

ATF Übung 3 Protokoll

Aufgabe 1

```
angle = float(data["steering_angle"])
steering = angle / 25

if steering > 0.95:
    global inc
    inc = -0.1
elif steering < -0.95:
    inc = 0.1

steering += inc

send_control(steering, 0.3)
```

Implementierung:

alten Lenkwinkel auslesen aus den Daten und umrechnen in Wertebereich [-1, 1], anschließend um 0.1 erhöhen bis 1, danach um 0.1 reduzieren bis -1 und wieder von vorne

Aufgabe 2

Parameter:

Die Spuren werden im Vergleich zur UE02 besser erkannt, da die Spurlinien nun 20% außerhalb der Bildunterkante sein können. Im Bild oben sieht man, dass die rechte Spur nicht auf der Unterseite erkannt schneidet innerhalb des Bildbereiches, und daher ursprünglich nicht erkannt wurde. Nun darf sie 20% außerhalb der rechten unteren Bildecke schneiden und wird richtig erkannt.

Der Hough Threshold, sowie die Thresholds für das erkennen von ähnlichen Linien hat sich gegenüber der alten Übung nicht geändert und funktioniert mit den alten Parametern.

Erklärung der Vorübung:

PARAMETER ERKLÄRUNG:

- `thresh_hough`:
 - nur Linien mit einem Score im Hough Space von $> \text{thresh_hough}$ werden bearbeitet
- `thresh_same_d`:
 - 2 Linien sind ähnlich wenn d-Wert im Bereich $\pm \text{thresh_same_d}$ ist
- `thresh_same_a`:
 - 2 Linien sind ähnlich wenn a-Wert im Bereich $\pm \text{thresh_same_a}$ ist
- `thresh_a`:
 - wichtig für die Erkennung der rechten und linken Spurränder, Winkel der Spurränder müssen im Wertebereich von $60 \pm \text{thresh_a}$ (=links) ODER $120 \pm \text{thresh_a}$ (=rechts) sein

```
@socket.io.on('telemetry')
def telemetry(data):
    if data:
        speed = float(data["speed"])
        angle = float(data["steering_angle"])
        image = Image.open(BytesIO(base64.b64decode(data["image"])))
        image = cv2.cvtColor(np.array(image), cv2.COLOR_RGB2BGR)

        thresh_hough = 40 # 40 80
        thresh_same_d = 50 # 50 20
        thresh_same_a = 15 # 15 15
        thresh_a = 15 # 25 20

        [image, left_x, right_x] = houghUE2(image, thresh_hough, thresh_same_d, thresh_same_a, thresh_a)
        print("L: ", left_x, "\nR: ", right_x)
        cv2.imshow("Result", image)
        cv2.waitKey(30)
```



Aufgabe 3

Implementierung:

Die Geschwindigkeit wird bei 15mph gehalten indem geprüft wird, ob dass Fahrzeug schneller als 15mph ist, und sobald dies eintritt, wird die Beschleunigung auf 0 gesetzt, andernfalls auf 0.5.

Für den Lenkwinkel haben wir uns folgendes gedacht: Der Winkel soll sich im Wertebereich $[-0.15; 0.15]$ befinden. Von der houghUE2 Funktion bekommen wir das image zurück, und zusätzlich noch left_x und right_x. Diese Werte bestimmen, wie weit sich die linke bzw. rechte Spurlinie von der Mittellinie befindet. Für das Ausrechnen des Lenkwinkels haben wir $(\text{links-rechts}) / -800$ gerechnet.

800 hat nur den Sinn, dass der Lenkwinkel irgendwo im Bereich zwischen -0.15 und 0.15 ist. Sollte der Wert außerhalb sein, wird dies verhindert durch die Funktion np.clip.

Sollte eine Spurlinie nicht erkannt werden, ist left_x oder right_x gleich 0 und somit würde die obige Berechnung nicht mehr funktionieren. Daher wird in diesem Falle einfach der vorherige Lenkwinkel benutzt. Dieses Szenario trifft im Grunde fast nie ein bei uns.

Parameter:

Die Erklärung befindet sich weiter oben, folgende Parameter wurden für die selbst implementierte Hough Funktion verwendet:

```
thresh_hough = 40
thresh_same_d = 50
thresh_same_a = 15
thresh_a = 15
```

```
if data:
    speed = float(data["speed"])
    angle = float(data["steering_angle"])
    image = Image.open(BytesIO(base64.b64decode(data["image"])))
    image = cv2.cvtColor(np.array(image), cv2.COLOR_RGB2BGR)

    thresh_hough = 40
    thresh_same_d = 50
    thresh_same_a = 15
    thresh_a = 15

    [image, left_x, right_x] = houghUE2(image, thresh_hough, thresh_same_d, thresh_same_a, thresh_a)
    #[image, left_x, right_x] = hough(image, 70, thresh_same_d, thresh_same_a, thresh_a)

    # STEERING
    buffer = left_x - right_x
    steering_range = 0.15
    adapt_steering_faktor = -800

    # one or more lines not detected, instead of calculating steering, use the old angle
    if left_x == 0 or right_x == 0:
        print("\nONE LINE NOT DETECTED\n")
        steering = angle / 25
    else:
        steering = np.clip(buffer / adapt_steering_faktor, -steering_range, steering_range)

    # THROTTLE
    throttle = 0.5
    if speed > 15:
        throttle = 0
```

Aufgabe 4

Eigentlich konnten wir die eigene Implementierung komplett verwenden. Da bei Kurven, die das Auto fährt die Spurlinien aber oftmals nicht den unteren Bildschirmrand schneiden, mussten wir auch Linien zum Spurhalten hernehmen, die links und rechts bis zu 20 Prozent weiter außerhalb den unteren Rand schneiden.

Parameter:

Die Erklärung befindet sich weiter oben, folgende Parameter wurden für die selbst implementierte Hough Funktion verwendet:

```
thresh_hough = 70  
thresh_same_d = 50  
thresh_same_a = 15  
thresh_a = 15
```