

# Protokoll ue02

## Aufgabe 1

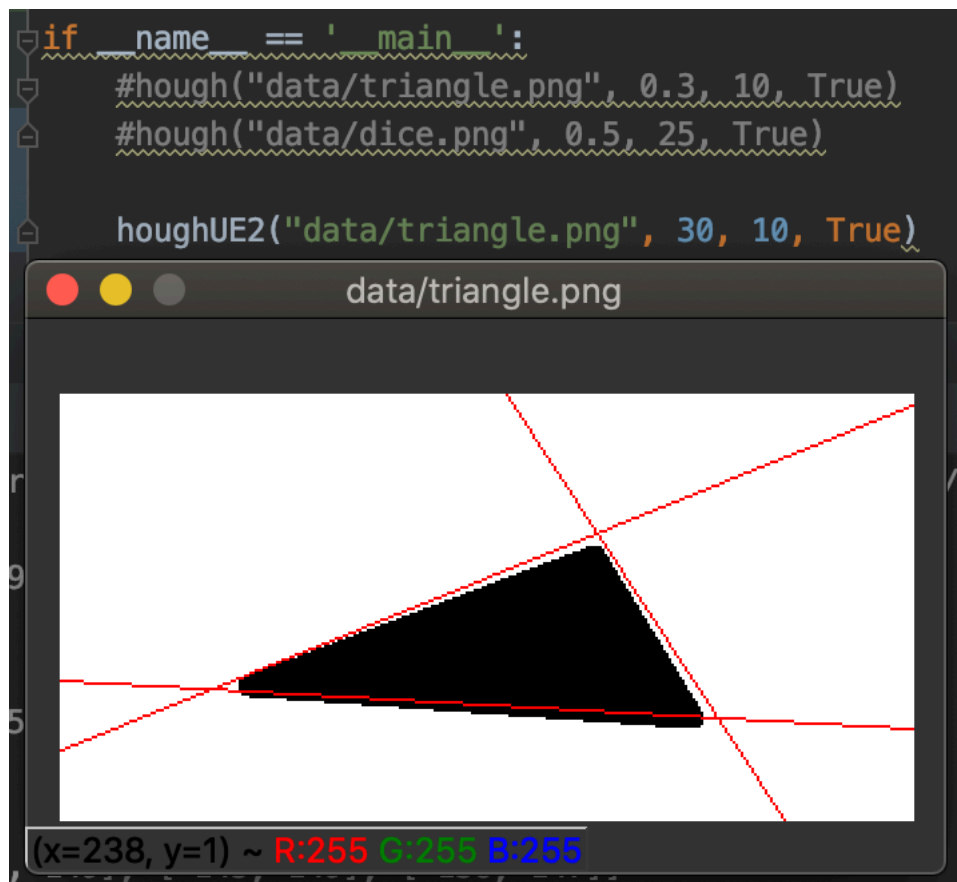
Funktion: houghUE2

threshold für cv2.HoughLines Funktion = 30

threshold für Erkennung von ähnliche Linien = 10

- wenn bei 2 Linien, d UND theta im Intervall  $\pm 10$  voneinander entfernt sind, werden sie als gleiche Linien genommen

Beispielsbild: selbe Funktionalität wie in Aufgabe 1, die Funktion wurde aber im Laufe der Übung angepasst um die rechte und linke Spurlinie zu erkennen! Wird also nicht mehr das selbe Ergebnis liefern wie im Beispielsbild!



## Aufgabe 2 und 3



Bild wurde skaliert auf (640x360)

Maske wurde erstellt (data/mask.png) um unwichtige Bereiche zu entfernen

Um die Fahrbahnränder zu erkennen wurde folgende Überlegung getroffen:

- die Spurlinien müssen den unteren Bildbereich schneiden (rechte Spurlinie darf 20% außerhalb des Bildes schneiden, da einige erkannten Linien sehr genau in die untere rechte Ecke gehen und daher hin und wieder eine Kante außerhalb des unteren Bildbereichs wäre)
- es darf nur eine rechte und eine linke Spurlinie geben
- die rechte Spurlinie muss einen Winkel zwischen  $120 \pm \text{threshold}$  haben
- die linke Spurlinie muss einen Winkel zwischen  $60 \pm \text{threshold}$  haben
- threshold, welcher gut funktioniert: 25

Laterale Position:

- L = vom Bildmittelpunkt bis zur linken Linie (welche unten das Bild kreuzt) in Pixel
- R = vom Bildmittelpunkt bis zur rechten Linie (welche unten das Bild kreuzt) in Pixel

Die Videos wurden aus dem Data Ordner entfernt aufgrund der Größe, fürs Testen müssten diese in den Data Ordner wieder kopiert werden!

## Aufgabe 4

Video: 4.mov

PARAMETER ERKLÄRUNG:

- thresh\_hough:
  - nur Linien mit einem Score im Hough Space von >thresh\_hough werden bearbeitet
- thresh\_same\_d:
  - 2 Linien sind ähnlich wenn d-Wert im Bereich +- thresh\_same\_d ist
- thresh\_same\_a:
  - 2 Linien sind ähnlich wenn a-Wert im Bereich +- thresh\_same\_a ist
- thresh\_a:
  - wichtig für die Erkennung der rechten und linken Spurränder, Winkel der Spurränder müssen im Wertebereich von 60 +- thresh\_a (=links) ODER 120 +- thresh\_a (=rechts) sein

main:

```
thresh_hough = 40
thresh_same_d = 20
thresh_same_a = 15
thresh_a = 25
video(thresh_hough, thresh_same_d, thresh_same_a, thresh_a)
```

in der Funktion "video" muss folgende Zeile benutzt werden:

```
img = houghUE2(...)
```

## Aufgabe 5

Video: 5.mov

main:

```
thresh_hough = 80
thresh_same_d = 50
thresh_same_a = 15
thresh_a = 20
video(thresh_hough, thresh_same_d, thresh_same_a, thresh_a)
```

in der Funktion "video" muss folgende Zeile benutzt werden:

```
img = hough(...)
```

Optimierungen:

- Sortierung der Linien absteigend nach Hough Score (Result: 1 Schleife weniger)
- Berechnung von x, y wo canny != 0, x und y sind nun arrays (Result: 1 Schleife weniger)

```
buff = np.argwhere(canny != 0)
```

```
x = buff[:, 1]
```

```
y = buff[:, 0]
```

- Mithilfe von np.bincount Hough Score richtig erhöht

```
d = x * np.cos(angle_rad) + y * np.sin(angle_rad)
```

```
test = np.bincount(d)
```

```
H[d, angle] = H[d, angle] + test[d]
```

Berechnung für HoughLine nurmehr eine Schleife für die Winkel:

```
for angle in range(0, max_angle, step_angle):
```

Mit den oben gewählten Parametern (wichtigster: thresh\_hough, welcher die Anzahl der Linien beeinflusst), läuft das Skript FAST in Echtzeit (siehe Video)

Nachteil: rechte Spur wird seltener erkannt aufgrund des höheren thresholds