

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

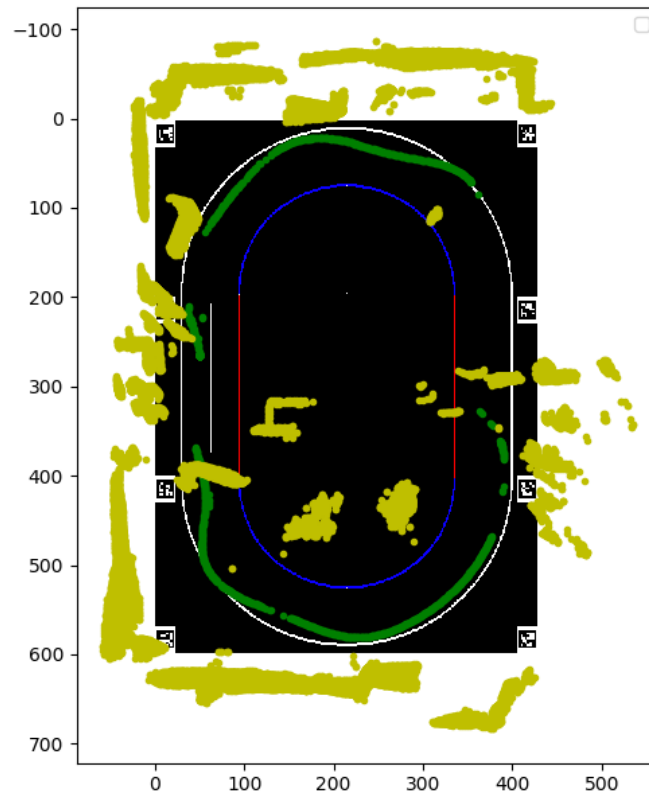


Abbildung 1: Karte mit Lokalisationen des Autos (grün) und erkannte Objekte des lidars (gelb).

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

- Wir benutzen 1.5 m als Distanz für den lidar sensor.
- Da die Odometriedaten von der Lokalisierung mit großer Verzögerung ankamen, gibt es die Lücken wie oben gezeigt auf dem Plot.
- Auf der Grafik erkennt man, dass das Auto auf der linken Linie vor dem obstacle auf zwei Linien zum Halt gekommen war. Auf der rechten Linie kann man sehen, dass dort ein obstacle war, diesem ist das Auto ausgewichen. Auf dem oberen Kreis, war ein weiteres obstacle, diesem ist das Auto ausgewichen und ist rechts rum gefahren.

1. Für die Transformation benutzen wir `tf.transformations` mit `TransformBroadcaster` und `sendTransform` und `TransformListener` sowie `transformPoint`. Dann rechnen wir in die anderen Koordinaten um.
2. Wir benutzen die Funktion `closest point` von Übungsblatt 9 mit einigen Änderungen an der Funktion für die Bereiche im Kreis. Dann berechnen wir einen `closepoint` relativ zum Punkt. Wir behalten einen Einheitsvektor vom Auto und merken uns die Richtung von diesem Richtungsvektor vom Auto. Ein weiterer Vektor zeigt auf den Kreismittelpunkt. Mithilfe der Vektoren zwischen Auto und `closest point` und Auto und Kreismittelpunkt und Einheitsvektor erhalten wir den Winkel.
3. In unserer `laneisfree` Funktion benutzen u. a. die `closest point` Funktion mit Eingabe der Punkte von der `point cloud` und überprüfen somit, ob ein Punkt auf der `lane` liegt.
4. Wir benutzen die `laneisfree` Funktion von 3. und übergeben dieser die `lane` auf der wir gerade fahren. Falls kein `obstacle` gefunden wurde, `publish` die Geschwindigkeit. Falls ein `obstacle` gefunden wurde dann setze die andere `lane` als aktuell. Falls keiner dieser beiden Fälle eintritt, dann ist ein `obstacle` auf beiden `lanes` und wir `publishen` die Geschwindigkeit 0.