

1 Calculate Distance to Nearest Obstacle on Lane (10 Points)

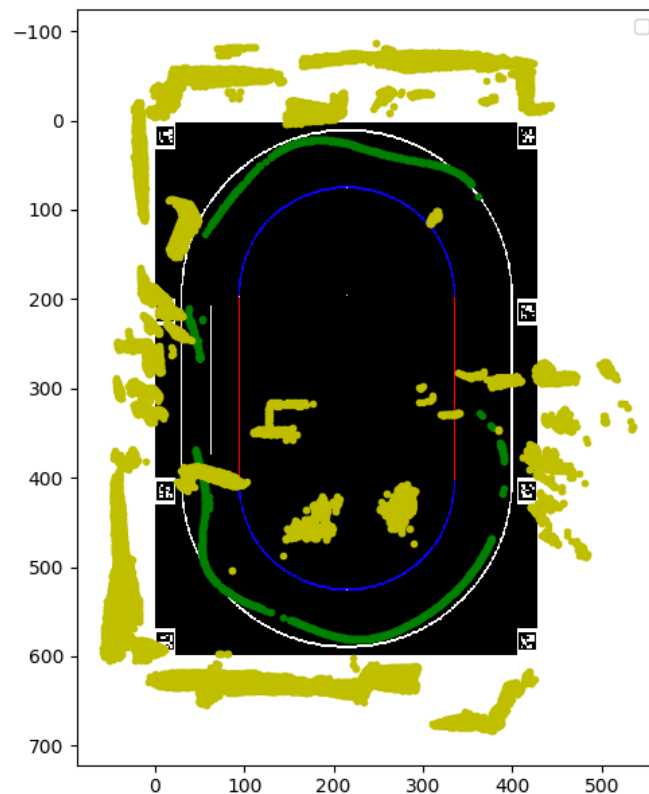


Abbildung 1: Karte mit Lokalisationen des Autos (grün) und erkannte Objekte des LIDARs (gelb).

- Quellcode: https://github.com/bigzed/model_car/blob/version-4.0/texinput/src/localization.py
- Video: https://github.com/bigzed/model_car/blob/version-4.0/texinput/videos/U11_video2.MP4

Der LIDAR erkennt alle Objekte innerhalb von 1.5m vor dem Auto. 'Vor dem Auto' sind hierbei alle Punkte die eine positive X-Koordinate im 'laser'-Koordinatensystem haben. Die LIDAR Daten sind kontinuierlich, allerdings benötigen wir die Daten der Odometrie um sie in ein gemeinsames Koordinatensystem zu bringen. Die Transformation wird durch einen

Wie man an den fehlenden grünen Punkten im Plot erkennen kann, hat das Positionssystem teilweise lange Verzögerungen oder wie in der Fahrt aus der der Plot stammt, kann auch eine komplette Kamera ausfallen. Bei dieser Fahrt war es wohl die mittlere was nach einem Neustart behoben war.

Das Auto ist in dieser Fahrt ungefähr an Koordinate (80,400) los gefahren und hat die Strecke gegen den Uhrzeigersinn befahren. Auf der rechten Gerade erkennt man die Ausweichbewegung um das Hindernis und auf oberen Hälfte der linken Gerade das Stoppen vor dem Hindernis.

Zu beachten ist, das die anscheinend komplette Blockade der Strecke auf der rechten Seite durch ein nachträgliches Verschieben des Hindernisses von der inneren auf die äussere Strecke entstanden ist während das Auto sich schon in der Kurve befand. Der LIDAR-Plot ist an der stelle irreführend.

1. Für die Transformation benutzen wir *tf.transformations* mit *TransformBroadcaster* und *sendTransform* und *TransformListener* sowie *transformPoint*. Die LIDAR Daten werden dann in eine *PointCloud2* umgewandelt und durch die Transformation in das Koordinatensystem der Odomotetrie übertragen.
2. Wir benutzen die Funktion *closest_point()* von Übungsblatt 9 mit einigen Änderungen an der Funktion für die Bereiche im Kreis und einem leichten *look ahead*.
Der Lenkeinschlag wird aus dem Winkeln zwischen Orientierung des Autos und dem Vektor vom Auto zum *closest_point* bestimmt. Diesen Winkel nehmen als Fehler fuer einen KP-Kontroller den Lenkwinkel bestimmt.
3. In unserer *lane_is_free* Funktion wird für jedes erkannte Hindernis des LIDARs getestet ob es auf der derzeitigen Spur liegt, falls nicht behalte die Spur bei.
Falls ein Hindernis auf dieser Spur ist, überprüfe die andere und falls diese frei ist, wechsele die Spur. Sollte diese ebenfalls belegt sein halten wir an, bis eine der Spuren wieder frei wird.