Advancing Connected Autonomous Vehicles: Trajectory Planning, Object Detection, and the Promise of 6G Networks

Aydan Namdar Ghazani

[1] Technical University, Vienna, Austria
[2] Distributed Systems Group

Abstract. Perceiving and sharing environmental data among Connected and Autonomous Vehicles (CAVs) in real-time is essential for ensuring safe and efficient autonomous driving. This paper synthesizes findings from three key studies to address critical challenges in CAV systems, focusing on trajectory planning, object detection, and the potential of 6G networks. By combining these works, the paper offers a cohesive perspective on leveraging innovative technologies to optimize real-time decision-making and collaborative vehicle behavior.
The first study explores dynamic shared maps, enabled by Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, to improve trajectory planning. Experimental results highlight enhanced safety in scenarios such as lane changes and emergency braking through efficient data sharing. The second study investigates image compression techniques for object detection, identifying optimal trade-offs between latency and accuracy. Moderate compression settings, such as H.265, balance reduced data size with reliable detection, making them suitable for both edge and cloud-based platforms.
The third study examines the transformative role of 6G networks, with capabilities such as sub-millisecond latency and peak data rates of 1 Tb/s. These advancements enable precise positioning, real-time data sharing, and edge intelligence, addressing limitations of current 5G systems. By integrating insights from these studies, this paper presents a comprehensive framework for improving CAV capabilities and highlighting future research directions.

Keywords: Connected and autonomous vehicles · 6G Networks · Real-Time Map Updates · Trajectory planning · Object detection · Image Compression · Edge Computing, · Cloud Computing

1 Introduction

Connected Autonomous Vehicles (CAVs) represent a groundbreaking advancement in urban mobility. These vehicles leverage advanced communication networks to interact with each other and their environment, allowing them to navigate more efficiently. However, as cities continue to grow and become more densely populated, the challenge of ensuring a reliable communication between vehicles and infrastructure becomes more critical.

A key aspect of this challenge is trajectory planning, which focuses on optimizing paths to reduce travel times and alleviate congestion. While considerable progress has been made in this area, achieving globally optimal trajectories in real-time across a network of interconnected vehicles remains a complex task. This complexity arises from the dynamic nature of traffic conditions, the computational demands of real-time decision-making, and the limitations of existing communication networks and infrastructure.

Object detection plays a pivotal role in ensuring the safety and effectiveness of connected and autonomous vehicles. By enabling vehicles to identify and classify objects in their surroundings, such as pedestrians, other vehicles, and obstacles, real-time object detection enhances decision-making capabilities. However, this task is computationally intensive and must be performed swiftly to ensure timely reactions in dynamic environments. Balancing the trade-off between high accuracy and low latency remains a key challenge, especially as CAVs are expected to operate in complex and congested traffic conditions. The integration of edge computing and cloud offloading offers potential solutions to manage the computational demands of object detection, yet it introduces additional challenges related to communication latency and reliability.

In addition to these challenges, the ability of CAVs to communicate directly with each other offers a unique advantage, significantly reducing the reliance on infrastructure. This proximity-based communication enables faster and more efficient exchange of critical information, enhancing their robustness in addressing safety-critical problems. For instance, accurate assessment of traffic situations becomes crucial when navigating congested areas, interacting with emergency vehicles, or responding to accidents on the road. This resilience stems, in part, from the aggregation of information shared by all vehicles in the network. Such swarm-like collaboration has demonstrated strong efficiency in solving complex problems in traffic automation, enabling CAVs to work together towards safer and more efficient outcomes in dynamic environments [9].

1.1 Motivation

The motivation for this paper arises from the need to address these challenges through innovative approaches to communication and computation in CAV systems. With the advent of emerging technologies such as Vehicle-to-Infrastructure (V2I) communication and advancements in edge and cloud computing, new possibilities for real-time collaboration and decision-making among vehicles are being unlocked.

This paper investigates three critical components of this ecosystem:

- **Trajectory Planning:** Using a publish–subscribe messaging pattern allows vehicles to exchange dynamic environmental data in real time.
- **Image Compression:** Investigating the role of image compression in object detection to cloud or edge devices.

– **Network:** With the development of 6G, new possibilities arise for better connectivity between vehicles.

1.2 Objective

The objective of this paper is to analyze the critical aspects of communication and real-time computing in Connected Autonomous Vehicle (CAV) systems, with a particular focus on dynamic shared maps, image compression for object detection. Additionally, this paper explores shortly how 6G networks can address current communication and computational challenges, unlocking new possibilities for real-time collaboration and significantly advancing CAV capabilities in traffic management and safety.

1.3 Methodology

This paper conducts a literature review to analyze the approaches to traffic efficiency and path optimization in the context of Connected Autonomous Vehicles. The goal is to synthesize existing research, critically assess current solutions, and identify key challenges and potential advancements in the domains of trajectory planning, object detection, and communication technologies.

The methodology follows these steps:

– **Literature Collection:** A comprehensive search was conducted across scholarly databases to gather research articles, conference papers, and technical reports related to CAVs. The focus was on studies addressing trajectory planning, traffic efficiency, and real-time object detection, along with the role of communication technologies such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I).
– **Synthesis of Findings:** The gathered literature was analyzed to identify the key approaches for optimizing vehicle trajectories and improving traffic efficiency.

1.4 Outline of the Paper

This paper is organized as follows. In the next Section we will have a brief overview about the background of this topic. In Section 3, we will have a closer look into the novel dynamic shared maps approach by M. Szántó et al. Section 4 will be about object detection leveraging image compression for achieving real-time navigation. Section 5 will be an analysis of the results of the previous two sections and a short discussion about combining the approaches and looking into the new possibilities of the 6G network. Finally, in Section 6 this work will be concluded with a summary of the findings, implications for future work and recommendations for advancing communication and computation in CAV.

2 Background

2.1 Communication Technologies for CAVs

CAV systems rely heavily on robust communication technologies to enable real-time data exchange. In the automotive world, these technologies have been categorized depending on the agents involved: Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Network (V2N) communication. They focus on the development of standardized services and generally make use of specialized hardware. Alongside these developments, vehicles with their complex networks, increasing connectivity, sensors, and actuators can also be considered Internet-of-Things (IoT) entities [9]. However, limitations in bandwidth, latency, and reliability often constrain the effectiveness of these communications.

2.2 Trajectory Planning in CAVs

Trajectory planning is a widely researched topic since the early stages of CAV development. The fundamental aspect of autonomous vehicle operation, involves the optimization of going from a starting position to a final one. Effective trajectory planning requires real-time data and predictive modeling to account for factors such as vehicle movements, traffic signals, and potential obstacles. The most relevant techniques are listed below: [9].

Numerical Optimization techniques aim to minimize or maximize a function while considering constraints. In trajectory planning, they smooth pre-calculated trajectories by incorporating vehicle kinematics. Common approaches include Function Optimization and Model Predictive Methods. Despite their precision, these methods are computationally expensive as they optimize at each motion state and rely on global waypoints. [9].

Graph Based techniques represent the state space as a graph and find optimal paths between the vehicle's current and goal states. These methods discretize the search space using occupancy grid maps. Common algorithms include State Lattice, Elastic Band, and A-star. While effective for trajectory planning, they require substantial memory and computational resources, which reduces efficiency. [9].

Geometry Based techniques interpolate predefined waypoints to create smooth trajectories that respect vehicle kinematics, dynamics, and passenger comfort. Methods like clothoid curves and Bézier curves are popular. Clothoid curves are computationally intensive due to integration processes, whereas Bézier curves are more efficient, as their curvature is controlled by defined points. [9].

2.3 Maps for Automated Vehicle Navigation

Crowdsourced mapping, low-latency map updating, and external perception represent three distinct yet interrelated methodologies aimed at enhancing the situational awareness and operational efficiency of CAVs.

Crowdsourced Maps are constructed through the aggregation of environmental data collected from a network of CAVs, contributing to a continuously evolving representation of road conditions and obstacles [4]. This decentralized approach leverages the collective sensor data from multiple agents, enabling the generation of detailed, high-frequency updates. The most popular crowdsourced map platoform right now is OpenStreetMap(OSM) [1]. The primary advantage lies in scalability and the ability to capture a wide geographical area in near real-time. However, the accuracy and reliability of crowdsourced maps are contingent upon the density of contributing vehicles and the quality of sensor data.

Low-Latency Maps focus on minimizing the delay between obstacle detection and map dissemination. This methodology emphasizes the rapid integration of new data into pre-existing maps, ensuring that CAVs receive timely updates to adjust their trajectories accordingly [9]. While low-latency updates enhance responsiveness, they are limited by the infrastructure's communication capacity and the processing efficiency of map management modules.

External Perception extends the capabilities of CAVs by integrating sensor data from external sources, such as infrastructure-mounted cameras and LIDARs [7]. This approach addresses occlusion and sensor range limitations, offering a supplementary data stream that enriches onboard perception systems. The main challenge lies in the seamless fusion of externally perceived data with internal sensor outputs, necessitating advanced data fusion algorithms and synchronization mechanisms.

2.4 Object Detection

Early methods of object detection relied on hand-crafted feature extraction techniques, such as the Viola–Jones face detector using Haar-like features [2]. In recent years, deep learning has introduced two main categories of object detectors: two-stage and single-stage models. Two-stage detectors first generate rough object proposals by identifying regions of interest, which are then refined by a trained model [2]. While these models are accurate, their computational demands make real-time implementation challenging. To overcome this, single-stage models like YOLO [8] have integrated the proposal and refinement steps into a single, more efficient process. For instance, YOLO divides input frames into cells and predicts multiple bounding boxes per cell. The YOLO architecture can be trained end-to-end with a loss function that incorporates bounding box accuracy, object detection confidence (objectness), and class probabilities.

3 Real-Time Map Updates

3.1 Automated Driving Framework

The automated driving (AD) framework used in this study is based on a six-block architecture, as shown in Figure 1, and comprises the following components:

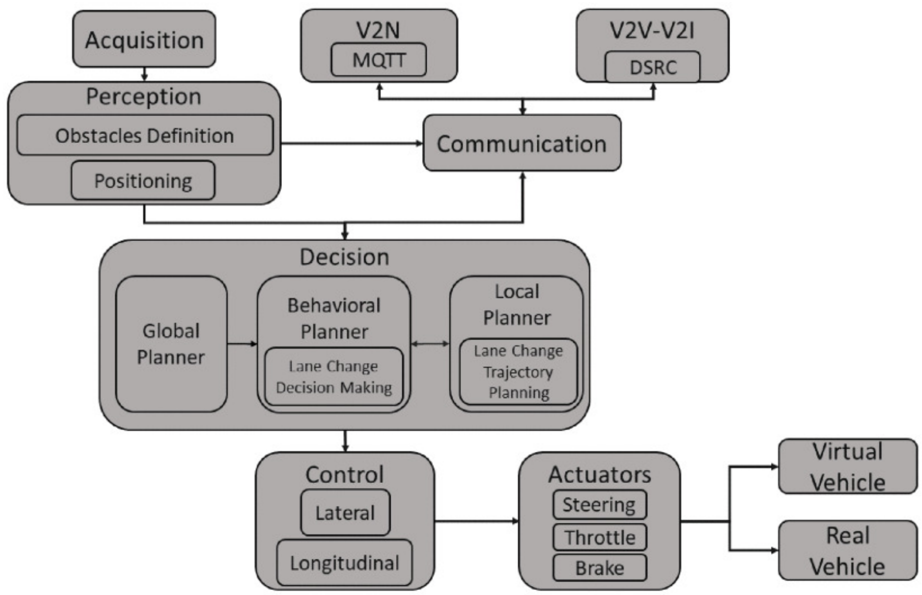

Fig. 1. The automated driving framework (ADF) by M. Szántó et al.

- **Acquisition:** This module collects raw data from various sensors, including LIDAR, cameras, GNSS, and laser sensors. The gathered data is passed to the perception module for further processing.
- **Perception:** This module interprets the sensor data to build an environmental representation, including the identification of surrounding obstacles and the vehicle's location.
- **Communication:** Responsible for Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Network (V2N) communication, this module enables the exchange of information between vehicles, infrastructure, and cloud-based systems.
- **Decision:** This module uses information from the perception and communication modules to generate trajectories. It is subdivided into three blocks:
  - **Global Planner:** Creates an initial trajectory using a predefined OpenStreetMap (OSM) representation and applies an A-star algorithm to find the optimal path.

    - **Behavioral Planner:** Implements a finite state machine (FSM) to manage maneuver decisions, including lane changes, based on environmental conditions and user inputs.
    - **Local Planner:** Generates smooth and safe trajectories using Bézier curve techniques, ensuring continuity and feasibility for the vehicle controllers.
- **Control:** Converts the generated trajectories into actionable throttle, brake, and steering commands. This block is divided into:
    - **Longitudinal Control:** Manages speed through throttle and brake signals.
    - **Lateral Control:** Adjusts steering to follow the planned trajectory.
- **Actuation:** Executes the control signals through physical or simulated vehicle actuators.

Lane Change Development The lane change maneuver in this framework is handled in two stages:

1. The **Behavioral Planner FSM** manages state transitions:
    - Transition to *Lane Change* state occurs if:
        (a) An obstacle is detected in the vehicle's path.
        (b) A user initiates a lane change.
        (c) The current lane ends or is closed.
    - Transition back to *Keep Lane* state occurs if:
        (a) The lane change is completed.
        (b) The lane change is aborted.
2. The **Local Planner** generates a smooth trajectory for the lane change using Bézier curves, following the methodology proposed by Lattarulo et al. [5].

3.2 Mapping Framework

The aim of M.Szántó et al. [9] mapping framework is to supply dynamically updated maps through the MQTT network: All vehicles as well as the map management module are connected to the MQTT broker. Individual CAVs publish pre-processed observation data obtained by their LIDAR sensors over designated MQTT topics. The map management module subscribes to these topics and receives observations from the vehicles. The mapping framework employs a novel candidate/employed map (C/EM) scheme. Two distinct maps are maintained:

- **Candidate Map (M_c):** A private map used to accumulate new observations. Items in this map are promoted to the employed map after surpassing a defined observation threshold.
- **Employed Map (M_e):** A public map containing validated items that are broadcast to connected CAVs.

The module follows a state machine architecture comprising initialization, idle, update map, broadcast map, and exit states. During initialization, the maps are empty. When observations are received, the system decodes and processes the data, matching it to existing map elements using Gaussian Radial Basis Functions (RBF) and the Hungarian algorithm for assignment.

Items that cannot be matched are added to the candidate map. When an item in the candidate map reaches a specified observation threshold, it is promoted to the employed map and broadcast. This process ensures that dynamic obstacles are accurately reflected in the map, facilitating real-time updates crucial for trajectory planning.

3.3 Communication Scheme

The communication scheme between individual CAVs and the map management module is established through the MQTT protocol. This lightweight, low-latency protocol enables efficient data sharing crucial for real-time map updates [6]. Figure 2 illustrates the system setup.

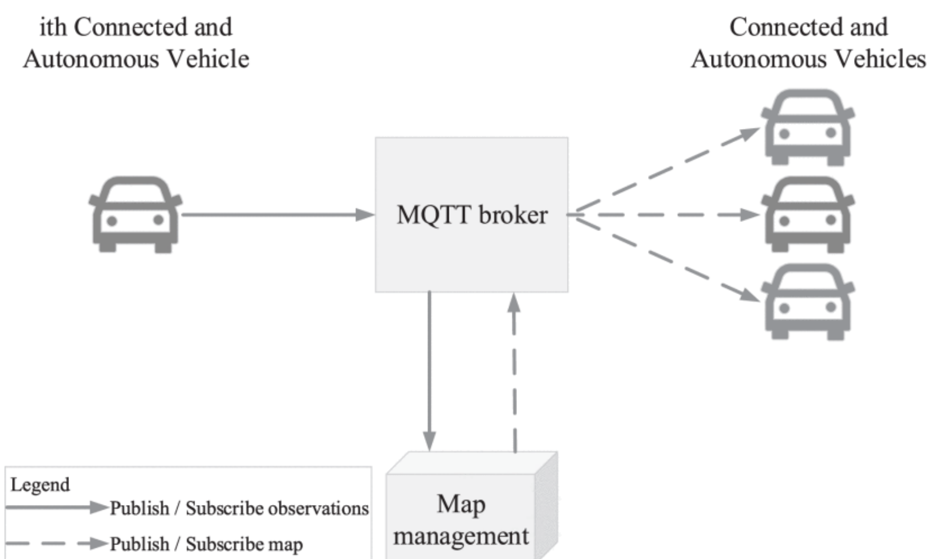

Fig. 2. Communication schema: Each CAV is connected to the MQTT broker as client, publishing and subscribing to the same topic.

– **CAVs as Publishers:** Each CAV publishes pre-processed observation data obtained from its LIDAR sensors to designated MQTT topics. This data includes obstacle attributes such as location, speed, and dimensions.

- **Map Management Module as Subscriber:** The map management module subscribes to these topics to receive and process observations. Using a state machine architecture, the module decodes, matches, and integrates this data into its maps.
- **Broadcasting Map Updates:** Once processed, updated maps are published on separate MQTT topics. All subscribing CAVs instantly receive these updates, ensuring that their internal trajectory planners use the latest environmental information.

4 Image Compression in Object Detection

Real-time object detection in autonomous driving necessitates efficient processing of high volumes of sensor data, such as camera frames, LiDAR, and radar inputs. However, the computational limitations and power constraints of onboard vehicle hardware pose significant challenges to running large detection models locally. To address this, the study in [2] explores the potential of offloading computational tasks to edge and cloud platforms, which provide superior processing capabilities but introduce additional transmission latency.

A core element of this approach is the use of image and video compression techniques to mitigate the data transfer overhead associated with streaming raw sensor data. The authors investigate two primary compression methods—JPEG and H.265—and evaluate their effectiveness in reducing latency while maintaining detection accuracy. JPEG is a well-established image compression standard that adjusts file size through a quality parameter, while H.265 exploits temporal relationships between frames, offering superior video compression by predicting changes across successive frames [2].

To evaluate the proposed offloading strategies for real-time object detection in autonomous driving, the authors set up a comprehensive testbed combining real hardware and simulation techniques. The key components of the testbed are as follows:

- **Hardware Configurations:** Three distinct hardware platforms were used:
  - *Local:* NVIDIA Jetson Xavier NX with Volta GPU and 384 CUDA cores for onboard detection, operating at a resolution of 640×640.
  - *Edge:* A laptop with a GeForce GTX 1650 GPU, 896 CUDA cores, and Intel i9-9980HK CPU.
  - *Cloud:* An HPC node with Tesla V100 GPU, 5120 CUDA cores, and Intel Xeon G6132 CPU, operating at a higher resolution of 1280×1280.
- **Dataset:** A synthetic dataset was generated using the CARLA simulator to simulate a camera-equipped car in urban environments. The dataset comprises $10,000$ frames captured at 1 Hz, annotated with ground truth bounding boxes for vehicles, pedestrians, and traffic lights. Frames were split into training, validation, and test sets in a $60\% : 20\% : 20\%$ ratio.
- **Compression Techniques:** Both JPEG and H.265 compression techniques were evaluated at multiple quality levels. For JPEG, quality values of 100

(high), 80 (medium), 30 (low), and 10 (very low) were analyzed. H.265 compression explored parameters like Constant Rate Factor (CRF), presets, and the use of I-frames and P-frames to balance quality and latency.
- **Networking Simulation:** The authors utilized Simu5G, an OMNeT++-based simulation framework, to estimate end-to-end latency. The testbed included a 5G Radio Access Network (RAN) and a Multi-access Edge Computing (MEC) host located 500 m from the base station. Network latency metrics, such as throughput and packet loss, were recorded for different scenarios.
- **Training Protocol:** YOLOv5 models of varying sizes were trained for 100 epochs using the Adam optimizer with an initial learning rate of 0.001. A batch size that maximized GPU memory utilization was selected for each experiment.

5 Results and Discussions

5.1 Real-Time Map Updates

The map management module was tested using simulated and real environments. In initial simulations, a single obstacle was broadcast at 3.33 Hz. The system exhibited a trade-off between mapping delay and localization jitter, influenced by the promotion threshold of from candidate to employed map. A threshold value of 8 provided optimal performance, balancing fast obstacle promotion with stable localization [9].

Experimental validation was conducted at the Tecnalia Test Track, involving a Renault Twizy 80 equipped with LIDAR, GNSS, and actuators. The Twizy detected static obstacles and broadcast data via MQTT. A virtual CAV received this data and executed corresponding maneuvers.

Two scenarios were tested: lane changes and emergency braking. In the lane change scenario, the virtual CAV successfully avoided obstacles by switching lanes with a 16-meter clearance, completing the maneuver in 3 seconds. In the emergency braking scenario, the CAV aborted the lane change upon detecting obstacles in both lanes and executed a controlled stop at 6.3 meters from the obstruction.

These results demonstrate the module's capability to enhance CAV trajectory planning by leveraging real-time external data. The integration of the C/EM framework with MQTT significantly improved maneuver safety and accuracy, underscoring the potential of dynamic map updates in automated driving systems.

5.2 Image Compression in Object Detection

One of the primary findings of F. Hawlader et al. [2] is the significant reduction in data size achieved through compression, directly translating to lower transmission latency. For edge platforms, the use of high-quality JPEG compression

(JPEG-H) reduces frame size by approximately 86%, decreasing latency from 123.2 ms (uncompressed) to 59.48 ms. H.265 compression, which leverages inter-frame prediction, achieves even greater reductions, with low-quality (H.265-L) and very low-quality (H.265-VL) settings resulting in up to 98% smaller data sizes and latencies as low as 37.47 ms.

While compression effectively reduces latency, the study identifies a clear trade-off between data size reduction and detection accuracy. Excessive compression, particularly at very low-quality settings, negatively impacts the performance of object detection models. More prominent objects such as vehicles and traffic lights maintain high detection accuracy across various compression settings. In contrast, pedestrian detection exhibits greater sensitivity to compression, reflecting the smaller pixel footprint of pedestrians within the input frames. For example, pedestrian detection precision drops by over 90% under very low-quality H.265 compression (H.265-VL) on edge platforms, whereas vehicle and traffic light detection remain relatively robust [2].

Moderate compression settings, such as medium-quality JPEG (JPEG-M) and H.265 (H.265-M), achieve a favorable balance. In cloud-based detection, JPEG-M and H.265-M yield mean average precision (mAP) values of 81% and 82%, respectively, with latencies under 50 ms, meeting the 20 Hz constraint. These results indicate that cloud platforms, equipped with high-resolution input and moderate compression, consistently outperform edge-based solutions, offering enhanced detection accuracy without breaching latency requirements [2].

The comparative evaluation of edge and cloud platforms underscores the superior performance of cloud-based detection, particularly for high-resolution inputs. Despite the increased baseline latency associated with cloud offloading, the advanced computational power and larger model capacity available in the cloud result in higher detection precision. In contrast, edge platforms, constrained by lower computational resources, show diminished performance, especially under aggressive compression.

However, the paper highlights that edge-based solutions remain viable for scenarios requiring lower power consumption or when network connectivity to the cloud is limited. For edge platforms, moderate H.265 compression (H.265-M) achieves latency of 41.61 ms with acceptable detection accuracy, making it a feasible alternative in latency-sensitive environments [2].

5.3 Integrating both approaches

Since it is not specified how exactly M. Szántó et al. [9] perception module of the Automated Driving Framework works, an integration of the object detection method of F. Hawlader et al. [2] could be seen as a positive addition to their driving framework. Processing the obstacles faster into the map of surroundings could speed up the lane change or trajectory planning of the moving vehicle. Potentially it could also reduce the jitter with promoting obstacles from candidates to employed obstacles faster and more accurately.

5.4 The Role of 6G in Enhancing Real-Time Map Updates and Image Compression

While 5G primarily focuses on communications between humans and machines, it lacks the optimization needed for advanced connected vehicle applications like object sensing and precise positioning [3]. The integration of 6G technologies with Connected Autonomous Vehicles (CAVs) holds immense potential for enhancing low-latency maps and offloaded object detection.

THz Band allows for reduced antenna sizes and closer separation distances between antenna elements due to its shorter wavelengths. For instance, at 300 GHz, over 200 antenna elements can fit into an area as small as 1 square centimeter. This enables the generation of narrow beams, which are crucial for precision positioning with outdoor location errors of less than one meter. Additionally, these capabilities address the limitations of traditional GPS, which suffers from low precision and degraded performance in urban environments, and high-definition mapping, which struggles to adapt to dynamic changes like road construction. By leveraging THz-based 6G networks, CAVs gain access to a cost-effective, precise, and reliable alternative for positioning and sensing, which is critical for achieving full automation. Despite these advancements, challenges remain in designing THz antennas and transceivers, extending communication range, and managing high mobility in CAV systems.

Edge Intelligence (EI) and Mobile Edge Computing (MEC) play crucial roles in 6G-enabled CAV systems, significantly enhancing their processing and decision-making capabilities. By leveraging recent advancements in deep learning and artificial intelligence (AI), 6G networks enable MEC stations to handle the heavy computational demands of CAV applications, such as high-resolution map updates, real-time environment perception, and autonomous driving decision-making.

CAVs can offload sensor data to nearby MEC stations over high-speed 6G links, where it is processed and aggregated with data from other CAVs. This aggregation allows MEC stations to produce a comprehensive understanding of the environment, enabling real-time decisions such as detecting obstacles or identifying hazards on the road. For example, MEC stations can not only return critical processed data to individual vehicles but also share insights across the network, improving cooperative sensing and enabling features like coordinated vehicle maneuvers.

In addition to these capabilities, distributed AI and federated learning frameworks are supported by 6G MEC. These frameworks enable collaborative learning and model training between CAVs and MEC stations without transferring raw data to central servers, thereby preserving user privacy and ensuring compliance with data security standards. This distributed approach is particularly important for building high-definition maps for navigation and training AI models for better driving decisions.

Edge Driving and Digital Twins Beyond these core applications, 6G networks enable advanced use cases such as edge driving and digital twins. Edge driving allows machine drivers at MEC stations or nearby CAVs to control vehicles, further reducing the computational burden on individual vehicles. Digital twins of CAVs, which replicate sensor data and create predictive models, can provide enhanced insights into vehicle behavior and road conditions, optimizing traffic flow and safety.

6 Conclusion

6.1 Real-Time Map Updates

M. Szántó et al. [9] presented a map management module which allows CAVs to communicate via MQTT. Their solution effectively introduces low-latency updates regarding externally detected obstacles, ensuring seamless information sharing among CAVs. By implementing an innovative candidate/employed mapping approach, they demonstrated that obstacles identified by a physical CAV on road segments can be rapidly registered and shared with other vehicles.

The capabilities of the candidate/employed mapping were validated through two specific maneuvers. First, a successful and safe lane change maneuver, maintaining a clearance of no less than 16 meters from obstacles and completing the lane change within 3 seconds. Secondly, a cases where the lane change was not feasible due to a blocked lane, the maneuver was safely aborted, and the vehicle came to a stop at a secure distance of 6.3 meters from all obstacles.

The results confirm that their approach, leveraging data from external sensors of other CAVs, enables reliable and timely decision-making, contributing to enhanced safety and coordination in dynamic traffic environments. Future work is planned towards more complex and realistic scenarios. Either by increasing the number of agents participating or testing complex maneuvers.

6.2 Image Compression in Object Detection

The findings of F. Hawlader et al.[2] demonstrate that the judicious use of image compression can unlock significant performance gains in autonomous driving perception systems. H.265 compression, in particular, emerges as a robust solution for reducing data transmission overhead, ensuring that object detection tasks can be efficiently offloaded to edge and cloud platforms, while preserving detection quality. JPEG has also shown to be sufficient in cases where 10 Hz is an acceptable latency. However, the results emphasize the importance of selecting compression settings that preserve detection accuracy, especially for critical classes such as pedestrians.

The paper concludes that cloud platforms, combined with medium-quality H.265 compression, present the most effective solution for balancing detection accuracy and latency and that future work needs to be done on mode switching to local perception of vehicles and the scalability of the system.

6.3 6G Networks

6G's ability to provide peak data rates of up to 1 Tb/s and radio latency as low as 0.1 ms makes it a strong enabler for real-time data sharing in CAV networks. By leveraging 6G, CAV systems can transmit high-resolution visual data to cloud or edge platforms, where advanced processing capabilities can be used to meet the stringent reliability and latency requirements of CAV mission-critical services, redefining the modern transportation system.

Acknowledgments

The authors thank the Distributed Systems Group at Technical University, Vienna, for their support.

References

1. Haklay, M., Weber, P.: Openstreetmap: User-generated street maps. IEEE Pervasive computing **7**(4), 12–18 (2008)
2. Hawlader, F., Robinet, F., Frank, R.: Leveraging the edge and cloud for v2x-based real-time object detection in autonomous driving. Computer Communications **213**, 372–381 (2024)
3. He, J., Yang, K., Chen, H.H.: 6g cellular networks and connected autonomous vehicles. IEEE network **35**(4), 255–261 (2020)
4. Howe, J.: The rise of crowdsourcing, wired. http://www. wired. com/wired/archive/14.06/crowds. html (2006)
5. Lattarulo, R., González, L., Martí, E., Matute, J., Marcano, M., Pérez, J.: Urban motion planning framework based on n-bézier curves considering comfort and safety. Journal of Advanced Transportation **2018**(1), 6060924 (2018)
6. Mishra, B., Kertesz, A.: The use of mqtt in m2m and iot systems: A survey. Ieee Access **8**, 201071–201086 (2020)
7. Müller, J., Strohbeck, J., Herrmann, M., Buchholz, M.: Motion planning for connected automated vehicles at occluded intersections with infrastructure sensors. IEEE Transactions on Intelligent Transportation Systems **23**(10), 17479–17490 (2022)
8. Redmon, J.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
9. Szántó, M., Hidalgo, C., González, L., Rastelli, J.P., Asua, E., Vajta, L.: Trajectory planning of automated vehicles using real-time map updates. IEEE Access **11**, 67468–67481 (2023)