automated vehicle and advanced driver assistance system applications," *Int. J. Automot. Technol.*, vol. 18, no. 1, pp. 147–156, Feb. 2017.
[78] A. I. Maqueda, A. Loquercio, G. Gallego, N. Garcia, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5419–5427.
[79] C. Fries and H.-J. Wuensche, "Autonomous convoy driving by night: The vehicle tracking system," in *Proc. IEEE Int. Conf. Technol. Practical Robot Appl. (TePRA)*, May 2015, pp. 1–6.
[80] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada, "MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5108–5115.
[81] T. B. Lee. *How 10 Leading Companies Are Trying To Make Powerful, Low-Cost Lidar*. Accessed: May 19, 2019. [Online]. Available: https://arstechnica.com/cars/2019/02/the-ars-technica-guide-to-the-lidar-industry/
[82] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "CarSpeak: A content-centric network for autonomous driving," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2012, pp. 259–270.
[83] E. Yurtsever, S. Yamazaki, C. Miyajima, K. Takeda, M. Mori, K. Hitomi, and M. Egawa, "Integrating driving behavior and traffic context through signal symbolization for data reduction and risky lane change detection," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 3, pp. 242–253, Sep. 2018.
[84] M. Gerla, "Vehicular cloud computing," in *Proc. 11th Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, 2012, pp. 152–155.
[85] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Netw. Comput. Appl.*, vol. 40, pp. 325–344, Apr. 2014.
[86] I. Din, B.-S. Kim, S. Hassan, M. Guizani, M. Atiquzzaman, and J. Rodrigues, "Information-centric network-based vehicular communications: Overview and research opportunities," *Sensors*, vol. 18, no. 11, p. 3957, 2018.
[87] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1161–1168.
[88] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," Apr. 2017, *arXiv:1704.05519*. [Online]. Available: https://arxiv.org/abs/1704.05519
[89] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD17: End-to-end DAVIS driving dataset," 2017, *arXiv:1711.01458*. [Online]. Available: http://arxiv.org/abs/1711.01458
[90] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.
[91] B. Schoettle, "Sensor fusion: A comparison of sensing capabilities of human drivers and highly automated vehicles," Sustain. Worldwide Transp., Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep. SWT-2017-12, Aug. 2017.
[92] Tesla Motors. *Autopilot Press Kit*. Accessed: Dec. 16, 2018. [Online]. Available: https://www.tesla.com/presskit/autopilot#autopilot
[93] SXSW Interactive. (2016). *Chris Urmson Explains Google Self-Driving Car Project*. Accessed: Dec. 16, 2018. [Online]. Available: https://www.sxsw.com/interactive/2016/chris-urmson-explain-googles-self-driving-car-project/
[94] M. A. Al-Khedher, "Hybrid GPS-GSM localization of automobile tracking system," 2012, *arXiv:1201.2630*. [Online]. Available: http://arxiv.org/abs/1201.2630
[95] K. S. Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4, Apr. 1997, pp. 2783–2788.
[96] C. Urmson, J. Anhalt, M. Clark, T. Galatali, J. P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, P. L. Koon, K. Peterson, B. K. Smith, S. Spiker, E. Tryzelaar, and W. L. Whittaker, "High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004," Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-04-37, 2004.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 25

[97] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[98] A. Hata and D. Wolf, "Road marking detection using LIDAR reflective intensity data and its application to vehicle localization," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 584–589.

[99] T. Ort, L. Paull, and D. Rus, "Autonomous vehicle navigation in rural environments without detailed prior maps," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2040–2047.

[100] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4372–4378.

[101] E. Takeuchi and T. Tsubouchi, "A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3068–3073.

[102] S. E. Shladover, "PATH at 20—History and major milestones," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 584–592, Dec. 2007.

[103] A. Alam, B. Besselink, V. Turri, J. MåRtensson, and K. H. Johansson, "Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency," *IEEE Control Syst. Mag.*, vol. 35, no. 6, pp. 34–56, Dec. 2015.

[104] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, S. Shladover, and S. Tsugawa, "Overview of platooning systems," in *Proc. 19th ITS World Congr.*, Vienna, Austria, Oct. 2012, pp. 1–8.

[105] E. Chan, "Sartre automated platooning vehicles," *Towards Innov. Freight Logistics*, vol. 2, pp. 137–150, May 2016.

[106] A. Keymasi Khalaji and S. A. A. Moosavian, "Robust adaptive controller for a tractor-trailer mobile robot," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 943–953, Jun. 2014.

[107] J. Cheng, B. Wang, and Y. Xu, "Backward path tracking control for mobile robot with three trailers," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, Nov. 2017, pp. 32–41.

[108] M. Hejase, J. Jing, J. M. Maroli, Y. Bin Salamah, L. Fiorentini, and U. Ozguner, "Constrained backward path tracking control using a plug-in jackknife prevention system for autonomous tractor-trailers," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2012–2017.

[109] F. Zhang, H. Stahle, G. Chen, C. C. C. Simon, C. Buckl, and A. Knoll, "A sensor fusion approach for localization with cumulative error elimination," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. for Intell. Syst. (MFI)*, Sep. 2012, pp. 1–6.

[110] W.-W. Kao, "Integration of GPS and dead-reckoning navigation systems," in *Proc. Vehicle Navigat. Inf. Syst. Conf.*, vol. 2, Oct. 1991, pp. 635–643.

[111] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments," in *Robotics: Science and Systems III*, W. Burgard, O. Brock, and C. Stachniss, Eds. Cambridge, MA, USA: MIT Press, 2007, ch. 16, pp. 4372–4378.

[112] A. Ranganathan, D. Ilstrup, and T. Wu, "Light-weight localization for vehicles using road markings," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 921–927.

[113] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax,and M. Walter, "A perception-driven autonomous urban vehicle," *J. Field Robot.*, vol. 25, no. 10, pp. 727–774, 2008.

[114] N. Akai, L. Y. Morales, E. Takeuchi, Y. Yoshihara, and Y. Ninomiya, "Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1356–1363.

[115] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor fusion-based low-cost vehicle localization system for complex urban environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1078–1086, May 2017.

[116] D. Gruyer, R. Belaroussi, and M. Revilloud, "Accurate lateral positioning from map data and road marking detection," *Expert Syst. Appl.*, vol. 43, pp. 1–8, Jan. 2016.

[117] X. Qu, B. Soheilian, and N. Paparoditis, "Vehicle localization using mono-camera and geo-referenced traffic signs," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 605–610.

[118] M. Magnusson, "The three-dimensional normal-distributions transform—An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Graduate School Robot. Automat. Process Control, Örebro Univ., Örebro, Sweden, 2009.

[119] R. W. Wolcott and R. M. Eustice, "Fast LIDAR localization using multiresolution Gaussian mixture maps," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2814–2821.

[120] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3D registration reliability and speed—A comparison of ICP and NDT," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3907–3912.

[121] R. Valencia, J. Saarinen, H. Andreasson, J. Vallvé, J. Andrade-Cetto, and A. J. Lilienthal, "Localization in highly dynamic environments using dual-timescale NDT-MCL," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2014, pp. 3956–3962.

[122] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, Dec. 2015.

[123] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 176–183.

[124] C. Mcmanus, W. Churchill, A. Napier, B. Davis, and P. Newman, "Distraction suppression for vision-based pose estimation at city scales," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3762–3769.

[125] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," Feb. 2016, *arXiv:1602.07261*. [Online]. Available: https://arxiv.org/abs/1602.07261

[126] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[127] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," Aug. 2016, *arXiv:1608.06993*. [Online]. Available: https://arxiv.org/abs/1608.06993

[128] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[129] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[130] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Apr. 2015, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[131] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[132] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward," *Comput. Vis. Image Understand.*, vol. 117, no. 8, pp. 827–891, Aug. 2013.

[133] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[134] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," Sep. 2018, *arXiv:1809.02165*. [Online]. Available: https://arxiv.org/abs/1809.02165

[135] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[136] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[137] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," Dec. 2015, *arXiv:1512.02325*. [Online]. Available: https://arxiv.org/abs/1512.02325

[138] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[139] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," Feb. 2018, *arXiv:1802.02611*. [Online]. Available: https://arxiv.org/abs/1802.02611

[140] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.

[141] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2000, pp. 445–461.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 26

[142] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 5695–5701.
[143] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1015–1026, Oct. 2008.
[144] M. Schonbein and A. Geiger, "Omnidirectional 3D reconstruction in augmented manhattan worlds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 716–723.
[145] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," Apr. 2019, *arXiv:1904.08405*. [Online]. Available: https://arxiv.org/abs/1904.08405
[146] R. H. Rasshofer and K. Gresser, "Automotive radar and lidar systems for next generation driver assistance functions," *Adv. Radio Sci.*, vol. 3, pp. 205–209, May 2005.
[147] P. Radecki, M. Campbell, and K. Matzen, "All weather perception: Joint data association, tracking, and classification for autonomous ground vehicles," May 2016, *arXiv:1605.02196*. [Online]. Available: https://arxiv.org/abs/1605.02196
[148] P. Hurney, F. Morgan, M. Glavin, E. Jones, and P. Waldron, "Review of pedestrian detection techniques in automotive far-infrared video," *IET Intell. Transp. Syst.*, vol. 9, no. 8, pp. 824–832, Oct. 2015.
[149] N. Carlevaris-Bianco and R. M. Eustice, "Learning visual feature descriptors for dynamic lighting conditions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 2769–2776.
[150] V. Peretroukhin, W. Vega-Brown, N. Roy, and J. Kelly, "PROBE-GK: Predictive robust estimation using generalized kernels," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 817–824.
[151] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Proc. Vis. Place Recognit. Changing Environ. Workshop, IEEE Int. Conf. Robot. Automat. (ICRA)*, Hong Kong, vol. 2, May 2014, p. 3.
[152] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," May 2014, *arXiv:1405.0312*. [Online]. Available: https://arxiv.org/abs/1405.0312
[153] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," Jun. 2015, *arXiv:1506.01497*. [Online]. Available: https://arxiv.org/abs/1506.01497
[154] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Washington, DC, USA: IEEE Computer Society, Dec. 2015, pp. 1520–1528, doi: 10.1109/ICCV.2015.178.
[155] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," May 2015, *arXiv:1505.04597*. [Online]. Available: https://arxiv.org/abs/1505.04597
[156] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," Dec. 2016, *arXiv:1612.01105*. [Online]. Available: https://arxiv.org/abs/1612.01105
[157] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," Jun. 2016, *arXiv:1606.00915*. [Online]. Available: https://arxiv.org/abs/1606.00915
[158] X. Ma, Z. Wang, H. Li, W. Ouyang, X. Fan, and P. Zhang, "Accurate monocular object detection via color-embedded 3D reconstruction for autonomous driving," Mar. 2019, *arXiv:1903.11444*. [Online]. Available: https://arxiv.org/abs/1903.11444
[159] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," 2018, *arXiv:1810.02695*. [Online]. Available: https://arxiv.org/abs/1810.02695
[160] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," *KI–Künstliche Intelligenz*, vol. 24, pp. 345–348, Oct. 2009.
[161] W. Wang, K. Sakurada, and N. Kawaguchi, "Incremental and enhanced scanline-based segmentation method for surface reconstruction of sparse LiDAR data," *Remote Sens.*, vol. 8, no. 11, p. 967, 2016.
[162] P. Narksri, E. Takeuchi, Y. Ninomiya, Y. Morales, N. Akai, and N. Kawaguchi, "A slope-robust cascaded ground segmentation in 3D point cloud for autonomous vehicles," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 497–504.
[163] J. Lambert, L. Liang, Y. Morales, N. Akai, A. Carballo, E. Takeuchi, P. Narksri, S. Seiya, and K. Takeda, "Tsukuba challenge 2017 dynamic object tracks dataset for pedestrian behavior analysis," *J. Robot. Mechtron.*, vol. 30, no. 4, pp. 598–612, Aug. 2018.
[164] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," 2017, *arXiv:1711.03938*. [Online]. Available: http://arxiv.org/abs/1711.03938
[165] S. Song and J. Xiao, "Sliding shapes for 3D object detection in depth images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2014, pp. 634–651.
[166] D. Z. Wang and I. Posner, "Voting for voting in online point cloud object detection," in *Proc. Robot., Sci. Syst.*, Jul. 2015, pp. 1–9.
[167] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," Nov. 2017, *arXiv:1711.06396*. [Online]. Available: https://arxiv.org/abs/1711.06396
[168] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. New York, NY, USA: Curran Associates, Inc., 2015, pp. 424–432.
[169] D. Lin, S. Fidler, and R. Urtasun, "Holistic scene understanding for 3D object detection with RGBD cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1417–1424.
[170] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," in *Proc. Robot., Sci. Syst.*, Jun. 2016, pp. 1–8.
[171] L. Liu, Z. Pan, and B. Lei, "Learning a rotation invariant detector with rotatable bounding box," Nov. 2017, *arXiv:1711.09405*. [Online]. Available: https://arxiv.org/abs/1711.09405
[172] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6526–6534.
[173] M. Ren, A. Pokrovsky, B. Yang, and R. Urtasun, "SBNet: Sparse blocks network for fast inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8711–8720.
[174] W. Ali, S. Abdelkarim, M. Zahran, M. Zidan, and A. El Sallab, "YOLO3D: End-to-end real-time 3D oriented object bounding box detection from LiDAR point cloud," Aug. 2018, *arXiv:1808.02350*. [Online]. Available: https://arxiv.org/abs/1808.02350
[175] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7652–7660.
[176] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3D vehicle detection," Apr. 2018, *arXiv:1804.05132*. [Online]. Available: https://arxiv.org/abs/1804.05132
[177] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[178] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. Erin Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," 2019, *arXiv:1903.11027*. [Online]. Available: http://arxiv.org/abs/1903.11027
[179] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," Dec. 2018, *arXiv:1812.04244*. [Online]. Available: https://arxiv.org/abs/1812.04244
[180] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," Dec. 2018, *arXiv:1812.05784*. [Online]. Available: https://arxiv.org/abs/1812.05784
[181] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "IPOD: Intensive point-based object detector for point cloud," Dec. 2018, *arXiv:1812.05276*. [Online]. Available: https://arxiv.org/abs/1812.05276
[182] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," Nov. 2017, *arXiv:1711.08488*. [Online]. Available: https://arxiv.org/abs/1711.08488
[183] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple object tracking: A literature review," Sep. 2014, *arXiv:1409.7618*. [Online]. Available: https://arxiv.org/abs/1409.7618
[184] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 802–807.
[185] J. Shi and Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 27

[186] M.-P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *Proc. 12th Int. Conf. Pattern Recognit.*, vol. 1, Oct. 1994, pp. 566–568.

[187] S. Hwang, N. Kim, Y. Choi, S. Lee, and I. S. Kweon, "Fast multiple objects detection and tracking fusing color camera and 3D LIDAR for intelligent vehicles," in *Proc. 13th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Aug. 2016, pp. 234–239.

[188] T.-N. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow, and M.-M. Meinecke, "Stereo-camera-based urban environment perception using occupancy grid and object tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 154–165, Mar. 2012.

[189] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha—A local, continuous method," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Jun. 2014, pp. 450–457.

[190] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *Int. J. Robot. Res.*, vol. 29, no. 14, pp. 1707–1725, Dec. 2010.

[191] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Auto. Robots*, vol. 26, nos. 2–3, pp. 123–139, Apr. 2009.

[192] M. He, E. Takeuchi, Y. Ninomiya, and S. Kato, "Precise and efficient model-based vehicle tracking method using Rao-Blackwellized and scaling series particle filters," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 117–124.

[193] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2D lidar," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 1039–1063, Jun. 2015.

[194] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng, "An empirical evaluation of deep learning on highway driving," Apr. 2015, *arXiv:1504.01716*. [Online]. Available: https://arxiv.org/abs/1504.01716

[195] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," Apr. 2016, *arXiv:1604.01802*. [Online]. Available: https://arxiv.org/abs/1604.01802

[196] S. Chowdhuri, T. Pankaj, and K. Zipser, "MultiNet: Multi-modal multi-task learning for autonomous driving," Sep. 2017, *arXiv:1709.05581*. [Online]. Available: https://arxiv.org/abs/1709.05581

[197] *Autoware*. Accessed: Jun. 12, 2019. [Online]. Available: https://github.com/autowarefoundation/autoware

[198] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[199] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 727–745, Apr. 2014.

[200] C. Fernández, D. Fernández-Llorca, and M. A. Sotelo, "A hybrid vision-map method for urban road detection," *J. Adv. Transp.*, vol. 2017, Oct. 2017, Art. no. 7090549.

[201] R. Labayrade, "A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance," *IEICE Trans. Inf. Syst.*, vols. E89–D, no. 7, pp. 2092–2100, Jul. 2006.

[202] Y. Jiang, F. Gao, and G. Xu, "Computer vision-based multiple-lane detection on straight road and in a curve," in *Proc. Int. Conf. Image Anal. Signal Process.*, Apr. 2010, pp. 114–117.

[203] M. Paton, K. MacTavish, C. J. Ostafew, and T. D. Barfoot, "It's not easy seeing green: Lighting-resistant stereo visual teach & repeat using color-constant images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1519–1526.

[204] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Finding multiple lanes in urban road networks with vision and lidar," *Auto. Robots*, vol. 26, nos. 2–3, pp. 103–122, Apr. 2009.

[205] H.-Y. Cheng, B.-S. Jeng, P.-T. Tseng, and K.-C. Fan, "Lane detection with moving vehicles in the traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 571–582, Dec. 2006.

[206] J. M. Álvarez, A. M. López, and R. Baldrich, "Shadow resistant road segmentation from a mobile monocular system," in *Pattern Recognition and Image Analysis*. Berlin, Germany: Springer, 2007, pp. 9–16.

[207] R. Danescu and S. Nedevschi, "Probabilistic lane tracking in difficult road scenarios using stereovision," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 272–282, Jun. 2009.

[208] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[209] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with Ransac and Kalman filter," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 3261–3264.

[210] A. V. Nefian and G. R. Bradski, "Detection of drivable corridors for off-road autonomous navigation," in *Proc. Int. Conf. Image Process.*, Oct. 2006, pp. 3025–3028.

[211] E. Yurtsever, Y. Liu, J. Lambert, C. Miyajima, E. Takeuchi, K. Takeda, and J. H. L. Hansen, "Risky action recognition in lane change video clips using deep spatiotemporal networks with segmentation mask transfer," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3100–3107.

[212] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2016.

[213] S. Yamazaki, C. Miyajima, E. Yurtsever, K. Takeda, M. Mori, K. Hitomi, and M. Egawa, "Integrating driving behavior and traffic context through signal symbolization," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 642–647.

[214] X. Geng, H. Liang, B. Yu, P. Zhao, L. He, and R. Huang, "A scenario-adaptive driving behavior prediction approach to urban autonomous driving," *Appl. Sci.*, vol. 7, no. 4, p. 426, 2017.

[215] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr, "A combined model-and learning-based framework for interaction-aware maneuver prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1538–1550, Jun. 2016.

[216] V. Gadepally, A. Krishnamurthy, and Ü. Özgüner, "A framework for estimating long term driver behavior," *J. Adv. Transp.*, vol. 2017, pp. 1–11, Jan. 2017.

[217] P. Liu, A. Kurt, K. Redmill, and U. Ozguner, "Classification of highway lane change behavior to detect dangerous cut-in maneuvers," in *Proc. 95th Annu. Meeting Transp. Res. Board (TRB)*, vol. 2, 2015, pp. 1–14.

[218] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 797–802.

[219] H. Darweesh, E. Takeuchi, K. Takeda, Y. Ninomiya, A. Sujiwo, L. Y. Morales, N. Akai, T. Tomizawa, and S. Kato, "Open source integrated planner for autonomous navigation in highly dynamic environments," *J. Robot. Mechatronics*, vol. 29, no. 4, pp. 668–684, 2017.

[220] F. Sagberg, Selpi, G. F. Bianchi Piccinini, and J. Engström, "A review of research on driving styles and road safety," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 57, no. 7, pp. 1248–1275, Nov. 2015.

[221] C. Marina Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 666–676, Mar. 2018.

[222] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1609–1615.

[223] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. C. González, "Safe driving using mobile phones," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1462–1468, Sep. 2012.

[224] N. Karginova, S. Byttner, and M. Svensson, "Data-driven methods for classification of driving styles in buses," SAE Tech. Paper 2012-01-0744, 2012.

[225] A. Doshi and M. M. Trivedi, "Examining the impact of driving style on the predictability and responsiveness of the driver: Real-world and simulator analysis," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 232–237.

[226] V. Vaitkus, P. Lengvenis, and G. Zylius, "Driving style classification using long-term accelerometer information," in *Proc. 19th Int. Conf. Methods Models Autom. Robot. (MMAR)*, Sep. 2014, pp. 641–644.

[227] F. Syed, S. Nallapa, A. Dobryden, C. Grand, R. McGee, and D. Filev, "Design and analysis of an adaptive real-time advisory system for improving real world fuel economy in a hybrid electric vehicle," SAE Tech. Paper 2010-01-0835, 2010.

[228] A. Corti, C. Ongini, M. Tanelli, and S. M. Savaresi, "Quantitative driving style estimation for energy-oriented applications in road vehicles," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 3710–3715.

[229] E. Ericsson, "Independent driving pattern factors and their influence on fuel-use and exhaust emission factors," *Transp. Res. D, Transp. Environ.*, vol. 6, no. 5, pp. 325–345, Sep. 2001.

[230] V. Manzoni, A. Corti, P. De Luca, and S. M. Savaresi, "Driving style estimation via inertial measurements," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 777–782.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 28

[231] J. S. Neubauer and E. Wood, "Accounting for the variation of driver aggression in the simulation of conventional and advanced vehicles," SAE Tech. Paper 2013-01-1453, 2013.
[232] Y. L. Murphey, R. Milton, and L. Kiliaris, "Driver's style classification using jerk analysis," in *Proc. IEEE Workshop Comput. Intell. Vehicles Veh. Syst.*, Mar. 2009, pp. 23–28.
[233] E. Yurtsever, K. Takeda, and C. Miyajima, "Traffic trajectory history and drive path generation using GPS data cloud," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 229–234.
[234] D. Dorr, D. Grabengiesser, and F. Gauterin, "Online driving style recognition using fuzzy logic," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1021–1026.
[235] L. Xu, J. Hu, H. Jiang, and W. Meng, "Establishing style-oriented driver models by imitating human driving behaviors," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2522–2530, Oct. 2015.
[236] B. Rajan, A. McGordon, and P. Jennings, "An investigation on the effect of driver style and driving events on energy demand of a PHEV," *World Electric Vehicle J.*, vol. 5, no. 1, pp. 173–181, 2012.
[237] A. Augustynowicz, "Preliminary classification of driving style with objective rank method," *Int. J. Automot. Technol.*, vol. 10, no. 5, pp. 607–610, Oct. 2009.
[238] Z. Constantinescu, C. Marinoiu, and M. Vladoiu, "Driving style analysis using data mining techniques," *Int. J. Comput. Commun. Control*, vol. 5, no. 5, p. 654, 2010.
[239] Y. Zhang, W. C. Lin, and Y.-K.-S. Chin, "A pattern-recognition approach for driving skill characterization," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 905–916, Dec. 2010.
[240] E. Yurtsever, C. Miyajima, and K. Takeda, "A traffic flow simulation framework for learning driver heterogeneity from naturalistic driving data using autoencoders," *Int. J. Automot. Eng.*, vol. 10, no. 1, pp. 86–93, 2019.
[241] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda, and F. Itakura, "Driver modeling based on driving behavior and its evaluation in driver identification," *Proc. IEEE*, vol. 95, no. 2, pp. 427–437, Feb. 2007.
[242] A. Bolovinou, I. Bakas, A. Amditis, F. Mastrandrea, and W. Vinciotti, "Online prediction of an electric vehicle remaining range based on regression analysis," in *Proc. IEEE Int. Electr. Vehicle Conf. (IEVC)*, Dec. 2014, pp. 1–8.
[243] A. Mudgal, S. Hallmark, A. Carriquiry, and K. Gkritza, "Driving behavior at a roundabout: A hierarchical Bayesian regression analysis," *Transp. Res. Part D: Transp. Environ.*, vol. 26, pp. 20–26, Jan. 2014.
[244] J. C. McCall and M. M. Trivedi, "Driver behavior and situation aware brake assistance for intelligent vehicles," *Proc. IEEE*, vol. 95, no. 2, pp. 374–387, Feb. 2007.
[245] E. Yurtsever, C. Miyajima, S. Selpi, and K. Takeda, "Driving signature extraction," in *Proc. 3rd Int. Symp. Future Act. Saf. Technol. Toward Zero Traffic Accidents (FAST-Zero)*, 2015, pp. 1–5.
[246] K. Sama, Y. Morales, N. Akai, H. Liu, E. Takeuchi, and K. Takeda, "Driving feature extraction and behavior classification using an autoencoder to reproduce the velocity styles of experts," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1337–1343.
[247] H. Liu, T. Taniguchi, Y. Tanaka, K. Takenaka, and T. Bando, "Visualization of driving behavior based on hidden feature extraction by using deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2477–2489, Sep. 2017.
[248] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," in *Algorithm Engineering*. Cham, Switzerland: Springer, 2016, pp. 19–80.
[249] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.
[250] D. Van Vliet, "Improved shortest path algorithms for transport networks," *Transp. Res.*, vol. 12, no. 1, pp. 7–20, Feb. 1978.
[251] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter, "Exact routing in large road networks using contraction hierarchies," *Transp. Sci.*, vol. 46, no. 3, pp. 388–404, Aug. 2012.
[252] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick, "Reachability and distance queries via 2-Hop labels," *SIAM J. Comput.*, vol. 32, no. 5, pp. 1338–1355, Jan. 2003.
[253] R. Bauer, D. Delling, P. Sanders, D. Schieferdecker, D. Schultes, and D. Wagner, "Combining hierarchical and goal-directed speed-up techniques for Dijkstra's algorithm," *J. Experim. Algorithmics*, vol. 15, pp. 2–3, Mar. 2010.
[254] D. Delling, A. V. Goldberg, A. Nowatzyk, and R. F. Werneck, "PHAST: Hardware-accelerated shortest path trees," *J. Parallel Distrib. Comput.*, vol. 73, no. 7, pp. 940–952, Jul. 2013.
[255] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *Proc. Int. Symp. Artif. Intell., Robot., Automat. Space*, 2005, pp. 1–7.
[256] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *Int. J. Robot. Res.*, vol. 10, no. 6, pp. 628–649, Dec. 1991.
[257] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.
[258] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.
[259] L. E. Kavraki *et al.*, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, 1996.
[260] H. Fuji, J. Xiang, Y. Tazaki, B. Levedahl, and T. Suzuki, "Trajectory planning for automated parking using multi-resolution state roadmap considering non-holonomic constraints," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Jun. 2014, pp. 407–413.
[261] P. Petrov and F. Nashashibi, "Modeling and nonlinear adaptive control for autonomous vehicle overtaking," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1643–1656, Aug. 2014.
[262] J. P. Rastelli, R. Lattarulo, and F. Nashashibi, "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Jun. 2014, pp. 510–515.
[263] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, Apr. 2010.
[264] J. Ren, K. A. McIsaac, and R. V. Patel, "Modified Newton's method applied to potential field-based navigation for mobile robots," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 384–391, Apr. 2006.
[265] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LIDAR-based driving path generation using fully convolutional neural networks," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.
[266] D. Barnes, W. Maddern, and I. Posner, "Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 203–210.
[267] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
[268] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2034–2039.
[269] C. A. Pickering, K. J. Burnham, and M. J. Richardson, "A review of automotive human machine interface technologies and techniques to reduce driver distraction," in *Proc. 2nd IET Int. Conf. Syst. Saf.*, 2007, pp. 223–228.
[270] O. Carsten and M. H. Martens, "How can humans understand their automated cars? HMI principles, problems and solutions," *Cognition, Technol. Work*, vol. 21, no. 1, pp. 3–20, Feb. 2019.
[271] P. Bazilinskyy and J. de Winter, "Auditory interfaces in automated driving: An international survey," *PeerJ Comput. Sci.*, vol. 1, p. e13, Aug. 2015.
[272] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. A. Hyder, E. Jarawan, and C. D. Mathers, *World Report on Road Traffic Injury Prevention*. Geneva, Switzerland: World Health Organization, 2004.
[273] D. R. Large, L. Clark, A. Quandt, G. Burnett, and L. Skrypchuk, "Steering the conversation: A linguistic exploration of natural language interactions with a digital assistant during simulated driving," *Appl. Ergonom.*, vol. 63, pp. 53–61, Sep. 2017.
[274] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[275] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving video database with scalable annotation tooling," 2018, *arXiv:1805.04687*. [Online]. Available: http://arxiv.org/abs/1805.04687
[276] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5000–5009.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 29

[277] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes," 2019, *arXiv:1903.01568*. [Online]. Available: http://arxiv.org/abs/1903.01568
[278] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," 2018, *arXiv:1803.06184*. [Online]. Available: http://arxiv.org/abs/1803.06184
[279] Udacity. *Udacity Dataset*. Accessed: Apr. 30, 2019. [Online]. Available: https://github.com/udacity/self-driving-car/tree/master/datasets
[280] H. Schafer, E. Santana, A. Haden, and R. Biasini. "A commute in data: The Comma2k19 dataset," 2018, *arXiv:1812.05752*. [Online]. Available: https://arxiv.org/abs/1812.05752
[281] Y. Chen, J. Wang, J. Li, C. Lu, Z. Luo, H. Xue, and C. Wang, "LiDAR-video driving dataset: Learning driving policies effectively," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5870–5878.
[282] K. Takeda, J. H. L. Hansen, P. Boyraz, L. Malta, C. Miyajima, and H. Abut, "International large-scale vehicle corpora for research on driver behavior on the road," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1609–1623, Dec. 2011.
[283] A. Blatt, J. Pierowicz, M. Flanigan, P. S. Lin, A. Kourtellis, C. Lee, P. Jovanis, J. Jenness, M. Wilaby, and J. Campbell, "Naturalistic driving study: Field data collection," SHRP 2 Rep. S2-S07-RW-1, 2015.
[284] S. G. Klauer, F. Guo, J. Sudweeks, and T. A. Dingus, "An analysis of driver inattention using a case-crossover approach on 100-car data," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep., 2010.
[285] M. Benmimoun *et al.*, "euroFOT: Field operational test and impact assessment of advanced driver assistance systems: Final results," in *Proc. FISITA World Automot. Congr.* Berlin, Germany: Springer, 2013, pp. 537–547.
[286] S. Wang, M. Bai, G. Mattyus, H. Chu, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun, "TorontoCity: Seeing the world with a million eyes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3028–3036.
[287] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "KAIST multi-spectral Day/Night data set for autonomous and assisted driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 934–948, Mar. 2018.
[288] S. Sibi, H. Ayaz, D. P. Kuhns, D. M. Sirkin, and W. Ju, "Monitoring driver cognitive load using functional near infrared spectroscopy in partially autonomous cars," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 419–425.
[289] C. Braunagel, E. Kasneci, W. Stolzmann, and W. Rosenstiel, "Driver-activity recognition in the context of conditionally autonomous driving," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1652–1657.
[290] M. Walch, K. Lange, M. Baumann, and M. Weber, "Autonomous driving: Investigating the feasibility of car-driver handover assistance," in *Proc. 7th Int. Conf. Automot. User Interfaces Interact. Veh. Appl.*, 2015, pp. 11–18.
[291] J. H. L. Hansen, C. Busso, Y. Zheng, and A. Sathyanarayana, "Driver modeling for detection and assessment of driver distraction: Examples from the UTDrive test bed," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 130–142, Jul. 2017.
[292] M. Everingham *et al.*, "The 2005 PASCAL visual object classes challenge," in *Proc. Mach. Learn. Challenges Workshop*. Berlin, Germany: Springer, 2005.
[293] E. Santana and G. Hotz, "Learning a driving simulator," 2016, *arXiv:1608.01230*. [Online]. Available: http://arxiv.org/abs/1608.01230
[294] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 719–726.
[295] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu Apollo EM motion planner," 2018, *arXiv:1807.08048*. [Online]. Available: http://arxiv.org/abs/1807.08048
[296] NVIDIA. *Driveworks SDK*. Accessed: Dec. 9, 2018. [Online]. Available: https://developer.nvidia.com/driveworks
[297] CommaAI. *OpenPilot*. Accessed: Dec. 9, 2018. [Online]. Available: https://github.com/commaai/openpilot
[298] B. Wymann, C. Dimitrakakis, A. Sumner, E. Espié, and C. Guionneau. (2015). *TORCS: The Open Racing Car Simulator*. Accessed: May 2, 2019. [Online]. Available: http://www.cse.chalmers.se/~chrdimi/papers/torcs.pdf
[299] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, vol. 2, 2017, pp. 2213–2222.
[300] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 4, Sep./Oct. 2004, pp. 2149–2154.
[301] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (Simulation of Urban MObility)—An open-source traffic simulation," in *Proc. 4th Middle East Symp. Simulation Modelling (MESM)*, 2002, pp. 183–187.
[302] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zollner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1455–1462.

EKIM YURTSEVER (Member, IEEE) received the B.S. and M.S. degrees from Istanbul Technical University, in 2012 and 2014, respectively, and the Ph.D. degree in information science from Nagoya University, Japan, in 2019.

Since 2019, he has been a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Ohio State University. His research interests include artificial intelligence, machine learning, and computer vision. He is currently working on machine learning and computer vision tasks in the intelligent vehicle domain.

JACOB LAMBERT (Student Member, IEEE) received the B.S. degree (Hons.) in physics from McGill University, Montreal, Canada, in 2014, and the M.A.Sc. degree from the University of Toronto, Canada, in 2017. He is currently pursuing the Ph.D. degree with Nagoya University, Japan.

His current research focuses on 3D perception through lidar sensors for autonomous robotics.

ALEXANDER CARBALLO (Member, IEEE) received the Dr.Eng. degree from the Intelligent Robot Laboratory, University of Tsukuba, Japan.

From 1996 to 2006, he worked as a Lecturer at the School of Computer Engineering, Costa Rica Institute of Technology. From 2011 to 2017, he worked at the Research and Development, Hokuyo Automatic Co., Ltd. Since 2017, he has been a Designated Assistant Professor with the Institutes of Innovation for Future Society, Nagoya University, Japan. His main research interests are lidar sensors, robotic perception, and autonomous driving.

KAZUYA TAKEDA (Senior Member, IEEE) received the B.E.E., M.E.E., and Ph.D. degrees from Nagoya University, Japan.

Since 1985, he has been working with the Advanced Telecommunication Research Laboratories and KDD R&D Laboratories, Japan. In 1995, he started a research group for signal processing applications at Nagoya University. He is currently a Professor with the Institutes of Innovation for Future Society, Nagoya University, and with Tier IV Inc. He is also serving as a member of the Board of Governors of the IEEE ITS society. His main focus is investigating driving behavior using data centric approaches, utilizing signal corpora of real driving behavior.

Ex. 1043 – Safegate Supplemental Evidence
Safegate v. Honeywell – PGR2023-00053
Page 30