Autoware in Construction: Gap Analysis and LiDAR Perception Toward Off-Road Autonomous Driving

Yu Otsuki[1], Teja Emmey[2], Sena Matsushita[2], Akiro Harada[2], and Takeshi Miura[2]

***Abstract*— Autoware is an open-source autonomous driving software platform widely adopted by researchers and industry developers. Originally developed primarily for public-road applications, including passenger vehicles, taxis, and buses, Autoware is increasingly being extended to off-road environments such as construction and agricultural sites. Construction sites, however, differ fundamentally from public roads and challenge many assumptions underlying conventional autonomous driving systems. They are characterized by unstructured terrain, airborne dust, continuously evolving site conditions, and construction-specific objects. This paper presents lessons learned from ongoing efforts to extend an Autoware-based autonomous driving system to large dump trucks operating at construction sites. We identify gaps between public-road and off-road construction applications across the sensing, mapping, localization, perception, planning, and control modules of the Autoware stack. We then discuss potential solutions for addressing these gaps, with particular emphasis on ongoing LiDAR-based perception pipeline development and findings from field testing. Finally, we propose a roadmap toward end-to-end autonomous driving for construction vehicles.**

I. INTRODUCTION

Autoware is a modular, open-source autonomous driving software stack that has supported research and development across a wide range of applications, including passenger vehicles, buses, taxis, delivery vehicles, industrial vehicles, and racing cars. Originally implemented on ROS 1 [1] and since migrated to ROS 2, its architecture integrates sensing, mapping, localization, perception, planning, control, and vehicle-interface components (Fig. 1). This modularity allows individual components to be replaced or extended for each operational design domain (ODD).

Construction sites represent a substantially different ODD from public roads. Construction environments introduce rough terrain, airborne dust, continuously changing site geometry, feature-poor environments, construction-specific objects, and coordinated operations among heterogeneous machines [3]. These conditions challenge several assumptions embedded in public-road autonomous driving systems and require adaptations throughout the Autoware stack. Recently, the Autoware Foundation has initiated an off-road pilot encompassing mining, racing, and planetary applications [4].

As part of our effort to extend Autoware to construction environments, we are integrating the Autoware software stack into a large construction dump truck, as shown in the center of Fig. 2. The vehicle has been retrofitted with LiDAR, camera, GNSS, and IMU sensors, onboard computers, and an electronic control unit interface.

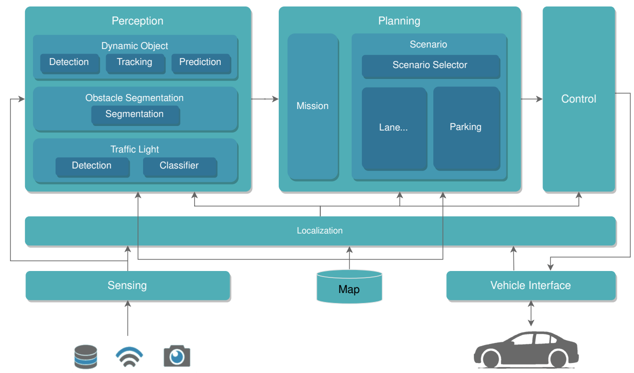

Fig. 1. Overview of Autoware modular architecture [2].

Fig. 2. Autoware-integrated autonomous construction dump truck tested in an off-road environment.

Based on this ongoing development effort, this paper makes three contributions. First, it presents a module-level gap analysis between the existing Autoware architecture, which was developed primarily for public-road autonomy, and the requirements of off-road autonomy in construction environments. Second, it distills lessons learned from the development and field testing of a LiDAR-based perception pipeline for construction sites. Third, it discusses a roadmap toward end-to-end (E2E) autonomous driving for construction vehicles.

II. Technology Gaps in Autonomous Driving Between Public Roads and Construction Sites

We identified key differences between public-road and off-road construction environments in each module of the Autoware stack, as summarized in Table I. Challenges in construction environments are interdependent across these mod-

This work was conducted while the first author was with TIER IV.

[1]The author is an Assistant Professor at the Department of Civil and Environmental Engineering, University of California, Irvine (UCI), Irvine, CA, USA. yu.otsuki@uci.edu

[2]The authors are with TIER IV, Inc., Shinagawa, Tokyo, Japan. {teja.emmey, sena.matsushita, akiro.harada, takeshi.miura}@tier4.jp

TABLE I

TECHNOLOGY GAPS IN AUTONOMOUS DRIVING BETWEEN PUBLIC ROADS AND CONSTRUCTION SITES.

Modules	Public-road capabilities available in Autoware	Off-road construction challenges and requirements
Sensing	Supports LiDAR, camera, radar, GNSS, and IMU sensors. Standard sensor drivers and sensor calibration routines are also provided.	Dust can cause occlusion. Mud can contaminate sensor surfaces. Large vehicle bodies can create blind spots. IMU measurements are degraded by strong vibrations. Severe vertical vibration requires 3D LiDAR distortion correction.
Mapping	Lane geometry and surrounding structures are static, so high-definition maps can be maintained offline. Environments are geometrically feature-rich, with abundant vertical structures.	Construction sites evolve continuously through excavation, grading, and material placement, requiring frequent map updates.
Localization	GNSS/RTK can be unreliable in tunnels, urban canyons, and beneath elevated roads. Map-based localization performs well with feature-rich high-definition maps. Planar localization in x, y, and yaw is often sufficient.	GNSS/RTK is generally reliable under open skies. Map-based localization is challenged by feature-poor, changing environments. Full 6-DOF pose estimation is often required due to slopes and rough terrain.
Perception	Supports detection, tracking, and sensor fusion using both non-ML and ML-based approaches. Its ML detectors support objects such as cars, trucks, buses, bicycles, and pedestrians.	Relevant objects include heavy construction machinery, passenger vehicles, construction workers, cones, rocks, berms, and temporary structures. The drivable space is unstructured and must be identified.
Planning	Scenario-based, such as lane driving and parking on predefined road networks, governed by traffic rules and dominated by forward driving.	Haulage roads are stable over days to weeks and can be given as waypoints, but loading and dumping areas change continuously and require coordination with excavators and dozers. Traversability must be evaluated to determine optimal trajectories off the haul road.
Control and Vehicle Interface	Tracks the planned trajectory by regulating steering and acceleration, typically assuming Ackermann-steered vehicles.	Must account for uneven terrain, wheel slip, steep slopes, and payload changes that alter mass and center of gravity, as well as articulated steering where applicable; hydraulic actuation introduces larger delay and stronger non-linearity than road-vehicle actuators. Additional machine-specific work equipment, such as the dump bed.

ules. Therefore, achieving construction autonomy requires system-wide integration rather than the isolated replacement of individual algorithms.

III. LESSONS LEARNED FROM FIELD DEVELOPMENT OF LIDAR-BASED PERCEPTION

Among the technology gaps in Table I, this section presents our LiDAR-based perception pipeline: LiDAR sensor → sensor driver → preprocessing filters → multi-LiDAR concatenation → postprocessing filters → ground segmentation → dust filtering → object detection, developed by extending the existing Autoware pipeline. Specifically, this section reports lessons learned from the LiDAR-based ground segmentation and dust filtering, as well as extrinsic sensor calibration. This work used Hesai AT128 LiDARs.

A. Ground segmentation

In Autoware, ground removal is commonly performed using `scan_ground_filter` [5] to separate ground points from non-ground objects in LiDAR point clouds. The algorithm divides the point cloud into radial sectors and classifies points using slope and height thresholds. However, on uneven off-road terrain, one set of thresholds misclassifies uneven ground as obstacles, whereas the other set of thresholds incorrectly classifies small obstacles, such as rocks, as ground.

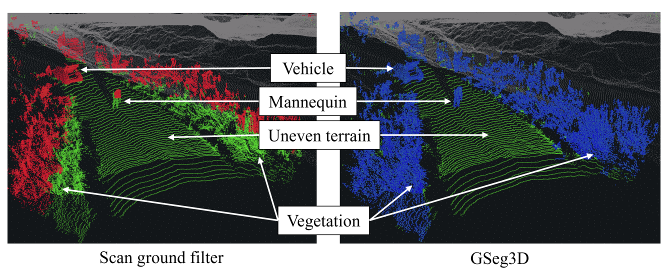

Fig. 3. Comparison of LiDAR ground segmentation results obtained at an off-road environment using the scan ground filter (left) and GSeg3D (right). The raw point cloud is shown in green, while non-ground points identified by the scan ground filter and GSeg3D are highlighted in red and blue, respectively.

Therefore, ground segmentation algorithms designed for or adaptable to off-road environments are needed. Recent methods, such as `Patchwork++` [6] and `GSeg3D` [7], have shown promising performance on public-road benchmarks. We compared `GSeg3D` with `scan_ground_filter`, as shown in Fig. 3. `GSeg3D` successfully classified the mannequin, vehicle, and vegetation as non-ground while preserving the uneven terrain as ground. In contrast, `scan_ground_filter` misclassified parts of these objects.

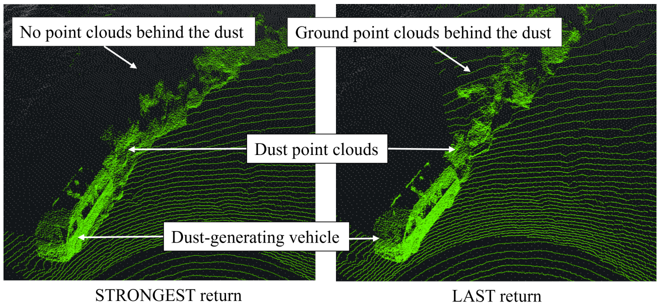

Fig. 4. Comparison of LiDAR return modes under dusty conditions. The strongest return (left) is more affected by dust particles, whereas the last return (right) better preserves ground surfaces and objects behind the dust.

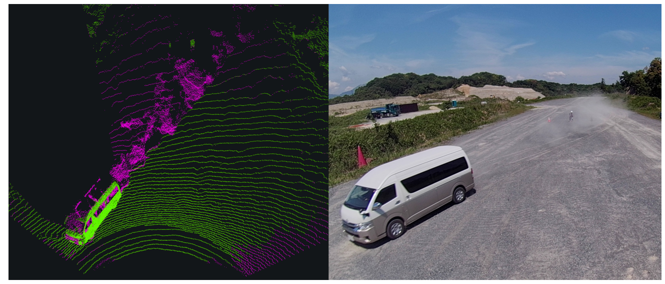

Fig. 5. LiDAR point cloud (left) and camera image (right). In the LiDAR point cloud, LAST returns with intensity values of 1 or lower are shown in pink; others are shown in green. Dust returns and ground returns obscured by the dust are predominantly pink, whereas unobscured ground returns and the cones and mannequin behind the dust appear green.

For further development, traversability analysis is another important direction [8]. Ground segmentation alone is insufficient because not all ground surfaces are safe for a vehicle to traverse. Traversability estimation should account for terrain properties, including slope, roughness, and material, while detecting positive obstacles, such as rocks, ruts, and berms, and negative obstacles, such as holes, trenches, and drop-offs. For example, a vehicle may be able to traverse a 30-cm-deep rut but should avoid a 30-cm-high rock. Such judgments therefore depend on vehicle geometry, dynamic state, payload, and operating conditions, which means traversability cannot be baked into a static map.

B. Dust filters

Airborne dust is a major challenge at construction sites. In our field deployments, it arose from various sources, including wind, tires traveling over dry soil, material spilling from a moving truck's dump bed, loading and dumping operations, and engine exhaust.

LiDAR return-mode selection is an important sensor configuration choice for operation in dusty environments. Many modern LiDARs support multiple return modes, including `STRONGEST`, `LAST`, and `FIRST`, which select the detected echo with the greatest received-signal strength, the farthest valid echo, and the nearest valid echo, respectively. Fig. 4 compares the `STRONGEST` and `LAST` return modes under dusty conditions. In the `STRONGEST` mode, dense nearby dust produces the strongest received echo, causing the dust cloud to dominate the point cloud. In contrast, the `LAST` mode better preserves the ground and other surfaces beyond the dust cloud and is therefore preferable in this setting.

Beyond return-mode selection, dust returns can also be identified based on their characteristic properties. LiDAR returns from airborne dust typically exhibit low intensity, spatial dispersion, and limited temporal persistence [9]. Although neighbor-search methods can leverage spatial and temporal information, they are generally computationally expensive for real-time operation. In contrast, simple intensity-based filtering has been shown to be effective [10].

This low-intensity characteristic of dust is evident in our experiment shown in Fig. 5, in which a vehicle generates a dense dust cloud. LiDAR points with intensity values of 1 or lower are highlighted in pink and closely correspond to returns from dust and ground obscured by the dust, while the intensities of cones and mannequins remain high. Based on this observation, an intensity-based filter can remove dust from non-ground LiDAR points identified by ground segmentation, thereby reducing false-positive object detections.

It should be noted that, depending on the sensor, the reported intensity may be a calibrated, range-independent quantity and may not directly correspond to the metric used for return selection. Consequently, a dust return in `STRONGEST` mode can still exhibit low intensity.

When multiple LiDARs are used, however, intensity-based filtering breaks down because intensity values are not directly comparable across LiDARs [11]. In our experience, differences in sensor architectures (e.g., spinning versus solid-state), as well as differences among manufacturers and among sensor models from the same manufacturer, result in distinct intensity ranges. Consequently, applying a single intensity threshold to a combined multi-LiDAR point cloud would require cross-sensor intensity normalization. Because such normalization is difficult in practical deployments, it is not adopted in this work.

Instead, we apply a sensor-specific low-intensity threshold in a preprocessing filter at the output of each LiDAR driver. LiDAR points below the low-intensity threshold are flagged separately for each LiDAR before the point clouds are concatenated, thereby eliminating the need for cross-sensor intensity normalization. These flagged points are retained during ground segmentation of the concatenated point cloud, since removing them beforehand would destabilize the ground estimation. After ground segmentation, non-ground points flagged as low-intensity are removed by the dust filter, while the remaining high-intensity non-ground points are passed to the subsequent object detection pipeline.

C. Extrinsic sensor calibration

Due to road-width limits and transportation regulations, large construction vehicles often cannot be transported fully assembled. Instead, they must be partially disassembled and reassembled at the construction site. Since sensors are often removed or repositioned during this process, extrinsic calibration must be performed or verified on-site.

Existing Autoware-based calibration workflows often assume the availability of flat ground and overlapping sensor fields of view containing distinctive environmental features. These assumptions may not hold at construction sites; suitable flat surfaces may be unavailable; the large dimensions of the vehicle and widely spaced sensors may yield limited overlap between sensor fields of view; and the environment may offer few distinctive features. Sensor calibration for large vehicles under such off-road conditions remains an active research topic [12], [13]. Practical deployment therefore requires an on-site calibration method that can accommodate uneven terrain and limited sensor overlap.

IV. Roadmap toward End-to-End autonomous driving in construction

A. Addressing data scarcity

Current learning-based perception modules in Autoware have been trained primarily on public-road datasets [14]. For example, a standard LiDAR-based CenterPoint model detects five object classes: cars, trucks, buses, bicycles, and pedestrians. Construction sites, however, contain a substantially different range of objects, including heavy construction machinery, service vehicles, workers wearing high-visibility clothing, traffic cones, rocks, and berms.

Publicly available off-road datasets have largely been collected using small unmanned ground vehicles or passenger-sized vehicles, whose sensor viewpoints, sensing ranges, and occlusion patterns differ substantially from those of large construction machines. AutoMine [15] is among the most relevant datasets because it includes data collected from wide-body trucks and a mining haul truck. Nevertheless, construction-domain datasets remain scarce.

B. End-to-end architecture

Extending Autoware to construction environments should proceed incrementally. In the near term, the priority is to replace public-road assumptions with construction-specific capabilities while retaining Autoware's modular architecture. Meanwhile, on-site data collection can establish construction-specific datasets to address data scarcity and support the training of data-hungry E2E models. The longer-term objective is to integrate learned E2E models that jointly perform perception, reasoning, and trajectory generation. This direction aligns with Autoware's roadmap for gradually evolving its modular architecture toward E2E autonomy [16].

Alpamayo–Autoware [17] provides an early example of this transition in public-road environments by integrating NVIDIA's Alpamayo [18] vision-language-action (VLA) model with Autoware's planning module. The model processes multi-camera observations and vehicle-motion history to generate an Autoware-compatible trajectory together with a textual explanation of its reasoning.

For construction applications, a VLA model could also accept task-level instructions such as positioning a haul truck for excavator loading, coordinating with a dozer in the dumping area, avoiding unstable terrain, and yielding to higher-priority vehicles.

Acknowledgment

We express our sincere gratitude to the following companies in Japan, listed in alphabetical order: EARTHBRAIN, Komatsu, and Taisei Corporation, for their contributions to this work. We also sincerely thank all TIER IV team members and collaborators for their contributions.

References

[1] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 287–296.

[2] Autoware Foundation, "Autoware Architecture overview," Available: https://tier4.github.io/autoware-documentation/tier4-main/design/autoware-architecture/, accessed Aug. 1, 2026.

[3] H. Uchiito, A. Bhat, K. Kusaka, X. Zhang, H. Kinjo, H. Uehara, M. Koyama, and S. Natsume, "Robotics under construction: Challenges on job sites," *arXiv preprint arXiv:2506.19597*, 2025.

[4] Autoware Foundation, "Autoware—Off-Road Autonomy Pilot," Available: https://github.com/autowarefoundation/autoware.off-road, accessed Aug. 1, 2026.

[5] ——, "Scan Ground Filter," Available: https://github.com/autowarefoundation/autoware_universe/blob/main/perception/autoware_ground_segmentation/docs/scan-ground-filter.md, accessed Aug. 1, 2026.

[6] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 276–13 283.

[7] M. H. K. Lodhi and C. Hertzberg, "Gseg3d: A high-precision grid-based algorithm for safety-critical ground segmentation in lidar point clouds," in *2025 7th International Conference on Robotics and Computer Vision (ICRCV)*. IEEE, 2025, pp. 119–126.

[8] H. Lim, M. Oh, S. Lee, S. Ahn, and H. Myung, "Similar but different: A survey of ground segmentation and traversability estimation for terrestrial robots," *International Journal of Control, Automation and Systems*, vol. 22, no. 2, pp. 347–359, 2024.

[9] A. Afzalaghaeinaeini, J. Seo, D. Lee, and H. Lee, "Design of dust-filtering algorithms for lidar sensors using intensity and range information in off-road vehicles," *Sensors*, vol. 22, no. 11, p. 4051, 2022.

[10] K. Nakajima and N. Nishio, "Removing snowfall noise from point clouds without considering the nearest neighbors," *IEEE Sensors Journal*, vol. 25, no. 3, pp. 5212–5220, 2024.

[11] K. Viswanath, P. Jiang, P. Sujit, and S. Saripalli, "Off-road lidar intensity based semantic segmentation," in *International Symposium on Experimental Robotics*. Springer, 2023, pp. 608–617.

[12] H. Li, G. Yu, Z. Wang, F. Zhao, and P. Chen, "Online calibration of lidar and gps/ins using multi-feature adaptive optimization in unstructured environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–15, 2025.

[13] J. Wang, Y. Qiao, C. Gao, and N. Wu, "Joint optimization-based targetless extrinsic calibration for multiple lidars and gnss-aided ins of ground vehicles," *arXiv preprint arXiv:2507.08349*, 2025.

[14] S. Tanaka, S. Thapa, K. S. Tan, A. Szymko, L. Kenzo, K. Minoda, S. Tomie, K. Uetake, G. Zhang, I. Yamashita *et al.*, "Awml: An open-source ml-based robotics perception framework to deploy for ros-based autonomous driving software," *arXiv preprint arXiv:2506.00645*, 2025.

[15] Y. Li, Z. Li, S. Teng, Y. Zhang, Y. Zhou, Y. Zhu, D. Cao, B. Tian, Y. Ai, Z. Xuanyuan *et al.*, "Automine: An unmanned mine dataset," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 21 276–21 285.

[16] Autoware Foundation, "Introducing Open-Source End-to-End Autonomous Driving," Available: https://autoware.org/wp-content/uploads/2025/10/Autoware-E2E-Report_Release.pdf.

[17] ——, "Alpamayo–Autoware: Alpamayo 1.5 ROS 2 Node," Available: https://github.com/autowarefoundation/alpamayo-autoware, accessed Aug. 1, 2026.

[18] Y. Wang, W. Luo, J. Bai, Y. Cao, T. Che, K. Chen, Y. Chen, J. Diamond, Y. Ding, W. Ding *et al.*, "Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail," *arXiv preprint arXiv:2511.00088*, 2025.