

Perception

Dynamische Objekte

Welche Objekte sind dynamisch?

- **Fußgänger (pedestrians):** Menschen, die mit einer Art Fahrzeug oder mobilem System laufen oder fahren. Z.B. Fahrräder oder Roller, Skateboards, Pferden, Rollschuhen oder Rollstühlen, etc.
- **Fahrzeuge (Vehicles):** Pkw, Transporter, Lkw, Motorräder, Fahrräder, Busse, Züge
- **Dynamisch (Dynamic):** Elemente, deren Position sich im Laufe der Zeit ändern kann.
z.B. Bewegliche Mülleimer, Buggys, Taschen, Rollstühle, Tiere, etc

Quelle: https://carla.readthedocs.io/en/latest/ref_sensors/

Was muss die Perception dynamischer Objekte alles ermitteln?

- Abstand zum vorausfahrenden Auto
- Abstand zu Autos, die auf der linken und rechten Spur fahren
- Geschwindigkeit der anderen Autos

Wie hat es Gruppe 2 gelöst?

Abstand zum vorausfahrenden Fahrzeug

- **Radar Sensor** misst den Abstand von dem vorausfahrenden Fahrzeug
- Alle Distanzpunkte werden nach ihrer Distanz zum lokalen Pfad gefiltert
- Aus der gefilterten Point Cloud wird der am naheliegendster Punkt zum Ego-Vehicle gewählt
- Der naheliegendster Punkt wird über die Euklidische Distanz bestimmt $\sqrt{\sum_{i=1}^n (q_i - p_i)^2}$
- Die Längenabstand Informationen wird dann über eine Radar/Distance Topic an den steering controller gesendet.

Parameter:

- Maximaler Abstand zum Pfad, der berücksichtigt wird: 0,5 (m)
- Pause, wenn kein Hindernis erkannt wurde: 2,0 (Sek)
- Gesendete Distanz, wenn kein Hindernis erkannt wurde: 100



Wie hat es Gruppe 2 gelöst?

Abstand zu Autos, die auf der linken und rechten Spur fahren

- Zur Berechnung der Entfernung werden zunächst alle vom **Lidar-Sensor** gelieferten Punkte in Kartenkoordinaten (x,y,z) umgewandelt.
- Alle Distanzpunkte werden nach ihrer Entfernung zur jeweiligen Fahrspur gefiltert.
- Aus der gefilterten Point Cloud wird überprüft, ob noch Punkte vorhanden sind.
 - >Falls Ja, befindet sich ein Objekt auf der nebenliegenden Fahrbahn.
 - >Falls Nein, ist die nebenliegende Fahrbahn frei.

Parameter:

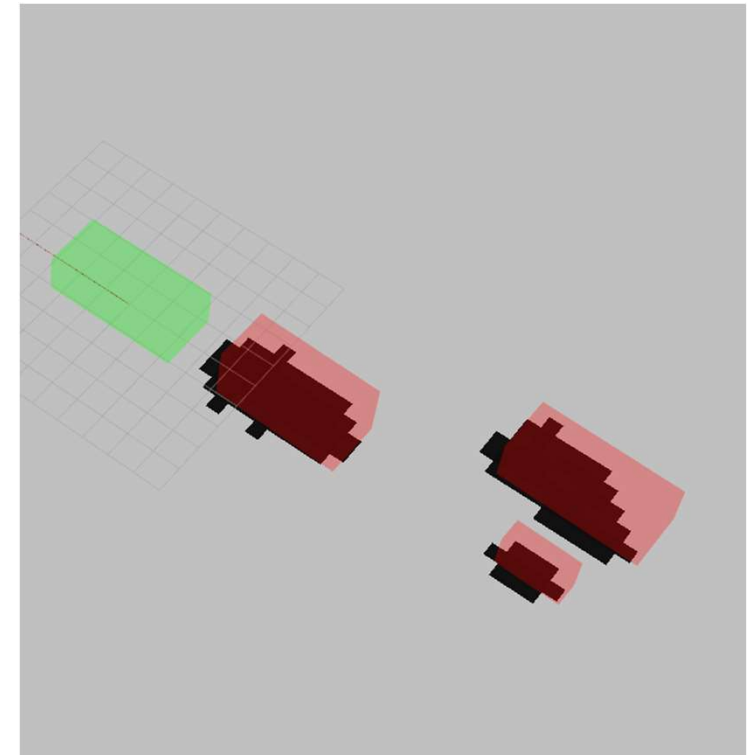
- maximaler Abstand von LIDAR-Sensor-Daten, der berücksichtigt werden: 20



Wie hat es Gruppe 1 gelöst?

Abstand zum vorausfahrenden Fahrzeug

- Über einen **Semantic Lidar-Sensor** (rotierender LIDAR) wird im 360° der Abstand zum Ego-Vehicle gemessen und in einer PointCloud2 gespeichert.
- Mithilfe der Tags des Semantic Lidar können mit den Tag-Konstanten aus der PointCloud Objekte zugeordnet werden und die irrelevanten Objekte wie z.B. Vegetation rausgefiltert werden (in clearing PointCloud verschoben)
- Die relevanten Hindernisse (in marking PointCloud) werden zu einer regelmäßig geupdateten 2D Hinderniskarte (costmap) aufbereitet.

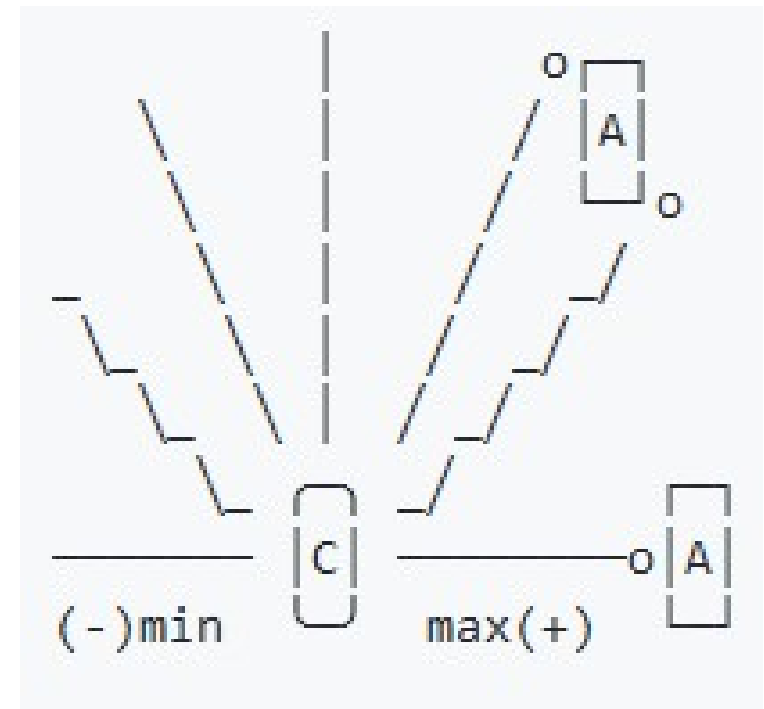


Quelle: https://github.com/ll7/psaf1/tree/master/psaf_ros/psaf_obstacle_layer

Wie hat es Gruppe 1 gelöst?

Abstand zum vorausfahrenden Fahrzeug

- Über einen **Costmap Raytracer** erkennt mittels Raytracing Hindernisse auf der Costmap
- Funktioniert innerhalb eines Teilkreises mit fixen Radius und kleinen Winkeln zwischen den einzelnen Strahlen. Durch niedrige Schrittweite werden andere Fahrzeuge mehrfach abgetastet und so Position, Abstand und Seite zum Ego-Vehicle bestimmt.



Wie hat es Gruppe 1 gelöst?

Abstand zu Autos, die auf der linken und rechten Spur fahren

- Mit Hilfe des Costmap Raytracers wird die linke bzw. rechte Seite des Ego-Vehicle auf Hindernisse geprüft.
- Entsprechend wird ein Teilkreis auf der Seite des Vorgangs (z.B. beim linksüberholen) mittels des Costmap Raytracers auf Hindernisse überprüft.
- Es können auch Bewegungen von anderen Verkehrsteilnehmern erkannt werden, indem zwei vorherige Iterationen der Costmap miteinander verglichen werden, umso festzustellen ob sich die Hindernisse bewegen.



Quelle: <https://www.youtube.com/watch?v=ssYcEJQ6nE> sek.50

Fazit: Vergleich der verwendeten Sensoren

Verwendete Sensoren

psaf 1

- **Kamera RGB**, 3200x1200, 120°, 13 fps
bei x=2m, z=2m
- 3rd person Kamera 20 fps (nicht verw.)
- **Lidar segm.** bei z=3m
-9°=>0°, 24x, 80k pts/s, 20 rot/s, 50m
- **Lidar segm.** bei z=3m
-37°=>-9.5°, 16x, 40k pts/s, 20 rot/s, 50m
- kein Radar
- **Kamera segm.** 1600x600, 120°, 13 fps
- **Kamera depth** 1600x600, 120°, 13 fps
- keine DVS-Kamera
- **GPS** (GNSS) und **IMU** (Accelerometer)
- **Kollision** und **Lane Invasion Sensor**

psaf 2

- **Kamera RGB**, 800x600, 90°, 20 fps
bei x=2m, z=2m
- 3rd person Kamera 20 fps (nicht verw.)
- **Lidar normal** bei z=0.5m
-5°=>1°, 32x, 20k pts/s, 20 rot/s, 50m
- **Lidar segm.** bei z=2.4m
-26.8°=>2°, 32x, 320k pts/s, 20 rot/s, 50m
- **Radar** "vorne" 400x300, 90°, 20 fps
- **Kamera segm.** 600x110, 90°, 20 fps
- **Kamera depth** 400x300, 90°, 20 fps
- **Kamera DVS** 400x70, 90°, 20 fps
- **GPS** (GNSS) und **IMU** (Accelerometer)
- **Kollision** und **Lane Invasion Sensor**

Abbildung von Julin Nowoczyn (Discord)