論文・報告

Exploration and Sweeping for Autonomous Sweeper Truck in the Geofence Scenario

Weifan Zhang* Wenbo Yu* Qian Jia* Yi-Chen Zhang*

Abstract

Autonomous road sweepers are an enabling technology in addressing cleaning tasks at public open spaces with both commercial and social values. In this paper, we investigate exploration and sweeping tasks in geofence scenarios (e.g., a parking lot) using autonomous road sweepers. We firstly design an autonomous truck system including sensors and software modules. Then, to solve the exploration and sweeping tasks, we develop algorithms with latest artificial intelligence (AI) technology. Finally, we validate the whole system by finishing a demonstration in the parking lot of Isuzu Technical Center of America (ITCA) using an Isuzu MY21 truck. The demonstration simulates the tasks that given an unknown geofence area, the sweeper truck explores the environment, builds the map, and follows the generated path to clean and coverage most of the areas. The effectiveness of the proposed approach is verified.

1 Introduction

With the continuous improvement of artificial intelligence, self-driving technology has become the trend of automobile development in the future. The combination of high-tech self-driving technology and daily sanitation sweeper is a bold, innovative, and practical trial. Capable of working overnight without human interventions, the autonomous road sweeper is a promising solution to effectively clean public open spaces with a reduced cost [(1), (2)]. To finish cleaning tasks in different scenarios, road sweepers should have two main functions: exploration and sweeping.

Exploration is required if the map is unknown, and in such case, exploration should be executed at least once before assigning the vehicle sweeping task. While driving through the area, simultaneous localization and mapping (SLAM) enables the vehicle to build a map as localizing itself in it. To decide where to go during the exploration, a goal point could be decided through algorithms to search on the current frontiers. Then path planning algorithms take effort to generate a path for the control module to follow. In the whole process, the perception module works to detect static and dynamic vehicles, pedestrians, etc. The detection result could either be utilized to improve map quality by removing the obstacles marked on map caused by dynamic vehicles, or to help the planning system for dynamic avoidance.

Sweeping task requires the map which is built through exploration as an input. Apart from the same modules during exploration, a decision-making module is added to decide when to switch between exploration and sweeping missions.

In addition, the decision module also monitors the vehicle state and surrounding objects to react for emergency. For the planning module, a cost-effectiveness coverage path planning (CPP) algorithm need to be appended. However, designing the path planning algorithm for the sweeping stage is harder compared with the exploration stage for two reasons. First, the sweeper needs to turn more frequently to cover the geofence area; however, road sweepers are typically large in size [(3)], leading to a large vehicle turning radius. Second, since the coverage radius of autonomous road sweepers is close to half the vehicle width, the vehicle needs to operate near obstacles to maximally cover an area [(4)], making the coverage path sensitive to the map uncertainty. As a result, to ensure safe sweeping in an obstacle-cluttered environment, the coverage path planning algorithm should take into account the vehicle dynamics and the uncertainty in the map from the perception system.

To solve these problems, we develop a complete autonomous truck system with sensor configuration and software modules including perception, localization, planning, decision, and control. We adopt ROS system (Robotic Operating System) to mount different parts

* Isuzu Technical Center of America, Inc.

together. For the exploration and sweeping tasks, we design original path planning algorithms and compare the performance with other benchmarks. To validate the effectiveness of the proposed system, we accomplish a demonstration which simulates the tasks that given an unknown geofence area, the sweeper truck explores the environment, builds the map, and follows the generated path to clean and coverage most of the areas. It was done in the parking lot of Isuzu Technical Center of America (ITCA) using an Isuzu MY21 truck.

Based on the discussion above, the novelties and contributions of this paper are summarized as follows.

- A complete autonomous truck system is built and presented for real-word application.
- This paper provides a hierarchical planning system. It can highly efficient build the map at the same time avoid dynamic and unforeseen objects with considering vehicle constrains.
- The paper presents newly developed CPP algorithm for autonomous road sweepers in the obstacle-cluttered environments.
- The effectiveness and robustness of the proposed method in real-world applications are validated by the experiments.

The remainder of this paper is organized as follows. In Section 2, we provide the hardware and software system developed for the road sweeper. The detailed algorithms are then introduced in Section 3. In Section 4, the proposed approaches are validated through a demonstration. Finally, conclusions and future work are given in Section 5.

2 System Description

In this section, the setup of the testing vehicle is first introduced. An overview of the road sweeper system is shown in Figure 1.

2.1 Hardware

In our case, the test truck is equipped with three mechanical LiDARs and three industrial cameras. Figure 2 gives the sensors layout of the truck. The long range 40-channel mechanical LiDAR is mounted on the top of the truck head to have a board view, and two short range 64-channel mechanical LiDARs are placed on each side of the truck head to avoid blind spots. All LiDARs are running at the rate of 10 Hz. Three industrial cameras are installed in front of the truck to achieve approximately 150 degrees of field of view. All cameras are running at the rate of 15 Hz.

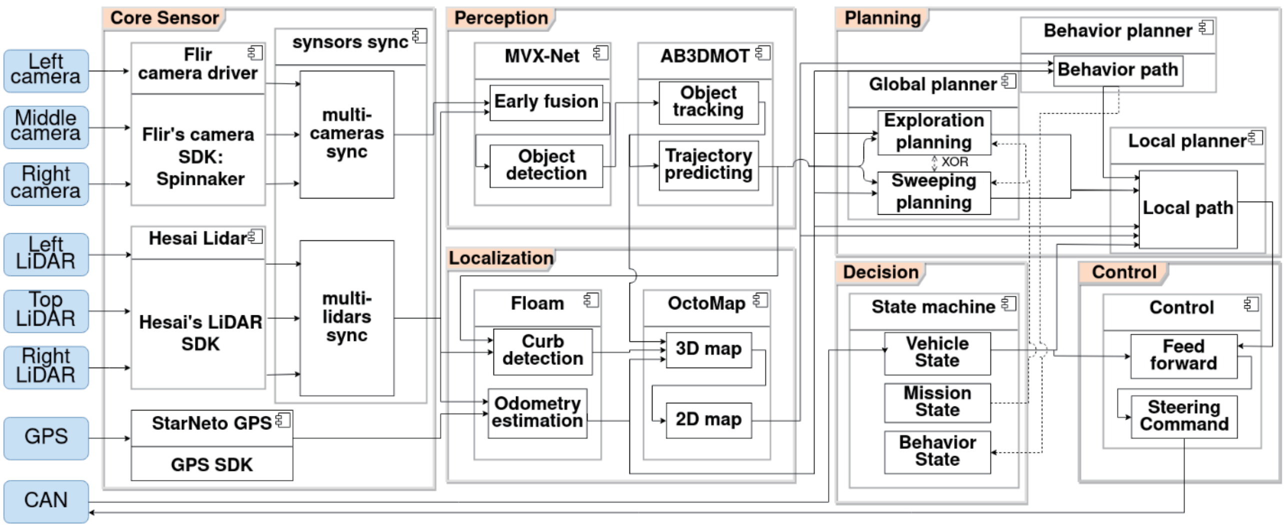


Figure 1　An overview of the systems involved in the perception, localization, planning, and control of the autonomous road sweeper.

Figure 2 Illustration of the sensors layout of the truck.

2.2 Perception and Tracking

As the "eye" of autonomous vehicles, the perception algorithm is designed to intelligently perceive the surrounding environments, which provides 3D information of surrounding objects to the downstream modules of the autonomous driving system. Moreover, by using 3D perception results as inputs, the tracking algorithm is implemented to track multiple objects over time. The tracking results will be the inputs of the downstream modules for decision making and planning algorithms.

2.3 Localization and Mapping

The important characteristic that could assist in autonomous navigation is the ability of the vehicle to concurrently construct a map for an unknown environment and localize itself within the same environment. This computational problem is known as SLAM.

To achieve the online SLAM task, we make use of the combination of the measurements from LiDAR, IMU, and GNSS to estimate the vehicle state in an unknown environment and output the vehicle's pose (i.e., position and orientation). The vehicle's pose and the lidar measurements are further used for registration of the point cloud to build a 3D occupancy grid map. In this work, the size of each grid in the occupancy grid is set to be 0.5 m × 0.5 m.

2.4 Decision Making

As the "brain" of autonomous vehicles, decision-making system is significant for the safe and efficient driving of vehicles. Decision making is the transition between the perception module and the motion planning module. In general, the inputs of decision-making system are environmental clues and status of ego vehicle, while the outputs are a series of strategies including high-level behaviors and low-level control commands that are fed into the motion planning module.

2.5 Planning

To solve the optimal planning task with abstraction and simplification, we break the task up into a hierarchy of optimization problems. By doing this, we can tailor the inputs and outputs of each optimization problem to the correct level of abstraction. At the top of this hierarchy is the mission planning, which focuses on solving the autonomous driving mission of navigating to the destination at the map level. The next level is the behavior planning, deciding which behaviors the vehicle should take. We then adopt a local planner to calculate a collision-free path and a velocity profile to the required goal state. Finally, the computed motion plan is passed to the controllers to follow. Each of these optimization problems has different objectives and constraints to solve it, which we will discuss in detail in the next section.

3 Methodology

In this section, the exploration and sweeping mission are firstly introduced. Then, the detailed algorithms for each module is formulated.

3.1 Tasks

3.1.1 Exploration

When the sweeper enters an unknown area for the first time, the exploration task is activated. The missions of the exploration task are fully exploring the designated area and constructing a map in the meantime. While exploring, obstacles like parked vehicles, curbs, and other static objects are plotted on the map. Dynamic obstacles, such as moving vehicles and pedestrians, won't be recorded into the final map, but they will be used for collision avoidance.

To explore continuously, the sweeper needs to update its target and plan paths at a fairly high frequency on the constructing map. Once the quality and completeness of the map reaches the desired level, the exploration mission is finished and the update of the map is terminated.

3.1.2 Sweeping

The sweeping mission requires the exploration mission to be completed at least once. The goal of the sweeping mission is to cover the designated area by the width of the sweeper cleaning system. At the beginning of the sweeping mission, a path optimizer generates a relative optimal sweeping path with high coverage rate and short traveling distance. Then, the sweeper follows the path and avoids any dynamic obstacles. The sweeping mission ends when the entire path is executed.

3.2 Perception and Tracking

3.2.1 3D Perception

As mentioned in Section 2, three LiDARs and three cameras are installed and need to be calibrated together. We use the open source ROS camera calibration package to get the intrinsic value for the camera and the Velo2cam [5] to get the extrinsic value between cameras and LiDARs. These calibrations are only needed to be done once unless any sensors layout changes.

As shown in Figure 3, the perception process has three steps. First, three LiDARs fusion is done to increase the density of the point cloud on objects. Second, using calibration parameters, the modified MVX-Net [6] fuses multi-modal low-level features from 3D LiDARs and industrial cameras to generate 3D information of surrounding objects in the local coordinate frame, including locations, dimensions, orientations, and confidence scores of the objects. Finally, combined with the localization information, these outputs in the local coordinate frame are converted into the global coordinate frame as the inputs for tracking component.

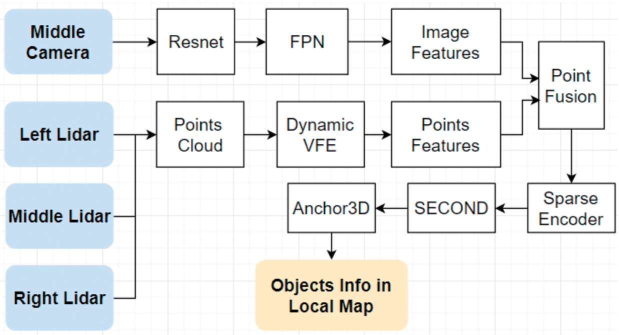


Figure 3 Structure of deep learning based neural network

3.2.2 3D Multi-Object Tracking

To generate dynamic and static objects, the AB3DMOT algorithm [7] is adopted. It mainly includes 4 modules, which are 3D object detection, 3D Kalman filter, data association, and birth and death memory. 3D object detection accepts the point cloud to generate perception results. 3D Kalman filter predicts the state of object trajectories from the previous frame to the current frame for state estimation. The Hungarian algorithm is utilized for data association. The Birth and death memory module is used for adding new trajectories or removing lost trajectories. After all these processes, dynamic and static objects can be differentiated.

3.3 Localization and Mapping

To achieve online SLAM task, we make use of the combination of F-LOAM [8] and OctoMap [9] to localize the truck and build the surrounding map in an unknown environment at the same time.

For the LiDAR odometry, F-LOAM formulates the SLAM problem as scan-to-scan match and scan-to-map refinement. The idea is to extract edge and planar features from the laser scan, and then the features are used to estimate the optimal pose of the truck between the current frame and the global map by minimizing point-to-plane and point-to-edge distance. In addition, we also modify the F-LOAM package and make it capable of receiving sensor data from IMU and GNSS. The estimated LiDAR odometry is then fused with the odometry from IMU and GNSS sensors to have a better 3D pose estimation and at the same time keep low computational cost.

The localization results are then passed to OctoMap for constructing a 3D map. The mapping approach is based on octrees and uses probabilistic occupancy estimation. It explicitly represents not only occupied space, but also free and unknown areas. The resulting 3D occupancy map is then used to generate multi-layered projected 2D maps, which allows the planning component to plan a drive-able path as well as avoiding any collision.

3.4 Decision Making

We adopt finite state machine (FSM) to implement our decision-making module. FSM is the most representative rule-based method. With discrete inputs and outputs, corresponding actions are generated depending on the responding to external events and states of agents are then transited from one to another.

Based on our tasks, we design a mission state machine as shown in Figure 4. There are two main phases: exploration and sweeping. First, when the vehicle is started, the current state directly moves forward to the exploration state and sends out signals to the planning module to start the exploration task. Then it waits for the exploration finish signal and changes to sweeping state. After entering the sweeping state, it will generate a signal, which triggers the planning module to generate coverage path.

Besides of the mission state machine, we also design a vehicle state machine and a behavior state machine. The vehicle state machine monitors the hardware's health status and generates alarm if anomaly features appears. The behavior state machine monitors whether there are unexpected vehicles or pedestrians appear. And if such case happens, it will generate signals to interrupt the current mission and trigger the planning module to generate a safe path.

3.5 Planning

3.5.1 Exploration Planner

As indicated in Figure 5, the task for the exploration planner is to analyze the map and vehicle state, and then provide a global reference path. The planning space of the exploration planner is an occupancy grid map, in which each node contains the position information (x, y) and the probability of whether it is occupied by an obstacle.

In the Figure 6, the exploration planner firstly detects frontiers, which form the boundary between known and unknown spaces. To be more specific, the unknown space has a value of -1 and the known space has a value from 0 to 100, indicating the possibility of occupancy. A frontier is a region of interest where the goal is set for the vehicle to reach. The position of the goal is obtained by calculating the distance between the frontiers and the vehicle state, and selecting the closest one.

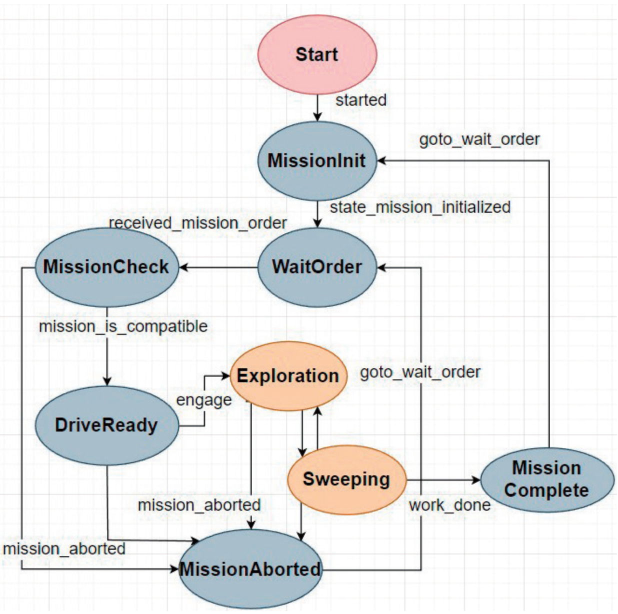


Figure 4　Mission state machine

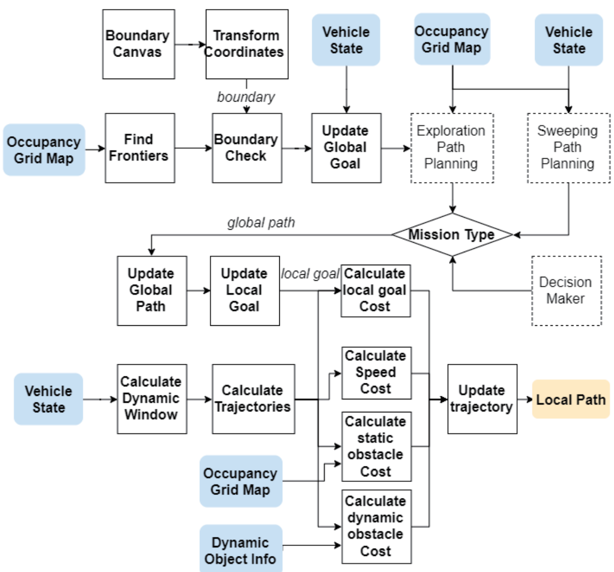


Figure 5　Planning flow chart

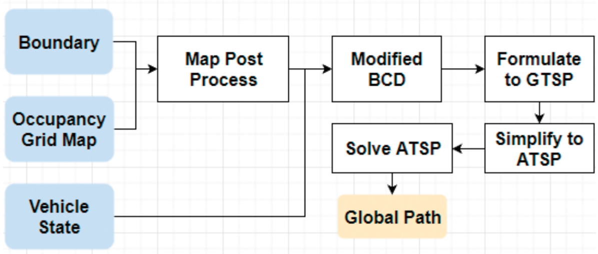


Figure 6　Exploration planner

To reach the goal position through a collision-free path, we implement a sampling-based path planning method based on the RRT-Connect algorithm (10), where both of the steering constrains and nonholonomic kinematics are taken into consideration.

The method works by incrementally building two rapidly- exploring random trees (RRTs) (11) rooted at the start and the goal configurations. While extending nodes, a sequence of points are inserted based on the kinematic model. The junction condition for both trees is checked at every iteration and the path is found once the condition meets. Notice that the junction condition also considers the vehicle model. Therefore, the exploration path is always feasible for vehicle maneuver.

The main challenge of the exploration planning is the update frequency. Although RRT-Connect is a relatively fast planning algorithm, the computation time scales up when the map grows. To stabilize the update frequency and speed up the planning process, we limit the sampling space using a time-varied ellipse, inspired by the batch informed trees (BIT) algorithm (12). The sampling space is an ellipse whose focuses are the goal position and the vehicle position, while the eccentricity of the ellipse is decided by the BIT algorithm and changed with the iteration times.

3.5.2　Sweeping Planner

To cover the designated area, we propose a novel coverage path planning algorithm, which includes three steps, as shown in Figure 7. First, the map is post-processed with morphological operations and convexification to reduce the uncertainty in the map. Second, based on the post-processed map, the boustrophedon cellular decomposition (BCD) is modified to generate path segments that cover the sweeping area considering vehicle dynamics. Finally, a generalized traveling salesman problem (GTSP) is formulated and solved to connect the path segments for a CPP path with the minimum length.

The optimization problem of the CPP has two objective variables: the coverage rate and the length of the sweeping path. The coverage rate of a sweeping area is defined as the ratio between the number of covered grids and the number of total grids. The algorithm maximizes the coverage rate and minimizes the length of the sweeping path to reach that coverage rate.

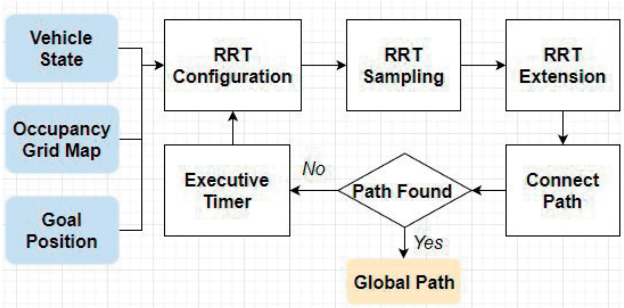


Figure 7　Sweeping planner

3.5.3　Behavior Planner

Behavior planner deals with unexpected objects. The inputs are the mission path and a list of the surrounding objects from the perception module and the tracking module. It monitors whether the surrounding objects (cars or pedestrians) block the path or tend to collide with the ego vehicle. If such case is detected, the behavior planner will start to replace the unsafe part of the mission path with the safe path. Our planner uses the hybrid A* algorithm to generate avoidance path. The hybrid A* is a robust path planning method for non-holonomic robots such as autonomous vehicles. Compared with the original A* algorithm, it also considers the limitation of the turning angle.

3.5.4　Local Planner

To increase the robustness of the system, a local planner is implemented between the global planner and the control module. The local planner is designed to generate a flexible and smooth path. For this purpose, the dynamic window approach (DWA) (13) is adopted. The DWA is capable of planning in real time and generating collision-free trajectories.

Compared to the regular DWA which has goals, obstacles, and speed costs, we add new cost functions to cope with our scenario. The cost functions are the following:

- Speed cost:

  Frequent velocity changing, including speed changing and steering oscillation, is undesirable and may cause uncomfortable maneuvers. This cost penalizes velocity changes. The cost value is proportional to the absolute difference between previous velocity state and new velocity state.

- Static obstacle cost:

  A newly introduced cost measures the distance from the considered trajectory to the static obstacles. The cost is zero when all obstacles are located out of the safe region along the trajectory.

- Dynamic obstacle cost:

  Similar to the static obstacle cost, this cost measures the distance from the ego vehicle trajectory and the trajectories of dynamic obstacles. The cost is zero when all trajectories of dynamic obstacles maintain a safe distance from the trajectory.

3.6 Control

In order to execute the path, we implement a PID controller as shown in Figure 8. First, the controller selects a sequence of way points from the local path and find a continuous fitting curve. Next, the input of the PID controller is the steering angle error and the lateral deviation, which are obtained from the fitted curve. Then, the PID controller with the dynamic vehicle model sends out a steering angle command in order to correct those two deviations. Finally, a discrete steering filter is introduced to the system, considering the thermoelectric property of the steering column.

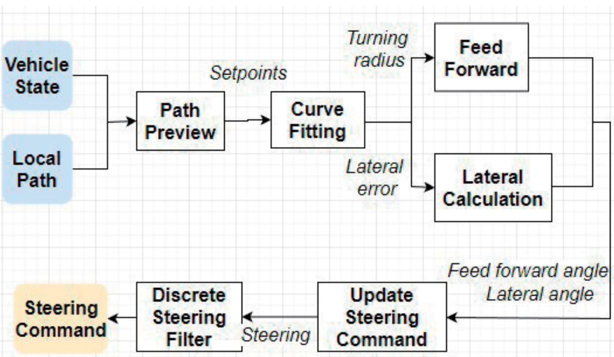


Figure 8 Control flow chart.

4 Demonstration Result

4.1 Maps and Experiment Settings

To access the performance of the proposed road sweeper system, we perform the test at the ITCA's parking lot as shown in Figure 9. The boundary highlighted in red in Figure 9 represents the designated area of the exploration and sweeping missions.

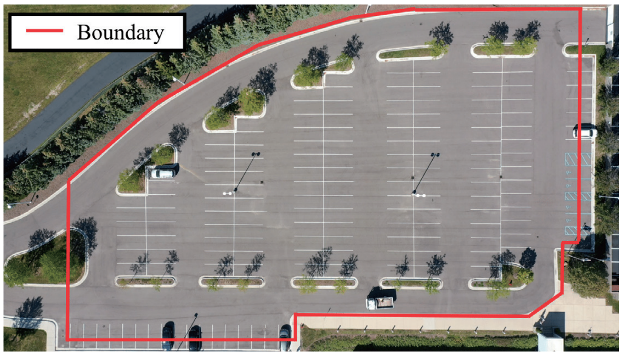


Figure 9 A bird' s eye view of the parking lot at Isuzu Technical Center of America. The truck on the road at the bottom is the testing vehicle.

4.2 Exploration Results

4.2.1 3D Map

The map built through the exploration stage is shown in Figure 10. The 3D map is then compressed to occupancy grid map.

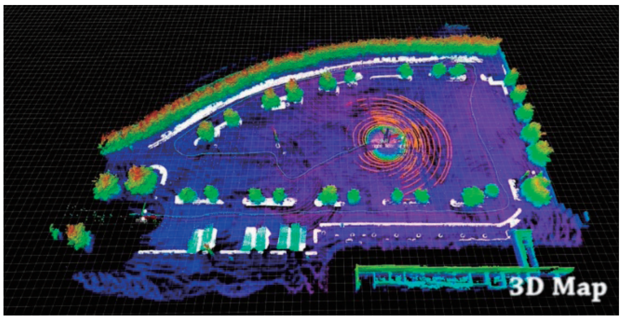


Figure 10 3D Map

4.2.2 Object Avoidance

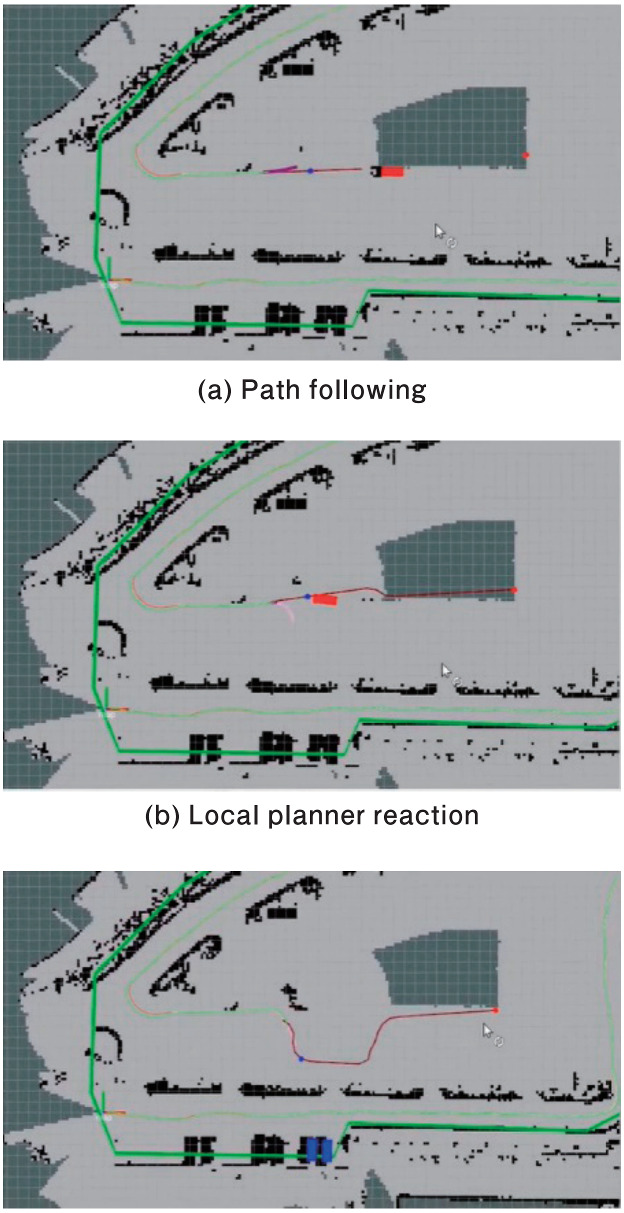
As is shown in Figure 11(a), during the exploration stage, an unexpected moving vehicle the exploration path. Though the exploration planner fails to generate a collision-free global path in time. The local planner detects the possible collision and generates a short collision-free path to avoid the moving vehicle Figure 11(b). It enables the vehicle to pass the object when the exploration planner finds a new path for vehicle to follow Figure 11(c).

4.3 Sweeping Result

In this section, the CPP algorithm proposed in Section 3 is validated. To demonstrate the dynamic feasibility of the proposed approach, a test-bed vehicle is deployed in the explored parking lot to evaluate the path. The coverage radius of the vehicle is set as 2 grids. The number of iteration for genetic algorithm is 1500 Figure 12.

- **Evaluation of the executed CPP path**

To further validate the proposed approach, the test-bed truck is employed to follow the CPP path planned by the proposed approach. The executed vehicle path is given in Figure 12. The coverage rate and the path length of the executed path are 83.01 % and 2924 m, respectively. As shown in Figure 12, the planned path can be well followed by the vehicle, and all the obstacles in the parking lot can be avoided. Meanwhile, the loss in the coverage rate is minimal (i.e. 1.61 %) between the planned and executed paths.



(a) Path following

(b) Local planner reaction

(c) Global planner reaction

Figure 11 Object avoidance example. The unexpected moving vehicle is represented in the red box in (a) and (b). The crimson line represents the global path planned by the exploration planner. The pink line in (b) represents the collision-free path planned by the local planner. (c) shows the avoidance path planned by the global planner.

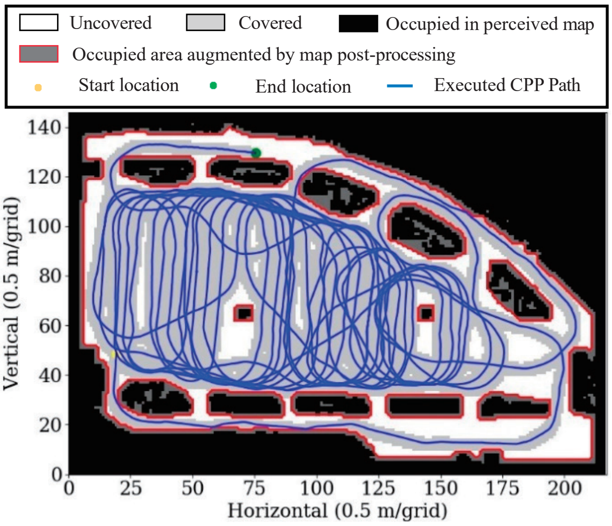


Figure 12　Executed path by following the CPP path planned with the proposed approach

5　Conclusion

In this paper we focus on exploration and sweeping tasks for autonomous driving system. We develop a hierarchical planning system which is highly efficient in building the map at the same time avoiding dynamic and unforeseen objects. We develop an original CPP algorithm with three steps: map post-processing, decomposition and optimal. A complete autonomous sweeper system is built for validation. The system contains sensors configuration and software modules including perception, localization, planning, decision making, and control. During the validation experiment, the sweeper truck is able to build a 3D point cloud map of the designated parking lot. Some designed behaviors have been tested during the exploration process that includes making right angle turns in a single lane, dodging incoming moving vehicles, and avoiding static obstacles. In the sweeping mission experiment, an optimal sweeping path with about 80 percent coverage rate is generated and successfully executed. The results show that the sweeper truck meets the design requirement. The proposed system can also be extended to complete other sweeping related tasks thanks to the modularized component design.

あとがき

米国のスーパーなどの大きな駐車場を深夜に清掃装置を架装した商用車が清掃しています。いすゞNシリーズの使われ方の一つであるそのような清掃車の自動運転化（無人運転化）という使命を受けて自動運転制御を自社開発しました。本稿で報告させていただいたように、初見の駐車場を無人で探索、最も効率的な清掃パターンを内包した最適化したシステム制御により自動清掃を実施し、かつ探索時点では存在しなかった外乱（他車両・自転車・歩行者の侵入など）にも耐えうるシステムとなっています。今後はこの開発で培った経験・スキルをいすゞ自動車といすゞ中央研究所の自動運転開発チームの一員として発揮していきます。

最後に例年と同じ言葉になりますが「いすゞの中で役に立ついすゞテクニカルセンターオブアメリカ　Inc.でありたい」と社員一同が一丸となって取り組んでいる様子が本報告で少しでも伝われば幸いです。

（ITCA チーフエンジニア　深井 泰雄）

References

(1)　E. Prassler, D. Schwammkrug, B. Rohrmoser, and G. Schmidl, “A robotic road sweeper,” in International Conference on Robotics and Automation, IEEE, vol. 3, (2000), p. 2364-2369.

(2)　J. Jeon, B. Jung, J. C. Koo, H. R. Choi, H. Moon, A. Pintado, and P. Oh, “Autonomous robotic street sweeping: Initial attempt for curbside sweeping,” in International Conference on Consumer Electronics. IEEE, (2017), p.72-73.

(3)　C. Wang and G. Parker, “Analysis of rotary brush control character- istics for a road sweeping robot vehicle,” in International Conference on Mechatronics and Control. IEEE, (2014), p.1799-804.

(4)　C. Luo, S. X. Yang, D. A. Stacey, and J. C. Jofriet, “A solution to vicinity problem of obstacles in complete coverage path planning,” in International Conference on Robotics and Automation, IEEE, Vol.1, (2002), p.612-617.

(5)　J. Beltra´n, C. Guindel, A. de la Escalera, and F. Garc´ıa, “Automatic extrinsic calibration method for lidar and camera sensor setups,” IEEE Transactions on Intelligent Transportation Systems, 2022.

(6)　V. A. Sindagi, Y. Zhou, and O. Tuzel, “Mvx-net: Multimodal voxelnet for 3d object detection,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, (2019), p.7276-7282.

(7)　X. Weng, J. Wang, D. Held, and K. Kitani, “3d multi-object tracking: A baseline and new evaluation metrics,” in 2020

IEEE/RSJ Interna- tional Conference on Intelligent Robots and Systems (IROS). IEEE, (2020), p.10359-10366.

(8) H. Wang, C. Wang, C.-L. Chen, and L. Xie, “F-LOAM: Fast LiDAR odometry and mapping,” p.4390-4396. [Online]. Ava

(9) A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” Autonomous Robots, 2013, software available at https://octomap.github.io. [Online]. Available: https://octomap.github.io

(10) J. J. Kuffner and S. M. LaValle, “Rrt-connect: An efficient approach to single-query path planning,” in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automa- tion. Symposia Proceedings (Cat. No. 00CH37065), IEEE, vol.2,(2000), p. 995-1001.

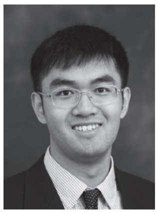
(11) S. M. LaValle et al., “Rapidly-exploring random trees: A new tool for path planning,” 1998.

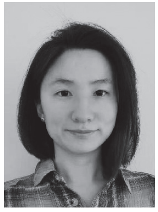
(12) J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in International Conference on Intelligent Robots and Systems. IEEE, (2014), p. 2997-3004.

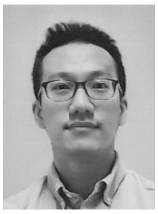
(13) D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” IEEE Robotics & Automation Magazine, Vol. 4, No. 1(1997), p. 23-33, 1997.

著　者

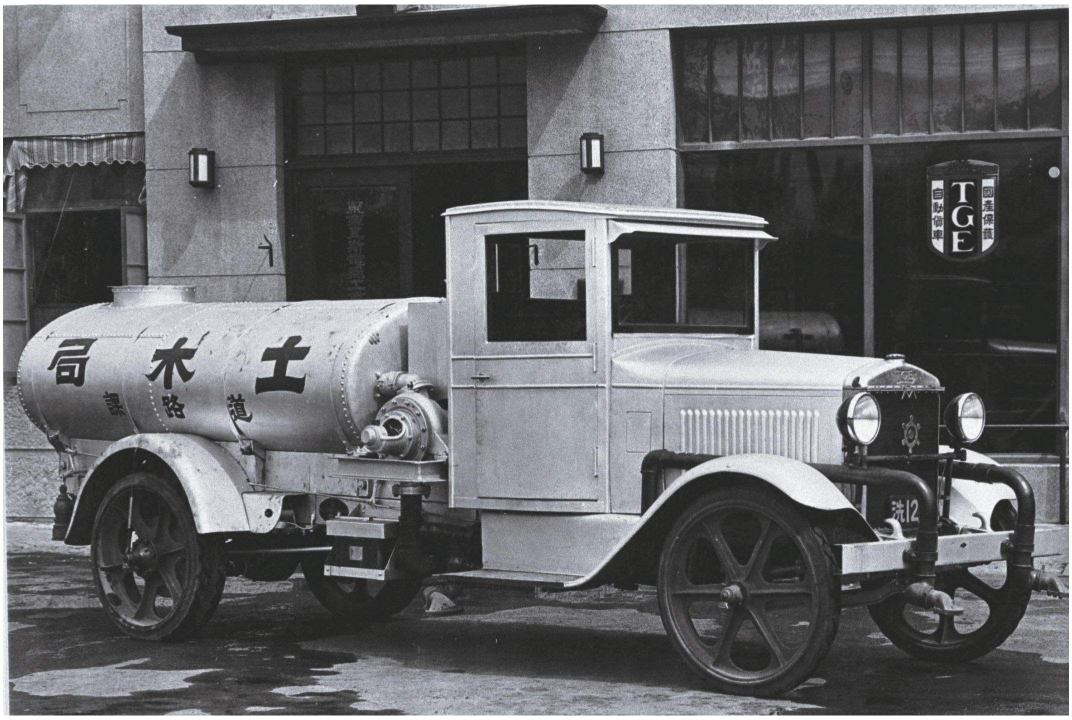

Weifan Zhang


Wenbo Yu


Qian Jia


Yi-Chen Zhang

◇いすゞ歴史の一こま



TGE-L型散水車（1930年ごろ）