Sensing Component
Sensing: Improving filtering of point cloud from LiDAR

Challenge:
Autoware applies various filters to LiDAR points used for object recognition. However, there are following issues:
- Outliers from rain droplets, fog, and dusts remain as noise in the point cloud
- Points from objects that do not need to be avoided should be removed (e.g., points from weed, plastic bags, etc.)

Objectives
Create a point cloud filter that remove points that do not need avoidance while preserving points from solid obstacles.
Perception Component
[Perception] Long range object detection using camera/LiDAR/radar

**Challenge:**

There is a very high demand for recognizing objects at long distances, such as objects 200m ahead. For example, in unprotected left turns (right turns in Japan), the ability to detect vehicles in the oncoming lane with high relative speeds from a long distance directly affects the performance of right and left turns. However, in cases where the LiDAR points become sparse at distances like 200m, it can be challenging.

**Objectives**

Sufficiency of the recognition of objects at long distances (the main target for this project is objects at around 200m) for Planning components
- It is acceptable to propose and evaluate using metrics other than mAP for this project.
- For example, Autoware uses Lanelet as a high-definition map, but an output like "There is an object with a speed of approximately yy in Lanelet ID xx" might also be acceptable.
Challenge:

Object detection in Autoware heavily relies on machine learning techniques, which currently struggle to recognize objects not labeled in the training data. On public roads, it's crucial to manage unexpected objects that the system has not previously encountered, such as prone humans, unfamiliar animals, and items like cardboard boxes. It is also necessary to accurately detect these unknown objects and make appropriate decisions on whether to avoid them. Furthermore, under conditions that generate noise for LiDAR, such as rain, snow, or dust, distinguishing between noise and actual objects is essential.

Objectives:

- **Unknown Object Detection**
  - Detect and classify unknown or unclassified objects, determining whether avoidance is necessary.
  - Identify anomalous objects that should be avoided, including very small objects detectable with minimal LiDAR measurements.

- **Noise object and pointcloud removal**
  - Remove false positives in challenging environments such as rain, dust, fog, and snow.
  - Perform filtering on either point clouds or objects, ensuring that both methods are easily tunable for adaptation in different environments.
Challenge:

Pedestrian behavior prediction is crucial for ensuring safety on roads where the boundaries between crosswalks, sidewalks, and roadways are ambiguous. The current pedestrian behavior prediction in Autoware is poor for several reasons, and we aim to improve both short-term and long-term pedestrian behavior prediction.

Reasons

- Lack of necessary information for judgment: The detection system does not incorporate information such as pedestrian posture and orientation.
- Simplistic prediction model: Only a simple control logic is used, such as extrapolating an EKF constant velocity model over time or assuming pedestrians walk straight towards a specific point on the crosswalk.

Objectives

- Short-term prediction improvement
  - Detection enhancement: Utilize camera and LiDAR information effectively to detect pedestrian posture and orientation.
  - Tracking improvement: Predict behavioral intentions (starting to move, straight walking, turning) more accurately based on the additional information.
- Long-term prediction improvement
  - Predict pedestrian future trajectories up to ~15 seconds using data-driven or end-to-end methods.
- Readiness of the proposed architecture to be integrated into Autoware
Challenge:

Turn signal recognition is one of the crucial tasks, and not only the algorithm for recognition but also how to utilize it in the entire Autoware system is an open question. One of the ways to utilize turn signal recognition is in Prediction (predicting the trajectories of other vehicles). For example, if the vehicle in front is indicating a hazard signal, the predicted trajectory should output a gradual stop. This allows the Planning module to utilize the turn signal recognition results without increasing the interface exposed by the Perception module to the Planning module.

Objectives

- Whether the proposed method is actually ready-to-use for Autoware (better if it is actually integrated).
- Whether the proposed method outputs effective prediction paths for left and right turn signals and hazard lamps (better if experiments are conducted with integration into Planning).
- Although the method of turn signal recognition itself is not the target this time, if integration experiments with turn signal recognition are conducted to increase persuasiveness, those will also be evaluated.
Planning/Control Component
Challenge:
Safety is the most important factor for autonomous driving even in emergency cases. One of such cases is when there is an unexpected behavior of dynamic objects and the ego is about to crash them. In order to deal with this kind of emergency case, the Autoware should have a feature to avoid or reduce the crash by utilizing the steering and acceleration.

Objectives
- Design and develop the feature to turn the steering wheel or accelerate/decelerate depending on the situation.
- To make this module work in the case of emergency, the feature should be independent of the existing components like perception, planning, and control if possible.
- By using FTA (fault tree analysis), clarify what kind of other modules have to work for the feature working expected.
[Planning] trajectory planning to imitate the human driving

Human driving data

Challenge:
Currently, the Autoware uses the rule-based, optimization-based, and sampling-based planner to generate the trajectory. In order to make the planned trajectory close to the human driving, these methods require various kinds of corner cases’ handling and pretty parameters’ tuning in the real environment.

Objectives
- Develop a trajectory planner whose behavior is close to the human driving
  - Examples of the implementation is
    - simple supervised learning
    - imitation learning
    - reinforcement learning

Source: AC photo
**Challenge:**

The weight of the large-size vehicle like busses or trucks changes depending on how many people is riding or how much cargo they have. In that case, the required actuator’s output (e.g. how much the accel pedal should be pushed) will change to meet a desired acceleration.

**Objectives**

Develop a control module that is robust against change in vehicle weight while operation.

- The proposed feature might include:
  - Estimating vehicle weight from control errors
  - Estimating change in flattening of tyres
  - Modifying control parameters used in control algorithms online
- One difficulty is that the weight can be estimated by the target acceleration and actual acceleration, but the converter in the vehicle component from the target acceleration to the actuator’s output may have an error. Therefore, the feature has to judge if the error comes from the converter’s error or the weight’s error.
- For the precise estimation, we can have an assumption that the weight will not change if the vehicle is moving.
- Make it sure that the feature works well in the simple planning simulator.
**Challenge:**

Most of the controllers in Autoware considers only the vehicle kinematics, and the vehicle dynamics such as skidding is not considered yet. Therefore, the Autoware may have a large tracking error on a high speed driving like 80-120 km/h. Additionally, we do not have a simulator with the vehicle dynamics to test it.

**Objectives**

- Develop a longitudinal and lateral controller which can deal with a high speed driving. Also, clarify how to calibrate or tune parameters of the controllers.
- Develop a simulator which can reproduce the vehicle dynamics including the skidding.
- Make it sure that the controller works with a low tracking error in the simulator.
Localization Component
[Localization] Improving Localization in Dynamic Environments with Inconsistent Map

**Challenge:**

Objective:
There are cases where the map data does not match the real-world environment, such as in parking lots or factories that change over time. Although PCD and LL2 data are provided, they may not be reliable. We want to ensure consistency with the existing map while estimating the position in the missing areas using some method.

Assumption:
PCD and LL2 data are available for the planner.

This challenge is to propose a localization system properly estimating the position in the missing areas using some method.

**Ex. Inconsistent Map:**
- Lack of PCDs or lines in map
- Dynamic occlusions
- Major changes in structure

Source: TIER IV
Source: free-materials.com
Source: Google Map The parkhouse Akabane front Aug. 2022
Challenge:

Background: Autoware currently use point cloud matching method to localize its pose. However, it can't detect the suspicious state in the situation where it can be unreliable like the following:
Sample Scenario 1: Vehicle is driving in a tunnel. It might be accurate in lateral direction but not in longitudinal direction.
Sample Scenario 2: Vehicle is driving in an open space. It might be accurate in z/roll/pitch, but not in x/y/yaw

This challenge is to propose an algorithm to detect the unreliability of localization in realtime.

Sample scenario 1:

![Image: Tunnel scenario with speed 46.1 km/h](Image source: TIER IV)
Others
Challenge:

Autoware, being a vast autonomous driving software capable of accommodating various sensor configurations and vehicles, features a multitude of interfaces. Moreover, with numerous contributors involved, these interfaces undergo frequent changes. However, to continue advancing autonomous driving development using Autoware, it is essential to keep track of these changes.

This challenge to propose tools or frameworks for managing various interfaces, such as ROS topic and service names, ROS parameter names, and map specifications changes. For instance, proposals could include mechanisms for detecting interface changes and automatically generating release notes, or suggesting development frameworks for automating the management of interface changes.

Sample of release notes summarizing interface changes

```
20xx-08 Release Notes

API / Interface: New and Changed Items

Topic / Service
- Door API
  - Added new AD API for opening/closing multiple doors.

Parameter
- Initial Pose Parameter
  - Added new Parameter xref_ defined_initial_pose to set initial pose of ego-vehicle

Map specifications
- Road Shoulder Lane
  - Added a new road type to Lanelet2 for representing road shoulder lanes.

Others
- AD API v1.2.0
  - AD API v1.2.0 has been released due to additions/changes to the above API.

Sample of release notes summarizing interface changes
```
Challenge:

AWSIM is one of the default simulators used for Autoware development. However, the simulator still has a lot of challenges for better user experience. These include:

- Less requirements for computation power to run simulation
- Poor UI to customize of vehicle models and environment models
- Unrealistic behavior of NPC vehicles/pedestrians in environments
- Simulation of adverse weather conditions (sensor noise from rains and splash from vehicle driving in front)

Objectives

Propose any ideas to make AWSIM more user-friendly for users with aforementioned examples above.
Challenge:

Autoware, with its wide range of parameters, occasionally experiences unexpected behavior due to combinations of these parameters. We aim to conduct extensive testing across various parameters to ensure robustness.

This challenge is to propose frameworks that efficiently test Autoware’s parameter space. Suggestions may include streamlined testing frameworks utilizing techniques like Pairwise Testing and Factorial Design to enhance testing efficiency.