Enhancing Campus Mobility

Achievements and Challenges of the Snow Lion Autonomous Shuttle

By Yingbing Chen, Jie Cheng, Sheng Wang, Hongji Liu, Xiaodong Mei, Xiaoyang Yan, Mingkai Tang, Ge Sun, Ya Wen, Junwei Cai, Xupeng Xie, Lu Gan, Mandan Chao, Ren Xin, Lujia Wang, Ming Liu, and Jianhao Jiao

In recent years, the rapid evolution of autonomous vehicles (AVs) has reshaped global transportation systems, leading to an increase in autonomous shuttle applications in people's daily lives. Leveraging the accomplishments of our earlier endeavor, particularly Hercules [1], an autonomous logistics vehicle for transporting goods, we introduce Snow Lion, an autonomous shuttle vehicle specifically designed to transform on-campus transportation, providing a safe and efficient mobility solution for students, faculty, and visitors.

Although similar research on autonomous shuttles has been presented by [2], [3], and [4], these studies primarily concentrate on consumer-level data collection and analysis [2], the social implications and governance challenges of autonomous shuttle services [3], and pedestrian behavior prediction [4], differing in focus from this work. In addition, this study presents outcomes that differ from our previous work [1] as the campus environment contrasts notably with the structured environments of highways and urban streets. Instead of employing a direct perception-to-planning framework [5], [6], we deploy a pipeline framework for the whole navigation system of the AV as it offers higher traceability of final planning performance, especially when encountering bugs or unexpected deviations during practical implementations. Additionally, it discusses supplementary devices and functionalities to enhance the AV's security, including warning systems for inattentive pedestrians.

Overall, this research (the project website: https://chenyingbing.github.io/xueshi_campus_av/) aims to enhance campus mobility by developing a reliable, efficient, and environmentally friendly autonomous transportation solution tailored to meet the diverse requirements of a university setting. It introduces a prototype autonomous driving system tailored for unstructured environments and delves into the system and algorithmic architecture, as well as the real-world challenges encountered during its development and implementation phases. As depicted in Figure 1, our experiments encompassed a 1,146-kilometer road haul and the transportation of 442 passengers over a two-month period. They provide valuable insights into the intricate process of integrating an AV within campus shuttle operations. Furthermore, a thorough analysis of the lessons derived from this experience furnishes a valuable real-world case study, along with recommendations for future research and development in the field of autonomous driving.

Digital Object Identifier 10.1109/MRA.2024.3433168

1070-9932/24©2024IEEE

SYSTEM INTRODUCTION

As shown in Figure 2, we initially present an overview of the modules and their interconnections within our autonomous shuttle. Furthermore, we explore the contextual functions and onboard sensors and elucidate their operational pipelines.

SENSORS AND DEVICES

The autonomous shuttle, from a hardware perspective, boasts dimensions of $4{,}350 \times 1{,}630 \times 2{,}280$ mm^3 and has the capacity to carry up to six passengers, with a maximum load capacity of 560 kilograms. The vehicle achieves a maximum driving speed of 15 km/h and is equipped with a navigation system that seamlessly integrates multisensor fusion for perception data, enabling circumstance map construction and localization functions. The platform houses an industrial personal computer (IPC) equipped with an Intel i7-8700 CPU (six cores and 12 threads) and 32 GB of memory, along with a 1050Ti NVIDIA graphics card. The vehicle is furnished with a removable 74-V lithium-ion battery that provides power to the chassis, IPC, sensors, and accessories, enabling the vehicle to operate autonomously for 24 h. Additionally, four 16-beam lidar units (HESAI XT16) are present: three are positioned at the front and sides to enhance the detection of surrounding obstacles, while one is installed on the roof primarily for localization purposes. The system also includes four fish-eye cameras utilized by Open Broadcaster Software (OBS) to record video data during operation, an inertial measurement unit, and a high-precision global navigation satellite system (GNSS) providing real-time kinematic capabilities, as well as latitude and longitude positioning information for the AV. Finally, we have a 4G/5G data transfer unit (DTU), six seat-side buttons, a sound player, and a rear LED screen at our disposal to bolster autonomous navigation. The DTU facilitates the connection of the IPC to our cloud management platform over the Internet, allowing for the initiation or suspension of navigation tasks via a mobile application. Additionally, seat-side buttons provide supplementary interaction opportunities for passengers on board during navigation. The sound player serves the dual purpose of broadcasting the vehicle's status and attracting the attention of nearby pedestrians who may be inattentive. Meanwhile, the LED screen is used for display purposes.

FUNCTIONS

The functionalities of the entire autonomous navigation system can be categorized into three primary components: autonomous navigation, task scheduling, and various auxiliary modules, including map editing and sound broadcasting.

Tasks related to autonomous navigation primarily occur within the onboard IPC, involving perception, localization, and planning functions. The perception task interprets point cloud data from lidars, extracting geometric characteristics and velocity details of nearby entities. In contrast, the localization module employs these data for robot positioning and map creation. Once the shapes and speeds of other entities are acquired, their trajectories are estimated using a constant velocity model. These details, along with localization data, are then transmitted to the planning module to calculate a feasible trajectory for the AV along the given global route. Simultaneously, the planning module sends commands to the onboard sound system to enhance other road users' comprehension of the AV's movements. Ultimately, the controller tracks the obtained trajectory and generates a control command for the chassis to execute.

At a remote operational level, two primary tools, the mobile device interface and the web-based interface, are utilized for querying status, recording data, and issuing commands to the autonomous shuttle. Both interfaces offer comparable functions to users, harnessing identical data and services sourced from the cloud. The scheduling server handles task assignments and collects the status of all registered running vehicles. It also performs various functions such as accessing map data for routing, transmitting sensor data to the map server, recording key information in the log server, and providing data replay functionality for traceability. From Figure 2, it is evident that

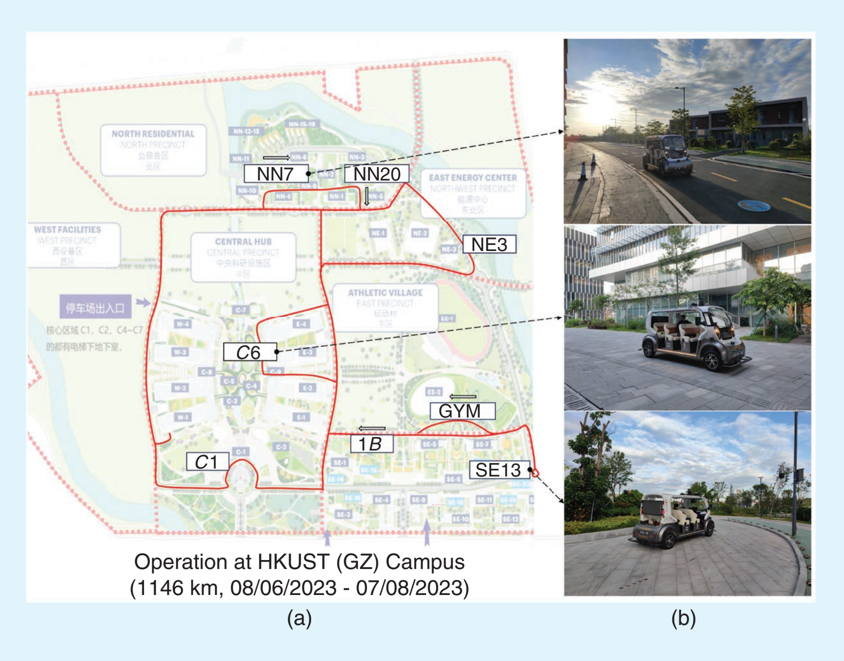


FIGURE 1. (a) The operational scenario of our autonomous shuttle during its service period at the Hong Kong University of Science and Technology (Guangzhou) [HKUST (GZ)]. The red lines represent the operational road map of the campus, with shuttle stations depicted as boxes (e.g., C1 and 1B). (b) Multiple images captured during the operation of the autonomous shuttle.

for task scheduling, three methods are available for assigning tasks to unmanned vehicles. First, developers can communicate with the scheduling server through web sockets. Second, during actual operations, users can utilize private smartphone applications to confirm their desired stops. Last, passengers can also indicate their intent to disembark by pressing an onboard button, prompting the scheduling module to arrange their exit at an appropriate location.

PERCEPTION AND LOCALIZATION

The lidar sensors collect point cloud data used in the perception and localization tasks of AVs. These functions are fundamental to the core of autonomous navigation. The perception function processes sensor data to comprehend the surrounding environment, while the localization function constructs environmental maps and provides location information.

MULTIPLE LIDAR-BASED 3D OBJECT DETECTION

The 3D perception module empowers the AV to detect and precisely locate vital objects, including automobiles, pedestrians, and cyclists, in 3D space via sensor data. The 3D object detection strives to both recognize and categorize objects while also estimating their positions and velocities relative to a designated coordinate system. Geometric data acquired by lidar sensors play a crucial role in perception as precise spatial information greatly enhances the accuracy of 3D object localization.

MULTILIDAR CALIBRATION

Multiple lidar sensors are used for object detection. The initial step involves calibrating the lidar sensors. In this study, we use our marker-based automatic calibration approach [8] that eliminates the need for extra sensors and human involvement. In this approach, it is assumed that three linearly independent planar surfaces, arranged in a wall corner configuration, serve as calibration targets, ensuring that the geometric constraints are adequate for calibrating each pair of lidar sensors. Following the matching of corresponding planar surfaces, our approach effectively retrieves the unknown extrinsic parameters in two stages: initial estimation utilizing the closed-form Kabsch algorithm and subsequent refinement through plane-to-plane iterative closest point (ICP).

OBJECT DETECTION

An illustration of our 3D object detection method is provided in Figure 3. The VoxelNet approach [7] is employed to process multiple point clouds captured by multilidar sensors as inputs. During the input stage, an early-fusion scheme is utilized to combine data from multiple calibrated lidar sensors. Assuming that the lidar sensors are synchronized, we align all the raw point clouds to a common base frame before passing the fused point clouds to the 3D object detector. Then, based on the given point cloud, we partition the 3D space into equidistantly spaced voxels. The point cloud information undergoes a systematic processing procedure, which commences with the mapping of raw point cloud data onto a 3D voxel grid. The

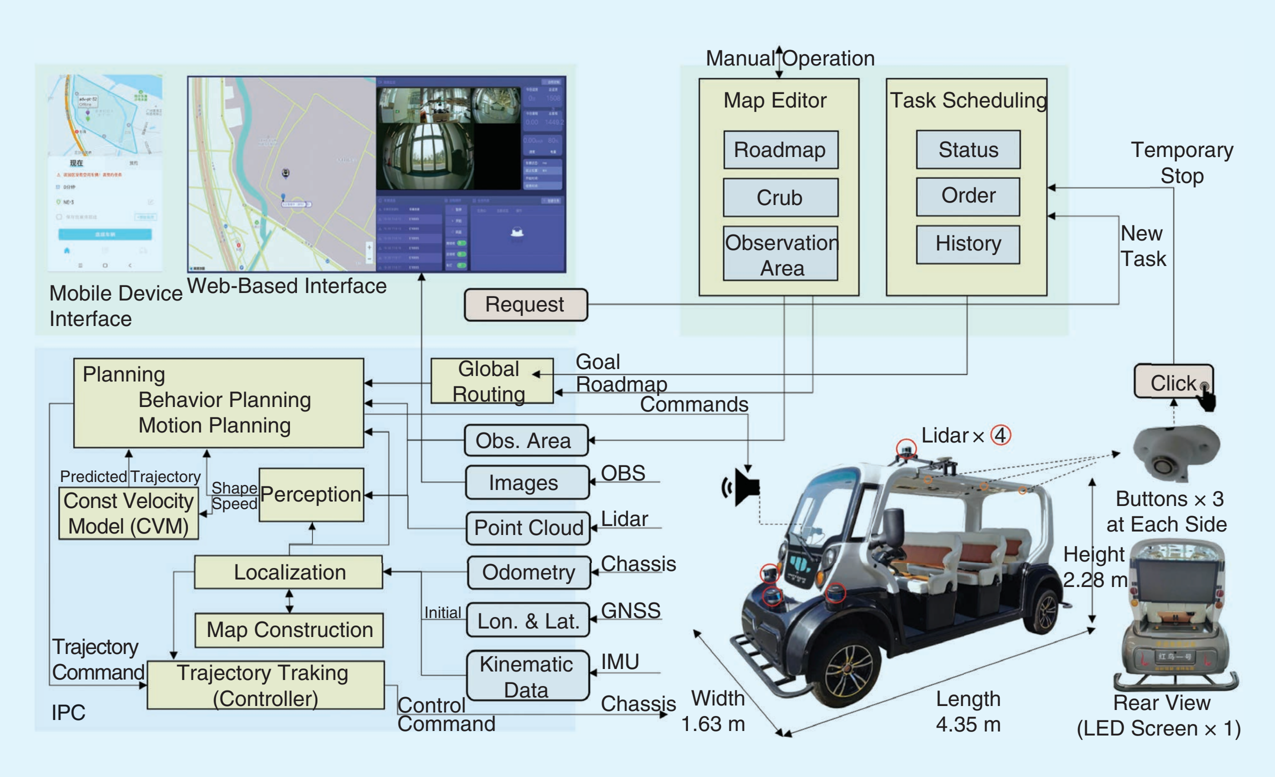


FIGURE 2. The functions and connections of the whole system of the AV. GNSS: global navigation satellite system; IMU: inertial measurement unit; IPC: industrial personal computer; OBS: Open Broadcaster Software.

voxelization step accomplishes two goals: it discretizes the continuous point cloud for computational manageability and addresses the issue of nonuniform point cloud distribution across voxels. For additional computational efficiency optimization, we employ a random subsampling strategy within each voxel, guaranteeing the capture of a representative data subset, thus mitigating the computational overhead linked to processing the complete voxel point cloud.

The following stage, known as the voxel feature encoding (VFE) layer, plays a crucial role in aggregating the key features of individual points within a voxel. The feature representation resulting from the VFE layer encapsulates the overall characteristics of the voxel. Additionally, the feature vectors in the vertical (z) direction are concatenated, effectively creating a bird's-eye view of the current scene. This transformed representation is subsequently utilized for subsequent classification and regression tasks.

At the feature point level, the region proposal network (RPN) header is crucial in predicting offset values for a specific set of anchors. These offsets consist of eight values, including center coordinates, 3D bounding box dimensions, orientation, and velocity. To enhance and optimize predictions while reducing redundancy, a nonmaximum suppression technique is employed to selectively retain predictions with the highest confidence scores. The RPN also plays a key role in classifying the object category associated with the current feature point, thereby enhancing the model's ability for comprehensive scene understanding. The resulting output comprises a series of 3D bounding boxes, each bearing its associated category label, and includes their 2D velocity in the horizontal plane.

THREE-DIMENSIONAL POINT CLOUD MAPPING AND LOCALIZATION

MAPPING

In unmanned vehicle applications, maps play a pivotal role in providing essential information for the localization, perception, and planning tasks of unmanned vehicles. Among various map forms, lidar point cloud maps are preferred due to their density, informativeness, and accuracy. Graph-based optimization techniques are commonly employed for lidar-based mapping. Within a constructed pose graph, edges represent constraints, while nodes correspond to poses. The primary objective of pose graph optimization is to minimize the error associated with all constraints. LeGO-LOAM [9] is a widely recognized graph optimization method known for its lightweight design, ground optimization, back-end enhancements, and loop detection mechanism. To further mitigate accumulation errors, we introduce GNSS measurements as a new constraint within the graph. The three constraints within the graph encompass the odometry constraint derived from lidar odometry, the GNSS constraint determined by the static transformation between the lidar and GNSS antenna, and the loop closure constraint established through the ICP algorithm. The optimization problem is addressed using the Levenberg–Marquardt (LM) algorithm, chosen for its effectiveness compared with the Gauss–Newton algorithm, owing to the LM algorithm's establishment of a trust region for valid nonlinear approximations. Figure 4 depicts a map resulting from the mapping process alongside the corresponding real-world satellite map. The inclusion of GNSS measurements as a new constraint significantly enhances the accuracy and reliability of the optimized map. Our proposed

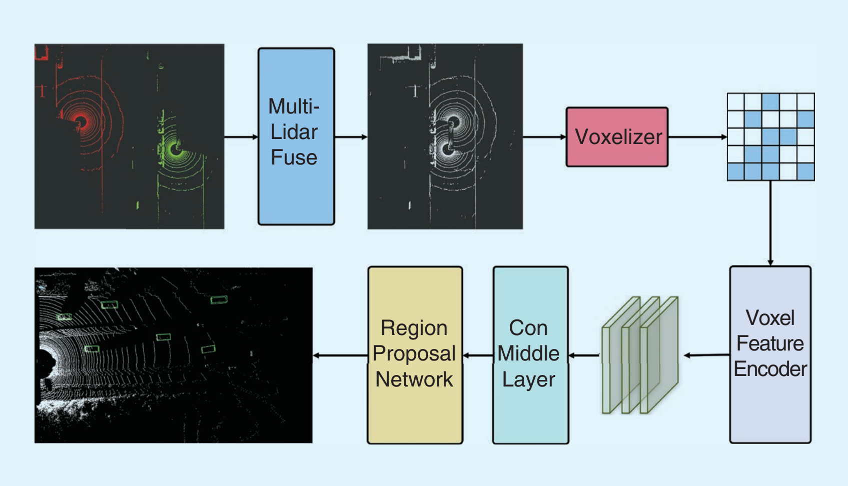


FIGURE 3. The 3D object detection module overview. Utilizing synchronized and well-calibrated lidar-captured point clouds, we employ an early-fusion technique to merge data from multiple calibrated lidars and apply VoxelNet [7] for 3D object detection using the fused results.

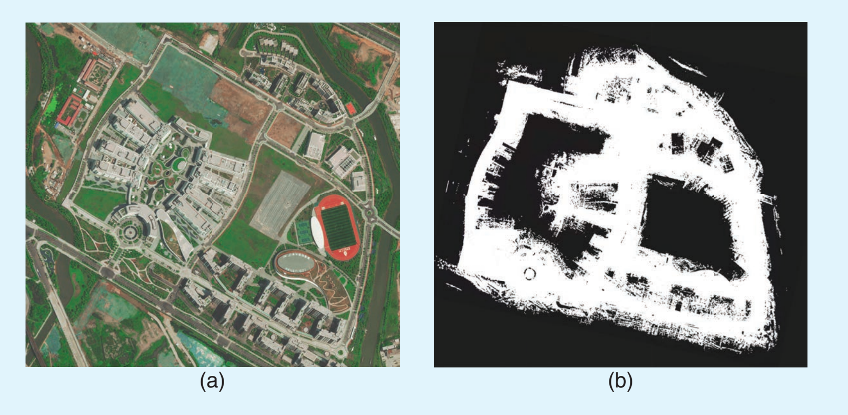


FIGURE 4. (a) A satellite map of the campus and (b) a constructed point cloud map.

approach holds promise for unmanned vehicle applications necessitating highly accurate and dependable maps.

LOCALIZATION

Accurate localization is a critical aspect of navigation and control. Our approach to achieve precision localization involves fusing lidar observation data with odometry data. To relieve the issue of observation data delays, our localization system is designed based on the steady-state approximation of the extended Kalman filter, as opposed to using a pure EKF. Upon acquiring a new frame of lidar data, it is imperative to register it with existing map data. In an effort to enhance computational efficiency and reduce storage demands, our initial step involves downsampling the map created through the method described in the section "Mapping." After acquiring an initial pose estimation using GNSS data, we proceed to extract the point cloud within the region of interest from the map. Subsequently, we employ the ICP algorithm to calculate the precise pose.

PLANNING FRAMEWORK

In a complex campus environment, motion planning for an autonomous campus mobility vehicle must address diverse interactions, involving pedestrians, bicycles, and vehicles, both within and outside traffic-regulated zones. These demands necessitate the AV to possess two crucial capabilities. First, the AV needs to demonstrate robust responses to environmental uncertainties. Second, it must ensure motion that is easily understood by other road users and comfortable for passengers. Figure 5 illustrates our approach, which utilizes a pipeline planner for campus mobility scenarios.

A NAVIGATION TASK WITH GLOBAL ROUTING

Building upon our prior research [1], [10], when provided with the road map data and the AV's initial position, the navigation task manager manages destination data for the AV and invokes the global routing module to determine the global route for autonomous driving. As depicted in Figure 7(a), we apply the A* algorithm to perform this function. Additionally, our designed task manager utilizes the status of the function button to truncate the acquired global route to a temporary destination when passengers request an unscheduled drop-off. After the AV has come to a halt, it remains stationary for a predefined period (typically 8 s) and checks for the absence of nearby pedestrians before resuming the initial navigation task.

BEHAVIORAL PLANNING

Utilizing input data from the local map (representing the drivable area) and obstacle information, our behavioral planning, depicted in Figure 6, computes a local reference route for subsequent motion planning tasks based on scenario tags in the road map. The resulting reference route is depicted as a lateral deviation d_r from the global reference route, using the Frenét framework. The goal of this module is to ensure a comfortable driving experience and define precise driving constraints for diverse scenarios. These scenarios encompass common roads, parking lots, and unprotected intersections. We specify distinct configurations and constraints for each scenario, subsequently integrating them into the motion planning task to ensure safety and stability during vehicle operation.

COMMON SITUATION

In the majority of cases, behavioral planning facilitates lane-changing maneuvers for AVs, allowing the sampling of lateral

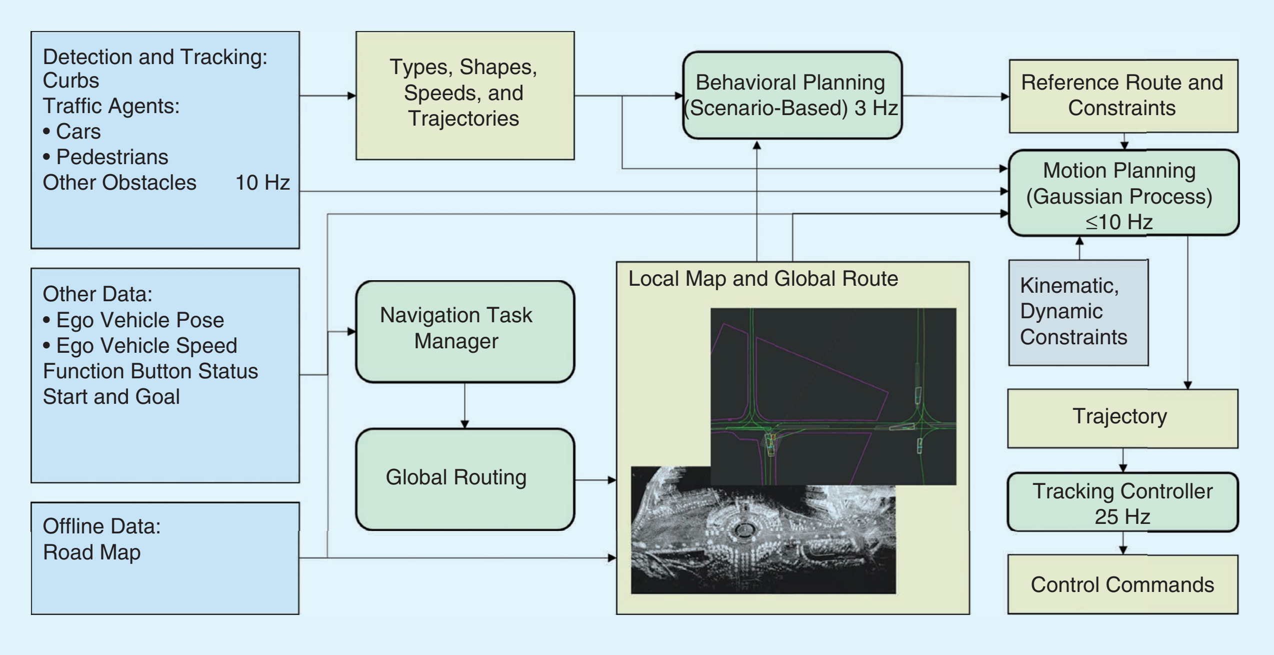


FIGURE 5. The planning pipeline mainly comprises global routing, behavioral planning, motion planning, and a tracking controller.

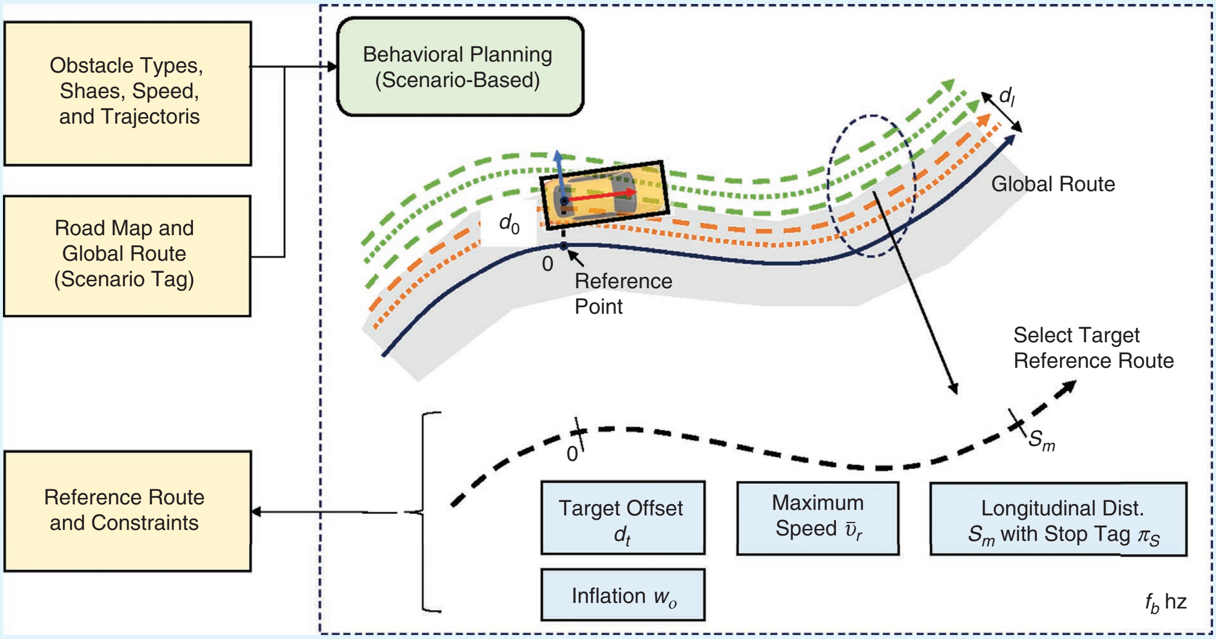

Obstacle Types, Shaes, Speed, and Trajectoris
Road Map and Global Route (Scenario Tag)
Reference Route and Constraints
Behavioral Planning (Scenario-Based)
d_l
d_0
0
Reference Point
Global Route
Select Target Reference Route
0
S_m
Target Offset d_t
Maximum Speed $\bar{v}_r$
Longitudinal Dist. S_m with Stop Tag π_S
Inflation w_o
f_b hz

FIGURE 6. An illustration of the behavioral planning, where d_0 represents the initial lateral offset of the AV (yellow block).

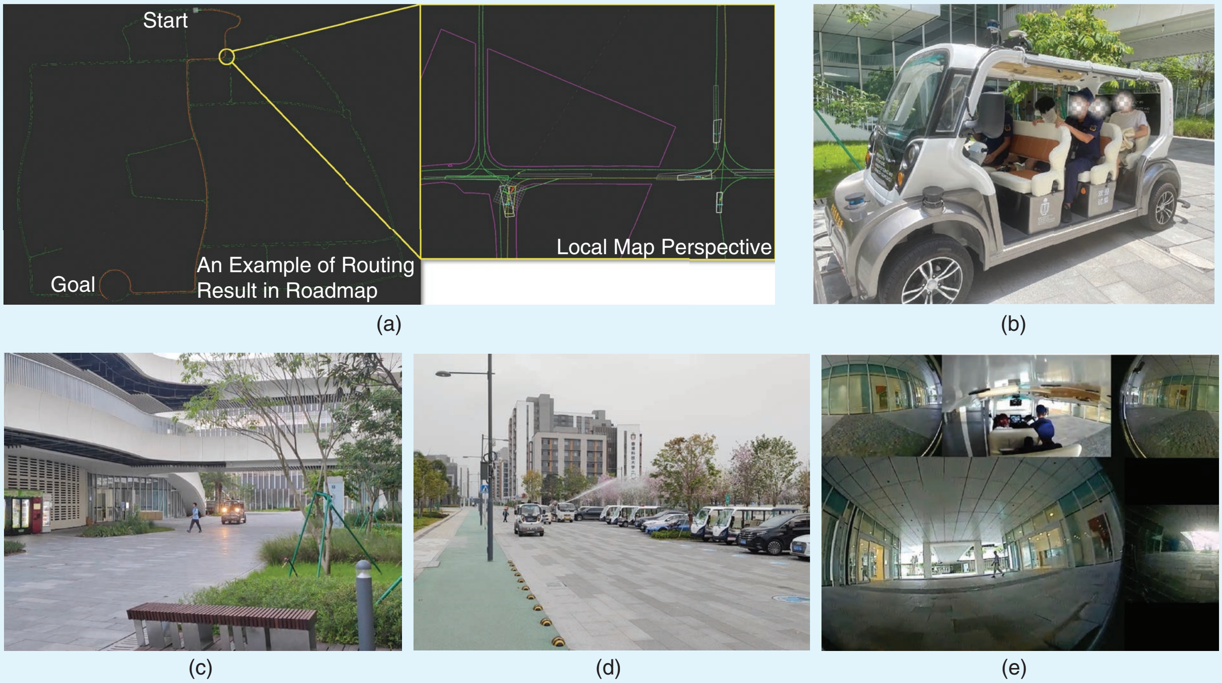


FIGURE 7. Examples of routing outcomes and operational photographs. (a) Examples of routing results and the local map perspective. The red line on the left represents the global route, while the right side displays local environmental representations used in the planning process. Purple areas signify predefined curb areas. Green areas designate the observation area utilized in behavioral planning (see the section "Behavioral Planning"). The yellow block indicates the stop line where the AV yields to dynamic obstacles within the intersection. (b) The autonomous shuttle is stationed at the designated stop, awaiting passengers (two are already seated in the rear row). Two security guards dressed in blue are present throughout the entire operation to ensure passenger safety in case of accidents or emergencies. (c) and (d) Photographs captured during the operation period. (e) A screenshot obtained from the OBS during the operation period.

offsets exceeding a specified threshold value d_l. Illustrated in Figure 6, in such instances, the lateral offsets of the reference route are constrained within the green parts of lateral offsets, and a time delay $1/f_p$ is implemented in the planning loop to mitigate the occurrence of serpentine vehicle movements, thus ensuring a more comfortable passenger experience.

PARKING LOT

Within parking lot scenarios, the environment becomes notably more uncertain due to the possibility of a parked car unexpectedly entering the AV's path. To ensure safe navigation performance, we employ both the predicted trajectories of other traffic agents and a width inflation operation to enhance driving safety.

UNPROTECTED INTERSECTIONS

During traversal of unprotected intersections, the behavioral planner prioritizes the consideration of dynamic obstacles within the corresponding observation areas [as depicted in Figure 7(a)]. When the AV detects dynamic obstacles within these zones, it performs a lane-stop maneuver, halting at a predefined stop line if their velocities and anticipated paths indicate potential future conflicts. This operation ensures stable driving performance, especially in intricately unprotected intersections, as relying solely on noise-affected trajectory predictions for planning purposes are generally inadequate.

This study utilizes the scenario tag of the global route to match predefined lateral offset options during the behavioral planning stage. Additionally, it determines parameters such as the obstacle width inflation volume w_o and the maximum reference speed $\bar{v}_r$. Details of the configuration and constraint outputs will be elaborated in the section "Experimental Results." The target reference route is derived by computing the cost trajectory among the offset candidates $\mathcal{D}$ as

$$d_t = \underset{d \in \mathcal{D}}{\operatorname{argmin}} (J_s + J_d + J_o + J_{\mathrm{dyn}})$$

where $J_s = w_s \cdot (1.0 - s_m(d)/s_{\max})$ promotes the achievement of an extended uncollided path distance. Here $s_m(d)$ represents the route distance at offset d, while $s_{\max}$ denotes the maximum sample distance. $J_d = w_{d1}\,|\,d - d_0\,| + w_{d2}\,|\,d\,|$ encourages the alignment of the target offset d_t with both the route center and the AV's current offset value d_o. The expression for J_o is given as $w_{o1}/c_{\mathrm{avg}} + w_{o2}/c_{\min}$, where $w_{(\cdot)}$ represents the weights, c_{avg} denotes the average clearance to obstacles along the reference route, and $c_{\min}$ represents the minimum clearance value. Additionally, J_{dyn} is a cost term that assesses potential collisions with dynamic obstacles, incurring a high cost in the presence of collision risks with other agents' trajectories. Furthermore, the ultimate output reference route is labeled with π_s, signifying whether it undergoes truncation as a result of colliding with a static obstacle, at which point the AV must come to a stop at the end of the reference route.

MOTION PLANNING AND CONTROL

Upon acquiring the reference route and constraints from the behavioral planning module, the subsequent objective is to create the trajectory for the AV following the specified reference route.

CONSTRAINTS

Apart from the maximum speed limit $\bar{v}_r$ of the reference route, the motion planning module encompasses various constraints: 1) the speed limit influenced by the path curvature; 2) the speed restriction related to pedestrian clearance, considering the obstacle's width inflation w_o; and 3) the terminal speed limit if a halt is necessary at the end of the reference route. For the speed limit from path curvature, it follows that

$$\bar{v}_\kappa(s) \leq \sqrt{\bar{a}_{\mathrm{lat}}/\big|\kappa(s)\big|} \tag{1}$$

where $\bar{a}_{\mathrm{lat}}(s)$ represents the maximum lateral acceleration limit at the position s, while κ symbolizes the curvature. Speed constraints due to pedestrian clearance are depicted through piecewise curves, following

$$\bar{v}(c) = \begin{cases} v_{\min} & \text{if } c < \delta_{\min} \\ \operatorname{lerp}(v_{\min}, v_{\mathrm{mdn}}, c) & \text{elif } c < \delta_{\mathrm{mdn}} \\ \operatorname{lerp}(v_{\mathrm{mdn}}, v_{\max}, c) & \text{elif } c < \delta_{\max} \\ v_{\max} & \text{else} \end{cases} \tag{2}$$

where $\operatorname{lerp}(\cdot)$ represents the linear interpolation function. The variable c represents the clearance to pedestrians at a specific sampling point, taking into account the shape of the AV. The symbol $\delta_{(\cdot)}$ signifies predefined clearance thresholds, while $\bar{v}(\cdot)$ represents the associated speed limits.

MOTION PLANNING

The objective of motion planning is to generate a trajectory for the AV that is both safe and kinodynamically feasible, in accordance with the decision made by the behavior planner. To achieve this, our method involves two steps. First, a local Frenét coordinate frame is constructed based on the reference route, and the surrounding static obstacles are projected onto this frame. Subsequently, we structure the path generation problem using our Gaussian process motion planning (GPMP) framework [11]. This framework converts collision and speed limit constraints [refer to (1) and (2)] into probabilistic factors. The resultant path is derived from a maximum a posteriori problem. Upon establishing the initial path, predicted trajectories of dynamic obstacles are projected relative to this path, and an s–t graph is formulated. We then utilize a breadth-first search on this graph to identify an initial speed profile, which is later refined using piecewise polynomials through a quadratic programming problem. By integrating the path and speed profile, we assess the kinodynamic feasibility of the trajectory. If the resultant trajectory is found to breach any constraints, additional curvature constraints sourced from the current speed profile are incorporated into the path generation procedure, prompting the generation of a new path. Likewise, a fresh speed profile is constructed based on this new path. This iterative process continues until the trajectory either satisfies all constraints or exceeds the stipulated time limit.

As shown in Figure 5, for this module, the planning cycle does not operate at a fixed time interval. This is because overly frequent planning loops may exacerbate navigation performance due to increased noise from localization information and perception. Conversely, planning with a low-frequency loop may reduce the AV's reaction ability when an accident is imminent. As a result, this work adopts a rule-based trajectory updating scheme. Specifically, when the AV is simply following a path, the output trajectory is updated only when it nears completion. However, in scenarios involving potential interactions with traffic agents, the planning cycle accelerates to a higher frequency (10 Hz) to enhance its ability to safely react to potential accidents.

CONTROL

The controller plays a crucial role in ensuring the safe operation of the vehicle, especially when dealing with varying road conditions. Therefore, we develop a controller to track the trajectory generated by the upstream motion planner accurately. Since our vehicle is based on Ackermann steering, the vehicle model can be simplified to a simpler bicycle model. Following [12], our controller takes into account the coupled lateral and longitudinal dynamics of the vehicle model, which allows us to simultaneously control the steering angle and acceleration of the vehicle. The system dynamics follow

$$\frac{d}{dt}x = \mathbf{A}x + \mathbf{B}u + \mathbf{C}\dot{\psi}_{\text{tar}}. \tag{3}$$

The system state is $x = [e_{\text{lat}}, \dot{e}_{\text{lat}}, e_{\psi}, \dot{e}_{\psi}, e_{\text{lon}}, \dot{e}_{\text{lon}}]^T$, where e_{lat} is the lateral position error, e_{ψ} is the heading error, e_{lon} is the longitudinal error, and $\dot{e}$ is their rate. The control input $u = [\delta, \text{acc}]^T$ includes the vehicle's steering angle δ and longitudinal acceleration acc, while $\dot{\psi}_{tar}$ denotes the vehicle's target heading rate. Inside (3), the matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\dfrac{C_f + C_r}{mV_x} & \dfrac{C_f + C_r}{m} & \dfrac{-C_f l_f + C_r l_r}{mV_x} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -\dfrac{C_f l_f - C_r l_r}{I_z V_x} & \dfrac{C_f l_f - C_r l_r}{I_z} & -\dfrac{C_f l_f^2 + C_r l_r^2}{I_z V_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

where m represents the weight, l_f and l_r are the distance from the center of gravity of the vehicle to its front or rear tire of the bicycle model, and C_f and C_r are the equivalent cornering stiffness coefficient. In addition, matrices $\mathbf{B}$ and $\mathbf{C}$ follow

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ \dfrac{C_f}{m} & 0 \\ 0 & 0 \\ \dfrac{C_f l_f}{I_z} & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 \\ \dfrac{-C_f l_f + C_r l_r}{mV_x} - V_x \\ 0 \\ -\dfrac{C_f l_f^2 + C_r l_r^2}{I_z V_x} \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

where I_z denotes the moment of inertia, and V_x is the current longitudinal speed of the vehicle. Given the system dynamics formulations, the controller is designed based on model predictive control (MPC), and the loss function follows

$$J = \sum_{k=0}^{N} x_k \mathbf{Q}^T x_k + \sum_{k=0}^{N} u_k \mathbf{R}^T u_k$$
$$\text{s.t.} \mid \delta_k \mid \leq \delta_{\text{max}}, \text{acc}_{\text{min}} \leq \text{acc}_k \leq \text{acc}_{\text{max}}. \tag{6}$$

In the above equation, $\mathbf{Q}$ and $\mathbf{R}$ are the weight matrices. Additionally, N denotes the time horizon, δ_{max} is the maximum steering angle, and $\text{acc}_{(\cdot)}$ represents the acceleration bounds. Overall, (3) and (6) can be solved by standard quadratic programming solvers like OSQP.

EXPERIMENTAL RESULTS

This section discusses the practical applications of our autonomous shuttle, Snow Lion, on the Hong Kong University of Science and Technology campus in Guangzhou, China [HKUST (GZ)]. Between 8 June 2023 and 7 August 2023, Snow Lion operated for three periods each day, with two hours in each period, to enhance campus mobility. It followed predefined routes, covering various fixed locations. The cumulative travel distance for the AV amounted to 1,147 kilometers. Further information regarding these transportation tasks is presented in Table 1. A selection of illustrative photos captured during the tasks is presented in Figure 7. It is noteworthy that in case of setbacks during autonomous shuttle operations, such as inevitable accidents or system errors, a security guard is consistently present within the AV. In the event of any

TABLE 1. The schedule specifies the autonomous shuttle's two-hour morning departure, with station locations provided in Figure 1. Similar operational shifts take place at noon (12:00–14:00) and in the evening (17:30–19:00).

STATION	NN7	NN20	C6	SE13	GYM	1B	C1
Time	08:00	08:03	08:10				
	08:20	08:23	08:30				
	08:40	08:43	08:50	09:02	09:05	09:06	09:13
				09:25	09:28	09:29	09:36
				09:48	09:51	09:52	10:00

autonomous navigation failure, the responsibility for vehicle control is promptly transferred to the security guard to ensure passenger safety. We anticipate reduced human intervention in the shuttle tasks. Performance evaluation is based on the frequency of human interventions and other pertinent criteria, all of which are automatically recorded in the ROS bag format. Detailed results are presented in Figure 8 and Table 2. This metric registers an instance when the acceleration falls below -1 m/s^2. Additionally, "takeovers" signifies the average distance an autonomous shuttle covers before necessitating manual intervention by security personnel.

Analysis of the data in Table 2 reveals that the autonomous shuttle consistently maintains a comfortable driving performance, with an absolute deceleration of less than 1.83 m/s^2 and an absolute jerk value of less than 1.93 m/s^2 over the course of the last 1,000 km of operations. Furthermore, upon reviewing the illustration in Figure 8(b), the autonomous shuttle consistently maintains a relatively high-speed performance, with speeds exceeding 9 km/h in in over 47.4% of operational instances, but experiences speed reductions at intersections and during turning scenarios.

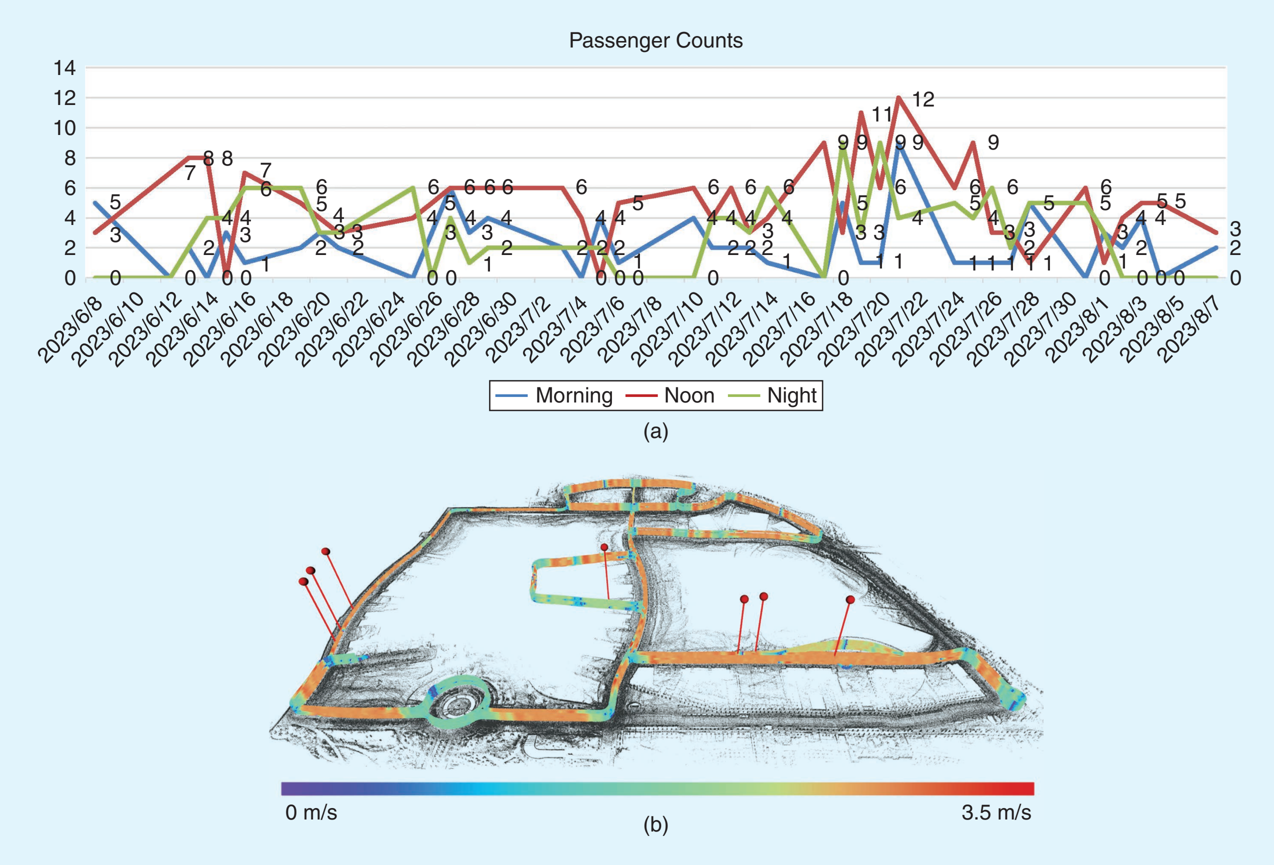
Additionally, Figure 9 presents the statistical analysis of calculation times for both behavioral planning and motion planning within our framework. The statistics were obtained by collecting data spanning 7 h of autonomous operation. The analysis indicates that our pipeline achieves real-time performance, with calculation times falling below 85.1 milliseconds in 95% of cases.

LESSONS LEARNED AND CONCLUSIONS

Navigating an AV through a university campus, marked by a highly unpredictable environment with diverse traffic participants and an absence of traffic regulations, presents formidable challenges. This section mainly discuss these challenges from a planning perspective, which predominantly emanate from several aspects.

LOCALIZATION INSTABILITY

Our framework heavily relies on localization information as it not only affects the controller's tracking performance but also influences the decision-making process (some behaviors are determined based on the localization's scenario tags). However, from the standpoint of point cloud mapping, localization instability may be unavoidable, particularly in scenarios with sparse point cloud features for mapping (empty square) or when map features evolve over time (e.g., tree growth and



(a)

(b)

FIGURE 8. The statistical results for the autonomous shuttle during its operation period. (a) The recorded passenger volumes at various times, with the periods distinguished based on the schedule in Table 1. In total, the operation spanned 39 days, averaging 10.82 passengers per day. The morning period resulted in 106 passengers, averaging 2.72 passengers. The noon period recorded 197 passengers, averaging 5.05 passengers. The night period produced 119 passengers, with an average of 3.05 passengers. (b) The speed distribution of the AV over the last 1,000 kilometers traveled within the campus.

TABLE 2. Performances of autonomous shuttle tasks over the last 1,000 kilometers traveled within the campus.

METRIC	VALUES	METRIC	VALUES
Maximum speed	11.76 km/h	Average speed	7.41 km/h
Percent speed ≥ 9 km/h	47.4%	Minimum acceleration	–1.83 m/s^2
Maximum acceleration	1.12 m/s^2	Minimum jerk	–1.93 m/s^3
Maximum jerk	1.84 m/s^3	Braking time (acceleration ≤–1 m/s^2)	0.77 km/time
Takeovers	13.34 km/time	–	–

leaf expansion affecting the effectiveness of an offline map). This failure may also manifest when the AV encounters a large truck, which can obstruct valuable environmental features for the lidar sensors. We address this issue by incorporating localization error information into the behavioral planning module. When a significant localization error arises (covariance increases), it triggers a forced speed limit $(0.5\ \mathrm{m/s})$ of the AV to alleviate the computational burden associated with the mapping process. This strategy is simple yet effective. Moreover, when the speed decreases, the mapping error rapidly recovers, restoring the normal autonomous driving status.

PERCEPTION DISTURBANCES

For results from object detection and trajectory prediction, they are doomed to be full of noise as the environment is full of uncertainty and fewer traffic regulations. We found that a constant velocity model and a lane-keeping speed planning scheme is enough for a stable navigation performance. When it really matters for the behavior decision process, for example, the lane change maneuver, we solve this by adding certain delays in the state machine in the behavioral planning, making the navigation performance more stable. Another challenge arises from the tree shades along the roadway, depicted in Figure 10. The dynamic tree shades may shift over time due to wind or extend toward the center of the road as they grow. This situation complicates the ability to maintain a secure and consistent path for the AV, particularly in narrow paths. It also causes disruptions in the perception module, leading to distorted shape detection and potentially impacting speed recognition. Exploring more effective technologies to eliminate tree-related obstacles is crucial in resolving this issue.

PLANNING ASPECT

Regarding the planning aspect, our investigation revealed that clear planning behavior for other road users and the understanding of their intentions are crucial factors for user-friendly autonomous driving. This includes recognizing whether other agents are aware of their interaction with the AV or understand that the AV acknowledges their intentions. Ensuring clarity in these interactions is vital to prevent sudden braking and ensure safety for the AV. Despite the slowdown strategy (2) implemented in our planning process effectively mitigating these concerns, resulting in an average of 13.34 kilometers for each manual takeover during the operational period, nevertheless, these challenges persist and warrant further investigation in future studies.

TRACKING ISSUES

While the raw trajectory computed by our GPMP [11] undergoes additional smoothing via an MPC solver to enhance the tracking performance, it is inevitable that a certain degree of tracking error occurs during execution by the chassis (whose

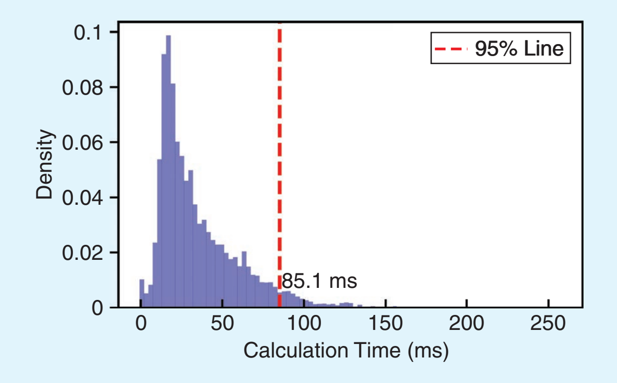


FIGURE 9. The statistics of calculation times for planning modules.

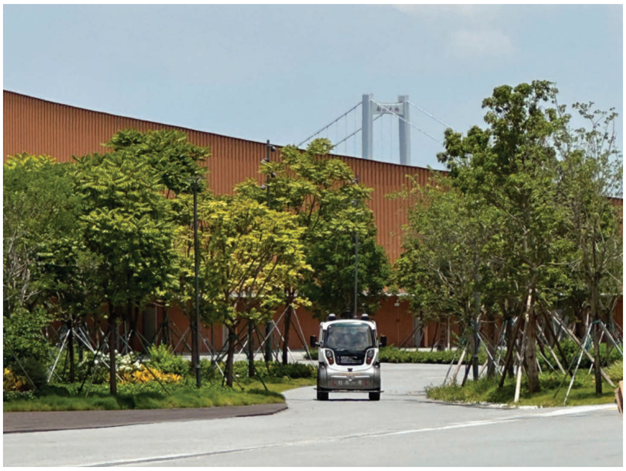


FIGURE 10. Situations where the presence of tree shade potentially impacts the perception results and navigation performance of AVs, especially when the trees sway in the wind, causing fluctuations in the boundary of the drivable area.

performance is immutable). Therefore, alongside (1), we incorporate distinct speed constraints tailored to each behavioral scenario if the desired tracking outcome cannot be achieved in the final performance.

FAST COLLISION CHECKING

In this study, bounding volume hierarchy (BVH) technology is deployed to enhance the collision checking process. This technology is widely used in robotics, and many collision checking libraries, such as the Fast Collision Library [13], have incorporated it. Specifically, we construct BVH trees based on agents' predicted trajectories at each time slot. We then update the grid of the *s–l–t* planning space by querying these BVH trees. Additionally, not all grids are queried; some grids are quickly identified as empty if their previous grid is away from obstacles (according to clearance).

OTHERS

Two additional considerations are involved during the implementation of our autonomous shuttle. First, the intention detection module faces challenges in effectively operating within a nonfixed road network environment. In such an environment, traffic agents often exhibit unexpected maneuvers, such as overtaking, jaywalking, and more, owing to the absence of fully compliant lane constraints in these scenarios. Current prediction methods heavily depend on these data and, consequently, frequently encounter difficulties in accurately forecasting the movements of other agents. Although we employ manually defined observation areas to address these challenges, particularly in intricate intersections and unstructured regions, this strategy still imposes a significant workload on fine-tuning navigation performance for new scenarios. Another concern pertains to responsibility. In the event of an accident involving an AV, determining liability requires identifying a responsible party, such as the onboard security guard. This poses challenges for the proliferation of autonomous driving technologies and underscores the need for sound legal regulations.

Overall, this article presents the design of an autonomous shuttle system, highlighting its quantitative performance and discussing challenges in scenarios with limited traffic regulation. The proposed system improves campus mobility, facilitating the transport of 235 passengers over 39 days of operation. In future research, we will endeavor to reduce takeover times and address the previously mentioned challenges.

ACKNOWLEDGMENT

This work is supported by the Guangdong Basic and Applied Basic Research Foundation (Grant 2024A1515011992) and the Guangzhou–HKUST (GZ) Joint Funding Program (Grant 2024A03J0618). Ming Liu and Jianhao Jiao are the corresponding authors.

AUTHORS

Yingbing Chen, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: ychengz@connect.ust.hk.

Jie Cheng, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: jchengai@connect.ust.hk.

Sheng Wang, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: swangei@connect.ust.hk.

Hongji Liu, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: hliucq@connect.ust.hk.

Xiaodong Mei, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: xmeiab@connect.ust.hk.

Xiaoyang Yan, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: xyanaq@connect.ust.hk.

Mingkai Tang, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: mtangag@connect.ust.hk.

Ge Sun, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: gsunah@connect.ust.hk.

Ya Wen, HKUST (GZ), Guangzhou 511400, China. E-mail: yawen@hkust-gz.edu.cn.

Junwei Cai, HKUST (GZ), Guangzhou 511400, China. E-mail: junweicai@hkust-gz.edu.cn.

Xupeng Xie, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: xxieak@connect.ust.hk.

Lu Gan, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: lganaa@connect.ust.hk.

Mandan Chao, HKUST (GZ), Guangzhou 511400, China. E-mail: mchao549@connect.hkust-gz.edu.cn.

Ren Xin, HKUST, Clear Water Bay, Hong Kong 999077, SAR, China. E-mail: rxin@connect.ust.hk.

Lujia Wang, HKUST (GZ), Guangzhou 511400, China. E-mail: eewanglj@hkust-gz.edu.cn.

Ming Liu, HKUST (GZ), Guangzhou 511400, China. E-mail: eelium@hkust-gz.edu.cn.

Jianhao Jiao, Department of Computer Science, University College London, London WC1E 6BT, U.K. E-mail: ucacjji@ucl.ac.uk.

REFERENCES

[1] T. Liu et al., “The role of the hercules autonomous vehicle during the COVID-19 pandemic: An autonomous logistic vehicle for contactless goods transportation,” *IEEE Robot. Automat. Mag.*, vol. 28, no. 1, pp. 48–58, Mar. 2021, doi: 10.1109/MRA.2020.3045040.

[2] K. Kolodge et al., “Mcity driverless shuttle: What we learned about consumer acceptance of automated vehicles,” University of Michigan, Ann Arbor, MI, USA, White Paper, 2020. [Online]. Available: https://pavecampaign.org/mcity-driverless-shuttle-what-we-learned-about-consumer-acceptance-of-automated-vehicles/

[3] A. Bucchiarone, S. Battisti, A. Marconi, R. Maldacea, and D. C. Ponce, “Autonomous shuttle-as-a-service (ASaaS): Challenges, opportunities, and social implications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3790–3799, Jun. 2021, doi: 10.1109/TITS.2020.3025670.

[4] B. Yang, S. Yan, Z. Wang, and K. Nakano, “Prediction based trajectory planning for safe interactions between autonomous vehicles and moving pedestrians in shared spaces,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10,513–10,524, Oct. 2023, doi: 10.1109/TITS.2023.3281157.

[5] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, “From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Singapore. Piscataway, NJ, USA: IEEE Press, 2017, pp. 1527–1533, doi: 10.1109/ICRA.2017.7989182.

[6] P. Wenzel, T. Schön, L. Leal-Taixé, and D. Cremers, “Vision-based mobile robotics obstacle avoidance with deep reinforcement learning,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Xi'an, China, 2021, pp. 14,360–14,366, doi: 10.1109/ICRA48506.2021.9560787.

[7] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3D object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4490–4499.

[8] J. Jiao et al., “A novel dual-lidar calibration algorithm using planar surfaces,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 1499–1504, doi: 10.1109/IVS.2019.8814136.

[9] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE Press, 2018, pp. 4758–4765, doi: 10.1109/IROS.2018.8594299.

[10] Y. Chen et al., “Efficient speed planning for autonomous driving in dynamic environment with interaction point model,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 11,839–11,846, Oct. 2022, doi: 10.1109/LRA.2022.3207555.

[11] J. Cheng, Y. Chen, Q. Zhang, L. Gan, C. Liu, and M. Liu, “Real-time trajectory planning for autonomous driving with Gaussian process and incremental refinement,” in *Proc. Int. Conf. Robot. Automat. (ICRA)*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 8999–9005, doi: 10.1109/ICRA46639.2022.9812405.

[12] R. Rajamani, “Lateral vehicle dynamics,” in *Vehicle Dynamics and Control, ch. 2*. Berlin/Heidelberg, Germany: Springer Science & Business Media, 2011.

[13] J. Pan, S. Chitta, and D. Manocha, “FCL: A general purpose library for collision and proximity queries,” in *Proc. IEEE Int. Conf. Robot. Automat.*, Piscataway, NJ, USA: IEEE Press, 2012, pp. 3859–3866, doi: 10.1109/ICRA.2012.6225337.