ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Z. Wang, A.D. Zhdanov,
N.B. Deryabin, D.D. Zhdanov

Methods and algorithms for organizing a vehicle autopilot system: a review

Recommended form of bibliographic references: Wang Z., Zhdanov A.D., Deryabin N.B., Zhdanov D.D. Methods and algorithms for organizing a vehicle autopilot system: a review // Keldysh Institute Preprints. 2025. No. 42. 29 p. EDN: ZRERMD
https://library.keldysh.ru/preprint.asp?id=2025-42&lg=e

О р д е н а   Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й   а к а д е м и и   н а у к

**Z. Wang, A.D. Zhdanov, N.B. Deryabin,
D.D. Zhdanov**

Methods and algorithms for organizing a vehicle autopilot system: a review

Москва — 2025

Ван Ч., Жданов А.Д., Дерябин Н.Б., Жданов Д.Д.

Методы и алгоритмы организации системы автопилота транспортных средств: обзор

Технология автономного вождения направлена на повышение безопасности и эффективности дорожного движения за счет снижения рисков, присущих транспортным средствам, управляемым человеком. Однако передовые системы (L3–L5) по-прежнему сталкиваются со значительными техническими проблемами. Текущие исследования в основном сосредоточены на трех основных модулях: восприятие (охватывающее слияние нескольких датчиков и обработку облаков точек на основе глубокого обучения), планирование (использование обучения с подкреплением и метаэвристических алгоритмов для оптимизации пути) и управление (использование Model Predictive Control (MPC) и нечетких моделей принятия решений). Несмотря на эти достижения, ограничения, такие как недостаточная интеграция подсистем, ограниченная адаптивность к динамическим средам и отсутствие этической структуры принятия решений, препятствуют дальнейшему прогрессу. В этой статье предлагается оптимизированный организационный алгоритм для улучшения взаимодействия в реальном времени между несколькими модулями. Он проверяет безопасность и эффективность с помощью имитационных тестов и исследует легкие модели и стандартизированные структуры тестирования. Двигаясь вперед, важно развивать многоагентную координацию, усиливать возможности обобщения в сложных сценариях и создавать надежную этическую структуру для содействия широкому внедрению технологии автономного вождения.

Ключевые слова: автономное вождение, восприятие, планирование, мультисенсорное слияние, сотрудничество подсистем.

Zhan Wang, Andrei Dmitrievich Zhdanov, Deryabin Nikolay Borisovich, Dmitry Dmitrievich Zhdanov

Methods and algorithms for organizing a vehicle autopilot system: a review

Autonomous driving technology aims to enhance road safety and efficiency by reducing the risks inherent in human-driven vehicles. However, advanced systems (L3-L5) still face substantial technical challenges. Current research predominantly focuses on three core modules: perception (encompassing multi-sensor fusion and deep learning-based point cloud processing), planning (employing reinforcement learning and meta-heuristic algorithms for path optimization), and control (utilizing Model Predictive Control (MPC) and fuzzy decision models). Despite these advancements, limitations such as insufficient subsystem integration, limited adaptability to dynamic environments, and the lack of an ethical decision-making framework hinder further progress. This paper proposes an optimized organizational algorithm to improve real-time collaboration among multiple modules. It verifies security and efficiency through simulation tests and investigates lightweight models and standardized testing frameworks. Moving forward, it is essential to advance multi-agent coordination, strengthen generalization capabilities in complex scenarios, and establish a robust ethical framework to promote the widespread adoption of autonomous driving technology.

Key words: autonomous driving, perception, perception, planning, multi-sensor fusion, subsystem collaboration.

Introduction

The three key components of transportation are currently vehicles, roads, and people. Among these, a vehicle is a complex machine with all its movements traceable, while the road serves as the environment in which both the vehicle and the person operate, acting as an entirely static element. Therefore, these two factors are not the primary causes of traffic accidents. In contrast, human behavior, characterized by subjective consciousness and unpredictability, tends to fluctuate under the influence of emotions such as fatigue, tension, and anger. According to a survey previously released by the World Health Organization (WHO), 95% of traffic accidents are attributed to drivers. Consequently, removing the driver as an unstable factor from the transportation system not only alleviates the burden on drivers but also contributes to reducing traffic accidents and enhancing road safety. Automated driving systems can be classified into five levels, as illustrated in Fig. 1, with the driver progressively being replaced by the automated system from level one to level five. Currently, the highest level of autonomous driving systems in mass production is level 2, while levels 3, 4, and 5 remain in limited experimental stages. Thus, advanced autonomous driving systems remain the focal point of ongoing research.

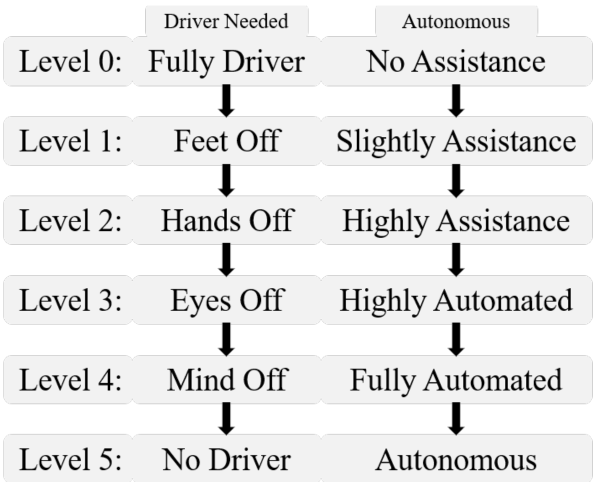

Fig.1 Criteria for Evaluating the Rating of Autonomous Driving Systems

The introduction of autonomous vehicles (AVs) is revolutionizing the way we perceive transportation. AVs hold the potential to significantly reduce accidents, alleviate traffic congestion, and offer mobility solutions for individuals unable to drive. Nevertheless, as AVs are deployed and tested in real-world settings, they encounter numerous challenges. These encompass navigating complex traffic scenarios, effectively interacting with other road users, and making decisions that balance safety and ethical considerations.

Autonomous driving systems (ADS) consist of various components, including perception, planning, decision-making, and control systems. These components collaborate to enable the vehicle to drive safely, respond effectively to environmental changes, and make optimal decisions. However, while these systems are typically developed independently, their seamless integration into a cohesive and fully functional system remains a challenge.

Organizational algorithms are crucial for integrating the various subsystems of autonomous vehicles (AVs) and ensuring their seamless collaboration. These algorithms allow the vehicle to adapt to abrupt environmental changes and make real-time decisions. By coordinating the interactions among system components, organizational algorithms enable AVs to make safe and efficient decisions. As AVs become increasingly prevalent, the demand for robust organizational algorithms is growing. This study focuses on designing and enhancing these algorithms to improve the efficiency, safety, and ethical decision-making capabilities of AVs.

Perception in the vehicle Autopilot system

This component is tasked with aggregating data from multiple sensors to construct a comprehensive understanding of the vehicle's surroundings. Advanced algorithms analyze the raw sensor data to detect, classify, and track objects in real-time, while also determining distances. Collectively, the integrated functions of sensors, sensing, and localization form the environment-awareness module, as these components are interdependent and collaborate to acquire information about the vehicle's surroundings and its precise position within the environment.

In summary, to address the complexities of diverse road environments, the sensing process frequently employs multi-sensor fusion detection solutions. Modern high-level autonomous driving systems are equipped with a variety of sensors, including LiDAR, millimeter-wave radar, cameras, and GPS, forming an interconnected network akin to a small-scale Internet of Things (IoT) among these sensors. If the vast amounts of sensor data are not processed accurately or effective collaboration between sensors is lacking, substantial amounts of unusable data will be generated. This can lead to false perceptions within the autonomous driving system, thereby posing risks to vehicle operation. Consequently, sensor fusion is critical for integrating data from various types of sensors located in different positions to generate a unified and precise representation of the environment. As depicted in Fig. 2, the IoT requires the fusion of multiple data sources alongside analytics to enhance understanding of hidden data patterns, eliminate uncertain data, and ultimately facilitate rapid decision-making.In an automated vehicle driving system, the environment perception module serves as the foundational component. If this module produces incomplete or ambiguous results, subsequent processes such as data fusion, semantic segmentation, and object tracking will also be compromised, potentially posing significant risks to both drivers and vehicles. Consequently, achieving stable and efficient target detection and recognition is a critical task in the vehicle environment perception phase. This process directly influences the planning, decision-making, and path control stages of intelligent vehicles. Currently, widely adopted methods for vehicle target detection include machine vision, millimeter-wave radar, LiDAR, and others.

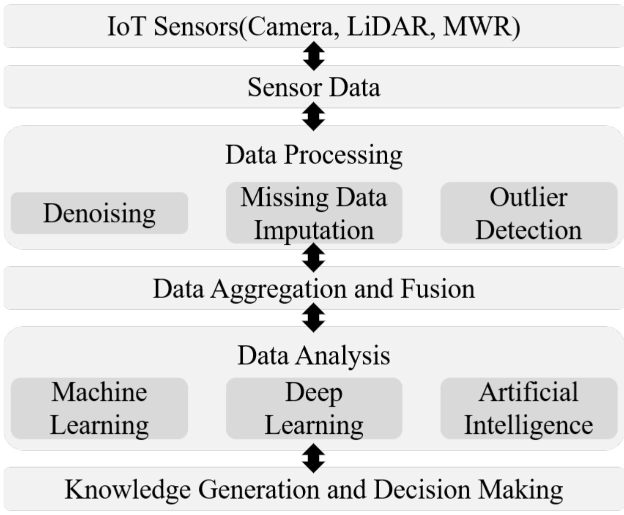

Fig. 2. IoT data analytics model

Machine vision remains the earliest and most extensively utilized sensor for vehicle detection [3]. These sensors are capable of capturing a wealth of perceptual information, including color, grayscale, texture, and semantics, from the traffic environment, demonstrating a robust ability to represent detailed vehicle targets. With the continuous improvement in performance and reduction in cost of machine vision systems, an increasing number of companies and researchers are adopting various types of cameras as fundamental solutions for environmental sensing in smart vehicles, exemplified by Mobileye [4] and Tesla [5].

Based on the underlying algorithmic principles, machine vision-based vehicle detection methods can be categorized into three distinct types: prior knowledge-based, machine learning-based, and deep learning-based

Perception based on prior knowledge. An approach based on prior knowledge considers that, in contrast to the complex and dynamically changing traffic environment, vehicles on the road possess fixed appearances and characteristics. These features can be leveraged for vehicle detection through the Hypothesis Generation (HG) and Hypothesis Verification (HV) phases. Specifically, the algorithm first generates a region of interest (ROI) based on the results of vehicle detection during the HG phase, and then verifies the presence of a vehicle within the ROI during the HV phase. This two-step process not only significantly enhances the efficiency of vehicle recognition but also reduces the likelihood of erroneous identifications. In this study, five distinct vehicle features are defined: The presence of shadows beneath the vehicle, with a significantly lower grayscale value compared to the surrounding road [6],[7]; the color of the headlights appears bright red, and information extracted from the red channel can accurately locate the position of the taillights of the preceding vehicle [8],[9]; the edges of vehicles exhibit distinct linear characteristics in contrast to the surrounding environment [10],[11]; the color of vehicles demonstrates continuity and aggregation within the image, which can be effectively extracted using different color channels combined with threshold segmentation [12],[13]; nearly all vehicles possess symmetry, allowing for easy segmentation from the environment via symmetry-based judgment, which also aids in refining vehicle boundaries [14],[15]. Based on these five features, recognition can comprehensively cover vehicles in any scene environment. Additionally, the symmetry feature of vehicles is frequently utilized for verification in the HV link.

Perception based on machine learning. Although feature engineering based on prior knowledge can effectively capture the static attributes of vehicles, it still encounters challenges such as sudden lighting changes and occlusion in dynamic traffic scenarios. To improve the generalization capability of the model, researchers have integrated machine learning approaches to adaptively learn more discriminative vehicle representations through data-driven strategies, thereby addressing the limitations of manually designed features.

Machine learning is employed in vehicle detection tasks to encode vehicle image information using manually designed features. Through specific algorithms, high-dimensional vehicle information is mapped into low-dimensional representations that are more suitable for training machine learning models. Subsequently, the model is optimized through iterative training to achieve the desired performance [16]. Feature extraction and classifier training constitute the core of machine learning-based vehicle detection methods. Vehicle feature extraction shares many commonalities with face recognition and pedestrian detection. Commonly used features such as Haar/Haar-like features, Histogram of Oriented Gradients (HOG) features, and Deformable Parts Model (DPM) features have been validated in vehicle detection. In addition, other feature extraction algorithms like wavelet features, Principal Component Analysis (PCA), Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and Local Binary Patterns (LBP) are also widely applied. Sometimes, combining these features can yield richer representations for vehicle detection [17]-[19]. Classifier training involves further processing of the extracted features to distinguish between vehicle and non-vehicle targets. Common classifiers in machine learning include K-nearest neighbors (KNN), Support Vector Machines (SVM), Decision Trees, and AdaBoost. Different combinations of feature extraction techniques and classifiers can lead to various vehicle detection algorithms, as illustrated in Table 1.

Table 1

Different combinations of feature extraction and classifiers

Classifier	Features	Advantage
SVM	HOG [20],[21]	Excellent performance in target detection and pedestrian recognition, particularly in the efficient processing of edge and gradient information.

Classifier	Features	Advantage
	DPM [22]-[24]	Strong robustness in complex deformation target detection, and is well-suited for handling occlusion and multi-attitude scenarios.
	Wavelet Features [25],[26]	Outstanding performance in image classification and multi-resolution analysis makes it particularly suitable for capturing both local details and global features effectively.
	PCA [27],[28]	High efficiency in dimensionality reduction and data classification, making it suitable for processing high-dimensional data and enhancing computational performance.
	LBP [29],[30]	Excellent performance in texture classification and facial recognition makes it particularly suitable for extracting local texture features with high accuracy and reliability.
	SURF [31]	Excellent performance in texture classification and facial recognition makes it particularly suitable for extracting local texture features with high accuracy and reliability.
	SIFT [32],[33]	High-precision object recognition and matching in complex scenes, which is suitable for processing scale-invariant features.
AdaBoost	SIFT [32],[33]	Excellent performance in target detection and feature matching, making it suitable for integrating local features to enhance classification accuracy.
	Haar / Haar-like [34-37]	High efficiency in real-time target detection makes it suitable for quickly extracting simple features and constructing robust classifiers.
KNN	Haar-like [38]	Performs effectively on simple classification tasks, making it suitable for small-scale datasets and real-time applications.
DecisionTree	HOG [38],[39]	Stable performance in object detection and image classification tasks, making it suitable for processing structured features and generating interpretable classification models.

Perception based on deep learning. Traditional machine learning approaches depend on the integration of manual feature extraction and classification algorithms, with their performance constrained by the capability of feature representation. As hardware computational power has advanced and data scales have expanded, deep learning has realized the joint optimization of feature extraction and target recognition via an end-to-end learning framework, offering a novel paradigm shift for perception.

Deep learning-based methods, particularly target-level vehicle detection, serve as a general approach for object recognition. Almost all target-level models can be applied to vehicle detection. The diversity in algorithm architectures has led to various detection methods, which are primarily categorized into one-stage and two-stage detection. One-stage detection demonstrates a significant advantage in real-time performance, while two-stage detection achieves higher accuracy due to the adoption of candidate region generation strategies, making it effective for detecting small target vehicles. However, with ongoing research, one-stage detection methods have gradually overcome issues related to low accuracy and difficulty in detecting small targets, achieving both speed and precision, thus becoming the mainstream algorithm today. Additionally, the choice of backbone network plays a crucial role in balancing real-time performance and accuracy. For instance, in the Faster R-CNN model, ZFnet, a 6-layer network, processes images at 17 FPS with an average accuracy (mPA) of 62%, whereas VGG16, a 16-layer network, processes images at only 5 FPS but achieves an mPA of 73% [40]. Compared with the two, ZFNet is 3.4 times faster but suffers an 11% drop in accuracy, indicating the importance of balancing real-time performance and accuracy [41]. The second approach involves vehicle semantic segmentation based on deep learning, which achieves better recognition of vehicle contours compared to target-level methods, thereby enhancing accuracy [42]. Semantic segmentation can be categorized into fully supervised algorithms and weakly supervised algorithms depending on the method of data labeling [43]. Fully supervised algorithms rely on manual labeling to achieve superior segmentation results, while weakly supervised algorithms reduce dependence on manual labeling at the cost of slightly lower segmentation quality. To achieve better autopilot outcomes, this paper focuses solely on the application of fully supervised algorithms. Representative semantic segmentation methods such as Mask R-CNN [44], DeepMask [45], and SharpMask [46] adopt a two-step process: first, the approximate vehicle candidate region is extracted using object detection, and then the pixels within the region are classified using a trained classifier. This process depends heavily on the extraction of the vehicle candidate region, resulting in relatively low real-time performance. Google Lab enhanced the fully convolutional network with greater computational power and proposed the DeepLab/V2/V3 models [47],[48], which extract as much image information as possible through techniques such as probability maps, image pyramids, and dilated convolutions, thereby improving the accuracy of vehicle semantic segmentation. However, due to the higher hardware requirements of semantic segmentation models, inference speed is often slower for these models. Therefore, designing an efficient lightweight semantic segmentation model remains a critical task for future research.

Millimeter-wave radar represents a more mature vehicle detection technology compared to machine vision. Even before the advent of machine vision, millimeter-wave radar was already widely used to assist in recognizing the peripheral information around vehicles. Millimeter-wave radar can detect obstacles within blind spots and measure the distance between the vehicle and the obstacle. It demonstrates excellent general adaptability under adverse weather conditions and is capable of obtaining both the depth information and motion status of targets. Despite the rapid development of various sensors today, millimeter-wave radar remains an indispensable component of the vehicle perception system. Depending on the accuracy of the millimeter-wave radar, it can be divided into target-level and image-level.

Target-level radar. Target-level radars are equipped with built-in chips and algorithms that directly convert the reflected signals received by the radar into target information, including but not limited to distance, angle, and reflection intensity. However, target-level radars can be easily confused by stationary targets and false targets. Stationary targets refer to objects such as streetlights, trees, and guardrails that consistently exist in the environment. These targets exhibit significantly different radar cross-sections and signal-to-noise ratios compared to vehicular targets, which can be effectively filtered out by setting appropriate thresholds. Additionally, machine learning methods such as Deep Belief Networks (DBN) [49] and Artificial Neural Networks (ANN) [50] can be employed to further categorize radar targets. False targets, on the other hand, arise due to factors like road bumps or signal interference and do not truly exist. Unlike stationary targets, false targets persist for a shorter duration and can be mitigated using techniques such as Kalman Filtering [51], Target Tracking Algorithms [52], and Life Cycle Algorithms [53].

Image-level radar. Target-level radar exhibits several limitations. As a millimeter-wave radar that was initially employed in vehicle perception systems, it demonstrates significant shortcomings when addressing complex environments. To address these challenges, researchers have initiated investigations into image-level radar.

Image-level radar: As the frequency restrictions on radar in the civil field are gradually being lifted, millimeter-wave radar is achieving higher operating frequencies and evolving into image-level millimeter-wave radar. This type of radar can convert radar signals into image signals by generating radar projection maps and point cloud maps. High-frequency radar signals can be used to generate point cloud maps, which serve as the basis for applying machine learning and deep learning methods for vehicle detection. Zhao et al. extracted feature vectors from point clouds using the DBSCAN algorithm and combined them with an SVM classifier to characterize vehicles and capture targets [54]. Guan et al. utilized a GAN network and a point cloud segmentation algorithm to further improve vehicle detection accuracy [55]. Additionally, coherent imaging radar offers higher resolution and generates SAR maps that are compatible with various image-based target detection algorithms such as YOLO [56], SSD [57], CNN [58], [59], among others.

Overall, millimeter-wave radar is insufficient to meet the demands of an autonomous driving system on its own. However, it remains an essential component of the system and must be integrated with data from other sensors to create an accurate and comprehensive perception of the vehicle's surroundings.

With the rapid advancement of sensor technology, LiDAR has become increasingly prevalent in vehicle target detection systems. By actively emitting and receiving laser signals, LiDAR is capable of detecting obstacles and operates effectively even in low-light conditions as an active sensing modality. Moreover, LiDAR can capture detailed three-dimensional environmental information with greater precision compared to image-level millimeter-wave radar, making it a critical component for achieving robust autonomous driving environment perception. LiDAR detection methods can be classified into traditional and deep learning-based approaches depending on the underlying algorithms.

LiDAR based on traditional methods. Constrained by the characteristics of lasers, traditional methods generate sparse and large-scale point cloud data using LiDAR, making it challenging to process raw data. Therefore, traditional methods typically transform point cloud data into 2D/2.5D representations, such as grid maps [60]-[62] and range images [63], [64]. Point cloud segmentation can significantly enhance the real-time performance of traditional methods while drastically reducing computational complexity. By leveraging road surface features such as reflection angle, reflection intensity, and continuity, road information in the point cloud can be efficiently segmented from vehicle and obstacle data when appropriate feature thresholds are set [65], [66]. However, the effectiveness of point cloud segmentation diminishes on uneven roads. To address this issue, some scholars have proposed a novel approach: first, fit the road surface to level it [67]-[70]; second, cluster point clouds with similar features using clustering algorithms; and finally, classify targets using pattern matching or machine learning algorithms [71]-[74]. This method resembles the prior knowledge-based approach used in machine vision, relying heavily on prior knowledge rather than focusing on intrinsic point cloud information, thus limiting its generalization capability.

LiDAR based on deep learning. To address the limitations of traditional methods, especially their reliance on manually designed features and prior knowledge, deep learning-based methods have emerged as a powerful alternative, offering higher adaptability and detection accuracy.

Deep learning methods have the capability to autonomously learn diverse information within point clouds, thereby enriching data features and enhancing detection accuracy. Deep learning-based vehicle target detection approaches can be classified into four categories: direct detection methods, projection-based methods, voxel-based models, and point cloud-based models. Direct detection methods utilize 3D neural networks to process point cloud data directly, preserving as much raw data as possible. However, this approach tends to be computationally intensive, which may hinder real-time system performance. Projection-based methods transform 3D point cloud data into 2D front or top views for feature extraction, significantly reducing computation time. Nevertheless, this transformation often results in the loss of depth information, leading to reduced accuracy in feature extraction [75]-[77]. Voxel-based models segment point cloud data into uniformly sized grids (voxels) to handle unordered point cloud data. For instance, reference [78] introduces the VoxelNet model, which first divides large-scale 3D grids using CNNs and subsequently employs a PointNet-like structure for constructing small-sized voxels. In this model, the voxel size critically impacts both system speed and accuracy, making it challenging to balance these factors. To address this issue, Ye et al. proposed a hybrid voxel model that integrates voxels of varying scales to improve performance [79]. Compared with the first three methods, the point cloud-based model is evidently more direct in processing data and thus better preserves depth information, which plays a crucial role in constructing the environment around the vehicle. Qi et al. [80] introduced the PointNet model, which adjusts the state of the point cloud via feature and spatial transformations and subsequently extracts features from the entire point cloud using maximum pooling. This approach effectively addresses the issue of feature disorder but simultaneously constrains the model's generalization ability at a fundamental level. Building upon this, researchers enhanced the capture of local features and proposed the PointNet++ model [81], which successfully tackles the problem of uneven data density and further enhances the performance of vehicle detection.

In summary, environment sensing for autonomous driving systems demands a substantial amount of experimental data to ensure their reliability and robustness. Traditional real-vehicle testing is expensive, time-consuming, and limited in test scenarios, making it incapable of covering the countless complex and dynamic road conditions worldwide. Therefore, simulation is essential for modeling the performance of autonomous driving systems under real-world road conditions. Simulation does not rely on the availability of physical sensors, vehicles, or drivers, and its scalability improves with advancements in computing power. This approach enables the investigation of rare yet critical road situations [82],[83]. Although autonomous driving system simulations are widely used, they often fall short in simultaneously ensuring both accuracy and speed. Simulating real-world environments poses significant challenges, particularly regarding the influence of ambient light on LiDAR and vision cameras. Many existing models struggle to replicate complex lighting conditions accurately. Nils Hirsenkorn et al. [84] from the University of Munich pioneered a simulation model using the OptiX ray tracing engine, significantly enhancing the real-time performance of environment perception algorithms.

Localization is essential for ensuring that the vehicle accurately determines its position on the map, which is critical for safe navigation in autonomous driving systems. The localization process enables the vehicle to precisely identify its location on the map, thereby supporting safe and reliable autonomous navigation. In a previous article, simultaneous localization and mapping (SLAM) algorithms based on the vehicle's onboard sensors—such as millimeter-wave radar, LiDAR, and cameras—were discussed due to their close relationship with environmental perception. These will not be elaborated upon here. Beyond SLAM, global positioning, map-based positioning, and path tracking are also employed to enhance the accuracy of localization within autonomous driving systems.

Global localization commonly relies on external signals, such as GPS or Global Navigation Satellite Systems (GNSS). Although GPS receivers are now widely accessible and can determine a vehicle's position by receiving signals from at least four satellites, their effectiveness is limited in certain scenarios. While GPS provides a broad field of view and plays a crucial role in navigating vehicles in environments like city streets and highways, it becomes unreliable in areas such as tunnels where signals are blocked or viaducts where multiple roads overlap. Due to its inherent limitations in accuracy and applicability in specific scenes, global positioning often needs to be complemented by more precise local positioning methods, such as SLAM, to achieve higher accuracy [85].

Map-based positioning determines the current location by comparing real-time onboard sensor data with pre-drawn, high-resolution environmental map data. Compared to GPS positioning, map-based positioning offers higher accuracy. Initially, digital maps were developed for driver assistance systems to enhance vehicle and driver safety, as seen in Advanced Driver Assistance Systems (ADAS) [86]. In recent years, high-definition (HD) maps have been introduced, providing detailed map information at a high resolution. These HD maps are structured in layers, encapsulating static information such as road geometry, lane markings, traffic signals, and other static features relevant to the vehicle's surroundings.

High-definition maps can be categorized into two primary models: feature-based and dense models. Feature-based models focus on static elements, including road surface details, lane configurations, traffic signs, and other fixed infrastructure, which are manually recorded and hierarchically organized into a layered structure, often referred to as static targets. Dense models, captured through sensors, encompass dynamic elements such as pedestrians, vehicles, and other moving objects, along with additional contextual information related to the vehicle's movement. This dynamic data is integrated into the layered stack as supplementary dynamic layers.

More advanced Local Dynamic Maps (LDMs) extend the static and dynamic layers mentioned above by incorporating additional information layers. These can include details such as neighborhood congestion levels, traffic light timings, and highly dynamic data layers to facilitate vehicle-to-everything (V2X) communication connections. To achieve the centimeter-level localization necessary for autonomous driving, high resolution is critical. Constructing high-resolution maps requires repeated collection of environmental data, followed by the use of algorithms like SLAM to convert and fuse measurements from multiple sensors [87],[88], thereby generating sufficiently accurate high-resolution static layers.

Nowadays, a variety of map formats have been developed for autonomous driving systems, with Lanelet2 and OpenDrive being the most widely adopted. Lanelet2 provides rich semantic road information, such as traffic signs, speed limits, and lane restrictions, enabling precise navigation in urban environments and making it highly suitable for autonomous driving systems. In contrast, OpenDrive is better suited for road network modeling and simulation, focusing primarily on detailed road structures rather than handling complex semantic data or real-time updates [89],[90], as illustrated in Table 2.

Table 2

Advantages and Disadvantages of Lanelet2 Maps vs. OpenDrive Maps

Feature	Lanelet2	OpenDrive
Map Type	Semantic, detailed, high-definition maps	Road geometry-focused, vector-based maps
Coverage	Limited in some regions; crowdsourced via OSM	Extensive global coverage; widely adopted
Map Content	High-level semantic road information (traffic signs, speed limits, lanes)	Primarily road geometry and lane positioning
Use Case	Autonomous driving, high-precision navigation	Road network modeling, simulations
Data Complexity	High (more detailed)	Low to medium (simplified road network)
Real-time Updates	Can integrate real-time updates via OSM	Typically static; updates require manual intervention
Adoption	Growing, particularly in autonomous driving	Highly adopted in simulation and road modeling
Main Strength	Detailed road semantics, lane-level control	Simplicity, large-scale simulations, widespread adoption
Limitations	Complex and requires high-precision sensors	Lacks semantic details, requires additional data for decision-making

Planning and Decision-Making in the vehicle Autopilot system

This component entails determining the appropriate actions for the vehicle to take in any given scenario. Path planning techniques can be categorized into three main types: traditional methods, machine learning and deep learning approaches, and meta-heuristic optimization algorithms. Traditional methods often face limitations that make it challenging for them to outperform the other two categories due to their inherent constraints. Machine learning and deep learning-based path planning techniques excel in learning and adapting swiftly in known environments. Meanwhile, meta-heuristic optimization algorithms serve as versatile processors capable of addressing complex problems. Additionally, hybrid algorithms that integrate these three techniques are gradually emerging, leveraging the respective strengths of each approach [91].

Risk assessment and prediction constitutes the initial phase of path planning. This process relies on environmental perception data to evaluate the overall risk of the current driving scenario and predict the movements of surrounding vehicles or pedestrians, thereby enhancing the emergency risk avoidance capabilities of autonomous vehicles. Risk assessment and prediction can be categorized into three main types: uncertainty and risk evaluation, dynamic target behavior forecasting, and driver's driving style classification.

1. The primary objective of uncertainty and risk assessment is to monitor the overall road environment, typically achieved through the integration of radar systems, camera technologies, and neural network algorithms.
2. Dynamic target behavior prediction involves not only surrounding vehicles and pedestrians but also all road traffic participants who are subjectively influenced by their consciousness. Accurate prediction of these participants enables the system to respond more swiftly in unexpected situations. However, existing algorithms often fail to focus on a single target for an extended period, resulting in insufficient data accumulation for predicting the behavior of each individual target.
3. In contrast, the categorization of a driver's driving style is arguably the most critical component of the risk assessment and prediction process. Driving styles can be classified based on data such as the frequency of lane changes over a specific time period, vehicle acceleration patterns, and other relevant metrics, following the detection of the target vehicle [92]. In this context, various algorithms, including neural networks, support vector machines (SVM), principal component analysis (PCA), and K-means clustering, are commonly employed to categorize driving styles [93].

The primary objective of the path planning phase is to determine a safe and efficient travel route for the vehicle. Initially, global planning is conducted using map and GPS data to navigate the optimal road from the starting point to the destination.

Subsequently, continuous local planning is performed along this forward route to address traffic signals, avoid obstacles, and prevent potential collisions. However, as the number of available nodes increases during local planning, the computational complexity grows exponentially, making the task of identifying the shortest collision-free path a significant challenge that remains to be addressed in the path planning process.

Behavioral decisions entail selecting actions in complex scenarios, such as interacting with other road users, handling accidents, and adapting to dynamic environments. This encompasses rule-based systems, behavior prediction, reinforcement learning, among others. Behavior prediction and reinforcement learning have been elaborated upon in the previous section. Rule-based systems involve employing predefined rules within traditional decision-making frameworks to direct vehicle behavior, such as stopping at red lights or yielding to pedestrians. While these rules are straightforward and effective in structured situations, they may exhibit insufficient adaptability in uncertain or ambiguous contexts.

Finally, the issue of ethical and moral safety decisions must be addressed. In emergency scenarios where an accident is inevitable, self-driving cars may need to make complex ethical judgments, such as whether to swerve into an obstacle to protect pedestrians or continue along their current trajectory. The development of these decision-making processes is ongoing and may draw upon ethical frameworks like utilitarianism or deontological ethics.

Summarizing the previous section, there are typically three approaches for conducting path planning and decision-making processes, which we will present in turn:

Traditional path planning algorithms are designed to address the challenge of identifying the optimal or suboptimal path from a starting point to an endpoint within a complex environment. These algorithms primarily encompass graph-based methods, sampling-based methods, gradient-based methods, optimization-based methods, and interpolation curve methods, among others.

Graph-based methods. Graph-based methods, such as Dijkstra and A*, are widely employed in path planning for autonomous driving systems. These methods represent the environment as a graph, where nodes correspond to potential locations and edges signify connections between these locations. The algorithm subsequently conducts a search to identify the shortest path from the start node to the destination node. While these methods yield precise solutions within a confined search space and exhibit strong performance in static environments by finding optimal paths, their computational demands can escalate rapidly in dynamic or complex scenarios. This increase in computation may render graph construction and path searching impractical for large-scale or continuous search spaces.

Although graph-based methods can theoretically generate optimal solutions in structured environments, their computational requirements escalate significantly in unstructured and dynamic scenarios. To balance real-time efficiency with path feasibility, sampling-based methods enable comprehensive exploration of high-dimensional state spaces using stochastic techniques, albeit with some compromise in optimality.

Sampling-based methods. Sampling-based methods, such as the Rapidly-exploring Random Tree (RRT), sample points within the state space and attempt to connect them to construct a tree structure. This tree is subsequently utilized to identify a feasible path from the start point to the destination. Sampling-based methods are generally faster than graph search methods, capable of addressing high-dimensional state spaces, and well-suited for dynamic and unknown environments. However, the generated paths may lack sufficient smoothness and exhibit "jitter," necessitating additional optimization steps to enhance both path efficiency and comfort.

Even though sampling-based methods enhances adaptability to the environment, the paths generated often encounter challenges in terms of smoothness and comfort. The potential field gradient method achieves continuous optimization through physical modeling, transforming the process of path search into a trajectory optimization problem within the potential energy field. This provides an innovative approach for ensuring smooth motion under dynamic constraints.

Gradient-based methods. Gradient-based methods leverage the gradient of a potential field to guide the search for a path. These methods usually define a potential function that quantifies the desirability of each point in the environment, and subsequently employ gradient descent or analogous techniques to identify a path that minimizes this potential. This approach is adept at managing complex constraints, such as obstacles and road boundaries, and in certain scenarios, produces smooth and efficient paths. Nevertheless, it is prone to converging on local optima in complex or unfamiliar environments and exhibits sensitivity to the formulation of the potential function and the selection of parameters.

The gradient-based methods tend to fall into local optima traps. Although it can address continuous constraints effectively, the inherent limitations of heuristic approaches should not be disregarded. Optimization-based methods grounded in mathematical programming facilitate precise path planning under global constraints via systematic modeling. Moreover, these methods unify multiple objectives, including safety and comfort, within an integrated framework for thorough consideration.

Optimization-based methods. Optimization-based methods for path planning in ADS utilize mathematical planning or optimization algorithms to determine optimal paths. These methods identify the optimal path by defining one or more objective functions and corresponding constraints. The objective function typically relates to factors such as path length, safety, and comfort, while constraints may encompass road boundaries, obstacle locations, and vehicle dynamics limitations, among others. Depending on actual requirements, different objective functions and constraints can be established, enabling the identification of an optimal path through mathematical planning to accommodate various driving scenarios and tasks. Due to their accuracy, flexibility, and scalability, optimization-based methods can be integrated with other algorithms, such as heuristic search algorithms and sampling algorithms, to enhance detection efficiency and path quality. Nevertheless, for large-scale or complex detection spaces, the computational demands of optimization algorithms may become substantial, potentially compromising real-time performance. Additionally, parameter selection can significantly impact the stability and performance of the optimization algorithm, with inappropriate parameters leading to suboptimal results.

Although optimization-based methods can effectively address multi-objective trade-off problems, discrete solutions frequently encounter continuity challenges when implemented at the vehicle execution level. As a post-processing module, interpolation curve methods convert discrete paths into continuous trajectories that comply with vehicle kinematics through parametric curve fitting. This ultimately ensures the closed-loop operation of the control layer interface.

Interpolation curve methods. The interpolation curve methods involve constructing and inserting a series of new data points, given a predefined set of nodes, to generate a path that satisfies specific requirements. In Autonomous Driving Systems (ADS), this method is frequently employed to refine and optimize initial paths generated by the global planner or other local planners. This ensures collision avoidance while adapting to the dynamic environment and adhering to vehicle constraints. Commonly used curves include Bezier curves, spline curves, polynomial curves, and gyratory curves. A key characteristic of the interpolation curve method is its requirement to handle a large number of data points and constraints, which can increase computational complexity. By strategically selecting appropriate interpolation curve types and integrating them with optimization algorithms and real-time feedback mechanisms, the path planning efficiency and driving safety of ADS can be significantly enhanced.

Machine learning and deep learning play a pivotal role in autonomous driving path planning algorithms. Machine learning models can effectively leverage rich vehicle sensor data and map information, enabling data-driven path planning by collecting and analyzing extensive historical driving data. Through this process, the models learn driving characteristics and optimal paths under various road conditions. Deep learning models, on the other hand, excel at capturing complex nonlinear relationships and efficiently fusing multi-sensor data to construct highly accurate environment models. This capability allows for more precise predictions of both the vehicle's driving path and potential future obstacles. Consequently, the autonomous driving system can make more accurate and intelligent path planning decisions while providing essential data support for subsequent planning stages. Both machine learning and deep learning models possess continuous learning and self-optimization capabilities, allowing them to adapt to diverse complex road conditions and dynamic environments by identifying patterns and rules within training datasets. This ensures that the autonomous driving system maintains efficient and stable operation even when faced with unexpected changes in the road environment. Additionally, these models can analyze vehicle driving behavior under different traffic flows, weather conditions, and road types, serving as a foundation for smarter, safer, and more secure path planning for autonomous driving systems.

1. Supervised and deep learning techniques have been utilized in path planning primarily in two ways: within the perception layer for processing images and sensory data, or as an end-to-end driving framework that incorporates path planning. The capacity to train with extensive datasets enhances the algorithm's ability to generalize, enabling it to deliver high-speed and accurate solutions in familiar scenarios. However, in unfamiliar scenarios or when anomalies occur, these techniques may exhibit suboptimal performance. This is due to their reliance on large volumes of labeled data and significant computational resources for model training. Additionally, end-to-end solutions can be challenging to interpret and debug, which may hinder their practical application.
2. In reinforcement learning, the intelligent agent interacts with its environment via sensors and learns through a trial-and-error approach. During the training process, the agent's performance is assessed using a reward function. In each state, the agent selects an action that transitions it to another state. If the new state aligns the agent closer to the goal, it receives a positive reward. The strength of reinforcement learning lies in its capacity to handle complex environments and dynamic constraints while discovering optimal strategies through continuous exploration and learning. However, reinforcement learning also encounters challenges, such as high computational demands, low sample efficiency, and difficulties in addressing partially observable environments. To address these issues, researchers are actively investigating novel algorithms and techniques, including deep reinforcement learning and inverse reinforcement learning.
3. Deep reinforcement learning integrates the representation learning capabilities of deep learning with the decision-making optimization capabilities of reinforcement learning, offering a robust learning framework for intelligences operating in complex environments. It can be classified into approaches such as deep Q-networks, gradient-based strategies, and model-based deep reinforcement learning, among others.

Meta-heuristic optimization techniques, such as Genetic Algorithm (GA), Differential Evolution (DE), Simulated Annealing (SA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO), serve as versatile problem-solving frameworks applicable to a broad spectrum of optimization problems, including path planning for autonomous driving systems. These methods possess robust global search capabilities for identifying near-optimal solutions, making them particularly suitable for complex, multi-constrained, and dynamic environments. Furthermore, they can be integrated with other algorithms to form hybrid approaches that enhance overall performance. However, meta-heuristic optimization is computationally demanding and may necessitate extended run times. The algorithm's parameter settings significantly influence its performance and require meticulous tuning. Additionally, these techniques often exhibit weaker local search capabilities, which can lead to suboptimal solutions (though this issue can be partially alleviated through algorithmic improvements). Due to the extensive number of meta-heuristic optimization algorithm derivatives, it is impractical to comprehensively introduce all variations here.

Path Tracking and Control in the vehicle Autopilot system

Path tracking and control play a pivotal role in automated vehicle systems, ensuring precise adherence to the intended trajectory while maintaining safety and stability. These systems enable vehicles to navigate diverse environments effectively, avoid obstacles, and respond dynamically to changing conditions such as traffic patterns and road alterations. Furthermore, they integrate seamlessly with other vehicle subsystems to execute planned maneuvers smoothly, including lane changes and turns. By minimizing erratic movements, these systems enhance passenger comfort, support ethical decision-making in critical situations, and improve driving efficiency through optimized route planning and speed regulation.

Path tracking refers to the process of ensuring that a self-driving car adheres precisely to a planned route or trajectory. It facilitates smooth and safe navigation, enabling the vehicle to maintain its lane, follow curves, and respond effectively to environmental changes such as obstacles or traffic conditions. Path tracking is essential for implementing decisions made by the pathfinding system and ensuring that the vehicle remains on course while behaving predictably. Motion control algorithms translate trajectory decisions into real-time actions, such as steering adjustments, braking, and acceleration. Proportional-Integral-Derivative (PID) control, speed control, and path tracking are all critical components of vehicle motion control that involve decision-making processes. Basic PID controllers can be employed to regulate a vehicle's steering and speed. However, more advanced methods, such as adaptive PID controllers or fuzzy logic control, are often better suited for handling complex driving scenarios. Speed control, which falls under longitudinal control, adjusts the vehicle's speed based on factors like road gradients, surrounding vehicles, and speed limits. The primary objective of path tracking is to ensure that the vehicle follows the planned trajectory as closely as possible while accounting for dynamic constraints. Pure pursuit and Stanley controllers are commonly utilized for lateral control to keep the vehicle within its designated lane.

Based on different strategies, path tracing can be classified into four distinct categories:

Model Predictive Control (MPC) is an advanced control strategy that leverages a vehicle dynamics model to predict future states and determines optimal control inputs via an optimization algorithm. In path tracking, MPC can effectively account for the nonlinear characteristics and constraints of the vehicle, generating smooth and stable control commands. This makes it well-suited for addressing multivariate and Mult constraint optimization problems. MPC's ability to predict future states and adapt to the vehicle's nonlinear characteristics provides a degree of foresight and stability. However, MPC is also associated with challenges such as high computational demands and limited real-time performance, particularly in complex environments. In these scenarios, the control effectiveness is significantly influenced by the accuracy of the model and imposes stringent hardware requirements.

Sliding mode control (SMC) is a robust control strategy that ensures the system state converges to the sliding mode surface within a finite time by designing an appropriate sliding mode surface. The system then moves along this surface toward the equilibrium point. In path tracking applications, SMC demonstrates robustness against vehicle model uncertainties and external disturbances. Additionally, the algorithm is straightforward and easy to implement, as it does not require an exact mathematical model of the system. However, the effectiveness of the control is significantly influenced by the design of the sliding mode surface, which can be highly sensitive to system parameters and may induce vibrations during the control process.

Path tracking based on proportional-integral-derivative (PID) control is a classical and widely adopted control strategy that adjusts the control inputs by calculating the proportional, integral, and derivative terms of the deviation to bring the system state closer to the desired value. In path tracking, the PID control algorithm is simple and frequently utilized for controlling basic maneuvers such as steering, acceleration, and braking. While PID control is robust to variations in system parameters, it exhibits limited adaptability to nonlinear and time-varying systems and may result in overshooting and oscillations during the control process.

Path tracking based on fuzzy control and neural network control represents intelligent control strategies capable of addressing complex, uncertain, and nonlinear problems. In path tracking, these strategies adapt to dynamic changes in the vehicle and environment through learning and optimization processes. A precise mathematical model is not required, making them highly adaptive to nonlinear systems and uncertainties while effectively managing multivariate and complex constraints. However, these strategies also have notable disadvantages, including high initial investment costs, lengthy training and learning periods, and control performance that is heavily influenced by the quality of training data and the structure of the model. Additionally, in large-scale networks, the real-time performance of fuzzy and neural network control may be constrained due to computational limitations.

At this stage, this paper systematically reviews and analyzes the current advancements in automated vehicle driving technology across three key dimensions: perception, planning and decision-making, and control. It also identifies several significant challenges, thereby providing a clear direction for future research efforts in this field.

Conclusion

This paper provides a thorough review of the three core modules in autonomous driving systems: perception, planning, and control. It further investigates the current challenges and future research directions of this technology. The perception module leverages multi-sensor fusion and deep learning techniques to extract comprehensive environmental information. However, the integration and processing of sensor data remain challenging, particularly in complex and dynamic scenarios. Future research should prioritize the development of lightweight models and standardized testing frameworks to improve the system's real-time performance and robustness. Path planning and decision-making serve as the backbone of autonomous driving systems, integrating traditional algorithms, machine learning, and meta-heuristic optimization methods. Although existing algorithms demonstrate strong performance in known environments, their adaptability and generalization capabilities in complex and dynamic settings need further enhancement. Future studies should focus on refining decision-making processes for multi-agent coordination and intricate scenarios. Path tracking and control algorithms are crucial for ensuring the precise execution of planned trajectories while maintaining stability and safety in dynamic environments. Advanced techniques, such as model predictive control (MPC) and fuzzy control, exhibit superior performance under complex driving conditions. However, computational complexity and real-time requirements continue to pose significant challenges. Future research should focus on developing more computationally efficient control algorithms tailored for complex driving scenarios. Additionally, autonomous driving systems must address ethical decision-making in emergency situations. While current ethical frameworks provide a foundation, they remain insufficient for handling all possible scenarios. Future work should aim to establish a comprehensive and robust ethical decision-making framework to ensure safe and ethical behavior in uncertain and complex environments. Simulation testing remains a critical component for validating these systems.

References

1. A novel heuristic simulation - optimization method for critical infrastructure in smart transportation systems / A. Kumar [et al.] // International Journal of Communication Systems. — 2020. — Vol. 33. — No. 11. — P. e4397. — https://doi.org/10.1002/dac.4397
2. An overview of IoT sensor data processing, fusion, and analysis techniques / R. Krishnamurthi [et al.] // Sensors. — 2020. — Vol. 20. — No. 21. — P. 6076. — https://doi.org/10.3390/s20216076
3. Bertozzi M., Broggi A., Fascioli A. Vision-based intelligent vehicles: State of the art and perspectives // Robot. Auto. Syst. — 2000. — Vol. 32. — No. 1. — P. 1-16. — https://doi.org/10.1016/S0921-8890(99)00125-6

4. Stein G. P. System and method for detecting obstacles to vehicle motion and determining time to contact therewith using sequences of images : U.S. Patent 7113867. — 2006.
5. Endsley M. R. Autonomous driving systems: A preliminary naturalistic study of the Tesla models // J. Cogn. Eng. Decis. Making. — 2017. — Vol. 11. — No. 3. — P. 225-238. — https://doi.org/10.1177/1555343417695197
6. Shadow-based vehicle detection in urban traffic / M. Ibarra-Arenado [et al.] // Sensors. — 2017. — Vol. 17. — No. 5. — P. 975. — https://doi.org/10.3390/s17050975
7. Chen X., Chen H., Xu H. Vehicle detection based on multifeature extraction and recognition adopting RBF neural network on ADAS system // Complexity. — 2020. — Art. no. 8842297. — https://doi.org/10.1155/2020/8842297
8. Pradeep C. S., Ramanathan R. An improved technique for night time vehicle detection // Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI). — 2018. — P. 508-513. — https://doi.org/10.1109/ICACCI.2018.8554712
9. Satzoda R. K., Trivedi M. M. Looking at vehicles in the night: Detection and dynamics of rear lights // IEEE Trans. Intell. Transp. Syst. — 2019. — Vol. 20. — No. 12. — P. 4297-4307. — https://doi.org/10.1109/TITS.2016.2614545
10. Multiscale edge fusion for vehicle detection based on difference of Gaussian / K. Mu [et al.] // Optik. — 2016. — Vol. 127. — No. 11. — P. 4794-4798. — https://doi.org/10.1016/j.ijleo.2016.01.017
11. Vehicle detection based on underneath vehicle shadow using edge features / S. A. Nur [et al.] // Proc. 6th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE). — 2016. — P. 407-412. — https://doi.org/10.1109/ICCSCE.2016.7893608
12. Russell A., Zou J. J. Vehicle detection based on color analysis // Proc. Int. Symp. Commun. Inf. Technol. (ISCIT). — 2012. — P. 620-625. — https://doi.org/10.1109/ISCIT.2012.6380975
13. Shao H. X., Duan X. M. Video vehicle detection method based on multiple color space information fusion // Advanced Materials Research. — 2012. — Vol. 546. — P. 721-726. — https://doi.org/10.4028/www.scientific.net/AMR.546-547.721
14. Video vehicle detection algorithm based on edge symmetry / L. Gao [et al.] // J. Beijing Univ. Aeronaut. Astronaut. — 2008. — Vol. 34. — No. 9. — P. 1113-1116.
15. Teoh S. S., Braunl T. Symmetry-based monocular vehicle detection system // Mach. Vis. Appl. — 2012. — Vol. 23. — No. 5. — P. 831-842. — https://doi.org/10.1007/s00138-011-0355-7
16. A review of vehicle detection techniques for intelligent vehicles / Z. Wang [et al.] // IEEE Transactions on Neural Networks and Learning Systems. — 2022. — Vol. 34. — No. 8. — P. 3811-3831. — https://doi.org/10.1109/TNNLS.2021.3128968

17. Yang X. F., Yang Y. A method of efficient vehicle detection based on HOG-LBP // Comput. Eng. — 2014. — Vol. 40. — No. 9. — P. 210-214.
18. Arunmozhi A., Park J. Comparison of HOG, LBP and Haar like features for on-road vehicle detection // Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT). — 2018. — P. 0362-0367. — https://doi.org/10.1109/EIT.2018.8500159
19. Tao H., Lu X. Smoke vehicle detection based on multi-feature fusion and hidden Markov model // J. Real-Time Image Process. — 2020. — Vol. 17. — No. 3. — P. 745-758. — https://doi.org/10.1007/s11554-019-00856-z
20. A two-stage approach to people and vehicle detection with HOG-based SVM / F. Han [et al.] // Proc. Perform. Metrics Intell. Syst. Workshop. — 2006. — P. 133-140.
21. Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos / X. Cao [et al.] // Proc. 18th IEEE Int. Conf. Image Process. — 2011. — P. 2421-2424. — https://doi.org/10.1109/ICIP.2011.6116132
22. Object detection with discriminatively trained part-based models / P. F. Felzenszwalb [et al.] // IEEE Trans. Pattern Anal. Mach. Intell. — 2010. — Vol. 32. — No. 9. — P. 1627-1645. — https://doi.org/10.1109/TPAMI.2009.167
23. Sivaraman S., Trivedi M. M. Vehicle detection by independent parts for urban driver assistance // IEEE Trans. Intell. Transp. Syst. — 2013. — Vol. 14. — No. 4. — P. 1597-1608. — https://doi.org/10.1109/TITS.2013.2264314
24. On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation / H. T. Niknejad [et al.] // IEEE Trans. Intell. Transp. Syst. — 2012. — Vol. 13. — No. 2. — P. 748-758. — https://doi.org/10.1109/TITS.2012.2187894
25. Sun Z., Miller R., Bebis G., DiMeo D. A real-time precrash vehicle detection system // Proc. 6th IEEE Workshop Appl. Comput. Vis. (WACV). — 2002. — P. 171-176.
26. Sun Z., Bebis G., Miller R. On-road vehicle detection using evolutionary Gabor filter optimization // IEEE Trans. Intell. Transp. Syst. — 2005. — Vol. 6. — No. 2. — P. 125-137. — https://doi.org/10.1109/TITS.2005.848363
27. Wang C. C. R., Lien J. J. J. Automatic vehicle detection using local features-A statistical approach // IEEE Trans. Intell. Transp. Syst. — 2008. — Vol. 9. — No. 1. — P. 83-96. — https://doi.org/10.1109/TITS.2007.908572
28. Sun R. M. Z., Bebis G. Monocular precrash vehicle detection: Features and classifiers // IEEE Trans. Image Process. — 2006. — Vol. 15. — No. 7. — P. 2019-2034. — https://doi.org/10.1109/TIP.2006.877062
29. Online vehicle detection using Haar-like, LBP and HOG feature based image classifiers with stereo vision preselection / D. Neumann [et al.] // Proc. IEEE Intell. Vehicles Symp. (IV). — 2017. — P. 773-778. — https://doi.org/10.1109/IVS.2017.7995810

30. Combining region-of-interest extraction and image enhancement for nighttime vehicle detection / H. Kuang [et al.] // IEEE Intell. Syst. — 2016. — Vol. 31. — No. 3. — P. 57-65. — https://doi.org/10.1109/MIS.2016.17
31. Hsieh J.-W., Chen L.-C., Chen D.-Y. Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition // IEEE Trans. Intell. Transp. Syst. — 2014. — Vol. 15. — No. 1. — P. 6-20. — https://doi.org/10.1109/TITS.2013.2294646
32. Ho W. T., Lim H. W., Tay Y. H. Two-stage license plate detection using gentle adaboost and SIFT-SVM // Proc. 1st Asian Conf. Intell. Inf. Database Syst. — 2009. — P. 109-114.
33. Chen X., Meng Q. Vehicle detection from UAVs by using SIFT with implicit shape model // Proc. IEEE Int. Conf. Syst., Man, Cybern. — 2013. — P. 3139-3144. — https://doi.org/10.1109/SMC.2013.535
34. Han S., Han Y., Hahn H. Vehicle detection method using Haar like feature on real time system // World Acad. Sci., Eng. Technol. — 2009. — Vol. 59. — No. 35. — P. 455-459.
35. Wen X., Shao L., Fang W., Xue Y. Efficient feature selection and classification for vehicle detection // IEEE Trans. Circuits Syst. Video Technol. — 2015. — Vol. 25. — No. 3. — P. 508-517. — https://doi.org/10.1109/TCSVT.2014.2358031
36. Negri P., Clady X., Prevost L. Benchmarking Haar and histograms of oriented gradients features applied to vehicle detection // Proc. ICINCO-RA. — 2007. — Vol. 1. — P. 359-364. — https://doi.org/10.5220/0001637503590364
37. Haselhoff A., Kummert A. A vehicle detection system based on Haar and triangle features // Proc. IEEE Intell. Vehicles Symp. — 2009. — P. 261-266. — https://doi.org/10.1109/IVS.2009.5164288
38. Ali A. M., Eltarhouni W. I., Bozed K. A. On-road vehicle detection using support vector machine and decision tree classifications // Proc. 6th Int. Conf. Eng. (MIS). — 2020. — P. 1-5. — https://doi.org/10.1145/3410352.3410803
39. Gonzalez J. P., Ozguner U. Lane detection using histogram based segmentation and decision trees // Proc. IEEE Intell. Transp. Syst. (ITSC). — 2000. — P. 346-351. — https://doi.org/10.1109/ITSC.2000.881084
40. Faster R-CNN: Towards real-time object detection with region proposal networks / S. Ren [et al.] // Proc. Adv. Neural Inf. Process. Syst. — 2015. — P. 91-99.
41. Canziani A., Paszke A., Culurciello E. An analysis of deep neural network models for practical applications. — 2016. — arXiv:1605.07678.
42. The cityscapes dataset for semantic urban scene understanding / M. Cordts [et al.] // Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). — 2016. — P. 3213-3223. — https://doi.org/10.1109/CVPR.2016.350
43. Learning to refine object segments / P. O. Pinheiro [et al.] // Proc. Eur. Conf. Comput. Vis. — 2016. — P. 75-91. — https://doi.org/10.1007/978-3-319-46448-0_5

44. Pinheiro P. O., Collobert R., Dollár P. Learning to segment object candidates // Proc. Adv. Neural Inf. Process. Syst. — 2015. — P. 1990-1998.
45. Mask R-CNN / K. He [et al.] // Proc. IEEE Int. Conf. Comput. Vis. — 2017. — P. 2961-2969. — https://doi.org/10.1109/ICCV.2017.322
46. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs / L.-C. Chen [et al.] // IEEE Trans. Pattern Anal. Mach. Intell. — 2018. — Vol. 40. — No. 4. — P. 834-848. — https://doi.org/10.1109/TPAMI.2017.2699184
47. Chen L.-C., Papandreou G., Schroff F., Adam H. Rethinking atrous convolution for semantic image segmentation. — 2017. — arXiv:1706.05587.
48. The warning algorithm and some key technologies of vehicle collision avoidance system / D. Hou [et al.] // Automot. Eng. — 2002. — Vol. 24. — No. 5. — P. 438-444.
49. Front vehicle detection with multi-source information based on deep belief network / W. Zhangu [et al.] // Automot. Eng. — 2018. — Vol. 40. — No. 5. — P. 554-560.
50. Cai X., Sarabandi K. A machine learning based 77 GHz radar target classification for autonomous vehicles // Proc. IEEE Int. Symp. Antennas Propag. USNC-URSI Radio Sci. Meeting. — 2019. — P. 371-372. — https://doi.org/10.1109/APUSNCURSINRSM.2019.8888647
51. Ramachandra K. V. Kalman Filtering Techniques for Radar Tracking. — 2018. — https://doi.org/10.1201/9781482273113
52. Hu B., Zhao C. Vehicle detection method based on MHT model using millimeter-wave radar // J. Nanjing Univ. Sci. Technol. — 2008. — No 3. — P. 450-455.
53. Liu Z., Wang J., Li K. Robust vehicular radar target determination // J. Tsinghua Univ. (Sci. Technol.). — 2008. — Vol. 48. — No. 5. — P. 875-878.
54. Point cloud features-based kernel SVM for human vehicle classification in millimeter wave radar / Z. Zhao [et al.] // IEEE Access. — 2020. — Vol. 8. — P. 26012-26021. — https://doi.org/10.1109/ACCESS.2020.2970533
55. Through fog high-resolution imaging using millimeter wave radar / J. Guan [et al.] // Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR). — 2020. — P. 11464-11473. — https://doi.org/10.1109/CVPR42600.2020.01148
56. YOLO-based simultaneous target detection and classification in automotive FMCW radar systems / W. Kim [et al.] // Sensors. — 2020. — Vol. 20. — No. 10. — P. 2897. — https://doi.org/10.3390/s20102897
57. Wang Z., Du L., Mao J., Liu B., Yang D. SAR target detection based on SSD with data augmentation and transfer learning // IEEE Geosci. Remote Sens. Lett. — 2018. — Vol. 16. — No. 1. — P. 150-154. — https://doi.org/10.1109/LGRS.2018.2867242
58. Ding J., Chen B., Liu H., Huang M. Convolutional neural network with data augmentation for SAR target recognition // IEEE Geosci. Remote Sens. Lett.

— 2016. — Vol. 13. — No. 3. — P. 364-368. — https://doi.org/10.1109/LGRS.2015.2513754

59. Huang Z., Pan Z., Lei B. Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data // Remote Sens. — 2017. — Vol. 9. — No. 9. — P. 907. — https://doi.org/10.3390/rs9090907
60. Stanley: The robot that won the DARPA grand challenge / S. Thrun [et al.] // J. Field Robot. — 2006. — Vol. 23. — No. 9. — P. 661-692. — https://doi.org/10.1002/rob.20147
61. Junior: The Stanford entry in the urban challenge / M. Montemerlo [et al.] // J. Field Robot. — 2008. — Vol. 25. — No. 9. — P. 569-597. — https://doi.org/10.1002/rob.20258
62. Petrovskaya A., Thrun S. Model based vehicle detection and tracking for autonomous urban driving // Auton. Robots. — 2009. — Vol. 26. — No. 2-3. — P. 123-139. — https://doi.org/10.1007/s10514-009-9115-1
63. Disocclusion of 3D LiDAR point clouds using range images / P. Biasutti [et al.] // ISPRS International Society for Photogrammetry and Remote Sensing. — 2017. — https://doi.org/10.5194/isprs-annals-IV-1-W1-75-2017
64. Range image-based density-based spatial clustering of application with noise clustering method of three-dimensional point clouds / M. Wen [et al.] // Int. J. Adv. Robotic Syst. — 2018. — Vol. 15. — No. 2. — Art. no. 172988141876230. — https://doi.org/10.1177/1729881418762302
65. Hybrid elevation maps: 3D surface models for segmentation / B. Douillard [et al.] // Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. — 2010. — P. 1532-1538. — https://doi.org/10.1109/IROS.2010.5650541
66. Bogoslavskyi I., Stachniss C. Efficient online segmentation for sparse 3D laser scans // PFG-J. Photogramm. Remote Sens. Geoinf. Sci. — 2017. — Vol. 85. — No. 1. — P. 41-52. — https://doi.org/10.1007/s41064-016-0003-y
67. Zhang X., Wanggen W., Li X., Junxing M. Mean shift clustering segmentation and RANSAC simplification of color point cloud // Proc. Int. Conf. Audio, Lang. Image Process. — 2015. — P. 837-841. — https://doi.org/10.1109/ICALIP.2014.7009912
68. Byun J. Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle // Field and Service Robotics. — 2015. — P. 49-60. — https://doi.org/10.1007/978-3-319-07488-7_4
69. 3D LIDAR based ground segmentation / C. Tongtong [et al.] // Proc. 1st Asian Conf. Pattern Recognit. — 2011. — P. 446-450. — https://doi.org/10.1109/ACPR.2011.6166587
70. 3D lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes / A. Asvadi [et al.] // Robot. Auton. Syst. — 2016. — Vol. 83. — P. 299-311. — https://doi.org/10.1016/j.robot.2016.06.007
71. Cheng J., Xiang Z., Cao T., Liu J. Robust vehicle detection using 3D lidar under complex urban environment // Proc. IEEE Int. Conf. Robot. Autom.

(ICRA). — 2014. — P. 691-696. — https://doi.org/10.1109/ICRA.2014.6906929

72. Wang D. Z., Posner I., Newman P. What could move? Finding cars, pedestrians and bicyclists in 3D laser data // Proc. IEEE Int. Conf. Robot. Autom. — 2012. — P. 4038-4044. — https://doi.org/10.1109/ICRA.2012.6224734
73. Natural terrain classification using three dimensional ladar data for ground robot mobility / J. F. Lalonde [et al.] // J. Field Robot. — 2006. — Vol. 23. — No. 10. — P. 839-861. — https://doi.org/10.1002/rob.20134
74. Real-time dynamic obstacle detection and tracking using 3D lidar / Y. Fei [et al.] // J. Zhejiang Univ. (Eng. Sci.). — 2012. — Vol. 46. — No. 9. — P. 1565-1571.
75. Li B., Zhang T., Xia T. Vehicle detection from 3D lidar using fully convolutional network. — 2016. — arXiv:1608.07916.
76. Wu B., Wan A., Yue X., Keutzer K. SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud // Proc. IEEE Int. Conf. Robot. Autom. (ICRA). — 2018. — P. 1887-1893. — https://doi.org/10.1109/ICRA.2018.8462926
77. BirdNet: A 3D object detection framework from LiDAR information / J. Beltran [et al.] // Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC). — 2018. — P. 3517-3523. — https://doi.org/10.1109/ITSC.2018.8569311
78. Zhou Y., Tuzel O. VoxelNet: End-to-end learning for point cloud based 3D object detection // Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. — 2018. — P. 4490-4499. — https://doi.org/10.1109/CVPR.2018.00472
79. Ye M., Xu S., Cao T. HVNet: Hybrid voxel network for LiDAR based 3D object detection // Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR). — 2020. — P. 1631-1640. — https://doi.org/10.1109/CVPR42600.2020.00170
80. PointNet: Deep learning on point sets for 3D classification and segmentation / R. Q. Charles [et al.] // Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). — 2017. — P. 652-660. — https://doi.org/10.1109/CVPR.2017.16
81. Qi C. R., Yi L., Su H., Guibas L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space // Proc. Adv. Neural Inf. Process. Syst. — 2017. — P. 5099-5108.
82. Radar Sensor Model for the Virtual Development Process / S. Bernsteiner [et al.] // ATZelektronik Worldwide. — 2015. — Vol. 10. — No. 2. — P. 46-52. — https://doi.org/10.1007/s38314-015-0521-1
83. Cao P., Wachenfeld W., Winner H. Perception Sensor Modeling for Virtual Validation of Automated Driving // IT-Information Technology. — 2015. — Vol. 57. — No. 4. — P. 243-251. — https://doi.org/10.1515/itit-2015-0006
84. Hanke T., Schaermann A., Geiger M. Generation and Validation of Virtual Point Cloud Data for Automated Driving Systems // Proc. IEEE 20th

International Conference on Intelligent Transportation Systems (ITSC). — 2017. — P. 1-6. — https://doi.org/10.1109/ITSC.2017.8317864

85. Charroud A., El Moutaouakil K., Palade V. Localization and Mapping for Self-Driving Vehicles: A Survey // Machines. — 2024. — Vol. 12. — No. 2. — P. 118. — https://doi.org/10.3390/machines12020118
86. Waldrop M. M. Autonomous vehicles: No drivers required // Nature News. — 2015. — Vol. 518. — No. 7537. — Art. no. 20. — https://doi.org/10.1038/518020a
87. Bampis L., Amanatiadis A., Gasteratos A. Encoding the description of image sequences: A two-layered pipeline for loop closure detection // Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. — 2016. — P. 4530-4536. — https://doi.org/10.1109/IROS.2016.7759667
88. Bampis L., Amanatiadis A., Gasteratos A. Fast loop-closure detection using visual-word-vectors from image sequences // Int. J. Robot. Res. — 2018. — Vol. 37. — No. 1. — P. 62-82. — https://doi.org/10.1177/0278364917740639
89. Poggenhans F. Lanelet2: A high-definition map framework for the future of automated driving // Proc. Int. Conf. Intell. Transp. Syst. — 2018. — P. 1672-1679. — https://doi.org/10.1109/ITSC.2018.8569929
90. Dupuis M., Strobl M., Grezlikowski H. Opendrive 2010 and beyond-status and future of the de facto standard for the description of road networks // Proc. of the Driving Simulation Conference Europe. — 2010. — P. 231-242.
91. Reda M., Onsy A., Haikal A. Y. Path planning algorithms in the autonomous driving system: A comprehensive review // Robotics and Autonomous Systems. — 2024. — Vol. 174. — P. 104630. — https://doi.org/10.1016/j.robot.2024.104630
92. Yurtsever E., Takeda K., Miyajima C. Traffic trajectory history and drive path generation using GPS data cloud // Proc. IEEE Intelligent Vehicles Symposium (IV). — 2015. — P. 229-234. — https://doi.org/10.1109/IVS.2015.7225691
93. Martinez C. M., Heucke M., Wang F. Y. Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey // IEEE Transactions on Intelligent Transportation Systems. — 2017. — Vol. 19. — No. 3. — P. 666-676. — https://doi.org/10.1109/TITS.2017.2706978

Contents

Introduction 3
Perception in the vehicle Autopilot system.. 4
Planning and Decision-Making in the vehicle Autopilot system................................ 14
Path Tracking and Control in the vehicle Autopilot system 19
Conclusion 21
References 21