THM TECHNISCHE HOCHSCHULE MITTELHESSEN | CAMPUS GIESSEN | MNI Mathematik, Naturwissenschaften und Informatik

II2021 Entwicklung eines autonomen Fahrzeugs

Team 2

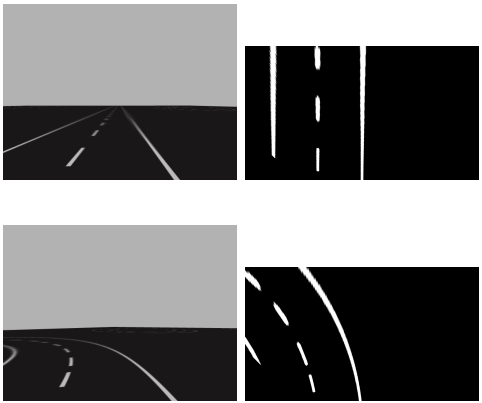
Florian Maximilian Dörr,
Hendrik Wagner

Technische Hochschule Mittelhessen

13. Juli 2022

Inhalt

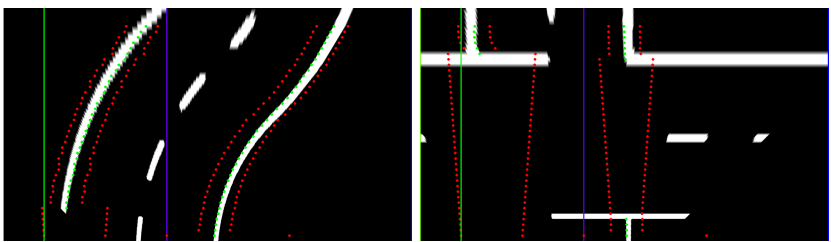
Bildtransformation



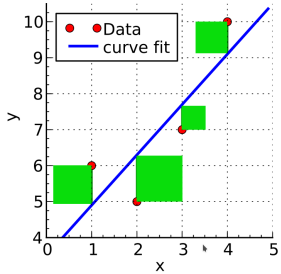
1. Zuschnitt des Inputs auf die Region of Interest
2. Ermittlung Kameraposition, FOV (*Homographie*) für Testtransformationsmatrix
3. Perspektivische Transformation mithilfe Zieltransformationsmatrix

Linienerkennung — „Regions of interest“

- Aufteilung des tranformierten Bildes in zwei Regionen
- Start in Bereich, in dem Anfang der Linie erwartet wird
- Es wird in festen Schritten, Punkte der Linie abgegangen → Auflösung frei wählbar (40 Punkte)
- Weiteres vorgehen abhängig, ob ein ein weißer Pixel gefunden wurde oder nicht:
  1. Weißen Pixel gefunden: Punkt wird in Liste abgespeichert und nächster Suchbereich wird nach diesem Punkt in Verbindung mit dem letzen Punkt (Steigungstendenz) bestimmt
  2. Kein Weißer Pixel gefunden: Der nächste Suchbereich ist der jetzige, er wird jedoch erweitert/aufgefächert



Linienerkennung — Steigungsermittlung



- Bildung einer Gerade über die ermittelten Punkte mithilfe der *Methode der kleinsten (Fehler-) Quadrate*
  - Sucht eine Gerade, dessen quadrierter Distanz zu allen Punkten minimal ist
  - Resultiert in einen Steigungswert m, welcher für den Lenkwinkel und der Geschwindigkeit verwendet werden kann
  - Dank der Transformation ist die Soll-Steigung $m = 0$!

Inhalt

PID — Lenkung

- Lenkung wird mittels zwei PID-Reglern gesteuert
  1. Kurven-Lenkung: *m* Wert wird als Eingabe verwendet
  2. Positionierung innerhalb der Fahrspur: Es existiert ein Sollwert für den ersten erkannten weißen Pixel, der eingehalten werden soll
- Es werden nur 50% der erkannten Punkte und die resultierende Steigung verwendet
  ↳ Es ist hinderlich für die Lenkung zu weit in die „Zukunft“ zu schauen
- Werte werden auf $45°$ bzw. $-45°$ beschränkt
- PIDs arbeiten „gegeneinander“ $\rightarrow$ Kurven-Lenkung ist der *dominante* Wert, Positionierung *korrigierender* Wert

PID — Geschwindigkeit

- PID-Regler für Geschwindigkeit in Abhängigkeit von vier Faktoren:
  1. Die Steigung der ersten 50% der erkannten Punkte (linke und rechte Linie)
  2. Die Steigung der hinteren 50% der erkannten Punkte (linke und rechte Linie) → hier ist eine Vorausschau sinnvoll und notwendig
- Mindestens 2.55 $\frac{m}{s}$ (9.18 $\frac{km}{h}$), maximal 5.4 $\frac{m}{s}$ (19.44 $\frac{km}{h}$) im normalen Modus
  ↳ Drosselung bei Überholmanöver (maximal 2.7 $\frac{m}{s}$)

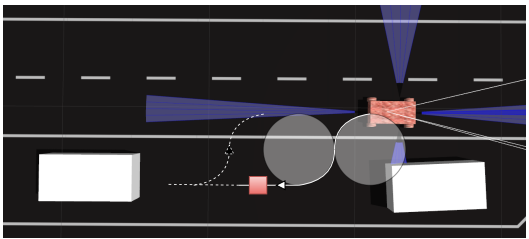
Überholmanöver

- Drosselung der Geschwindigkeit $3m$ vor einem Hindernis
- Spurwechsel $1.6m$ vor dem Hindernis
  ↳ Soll-Werte für Lenkung-PIDs wechseln auf linke Linie
- Spurwechsel sobald rechts neben dem Auto keine Box detektiert wird
  ↳ Soll-Werte für Lenkung-PIDs wechseln zurück auf die rechte Linie
- Problem: Region-of-Interests stimmen nicht mehr!
  ↳ Verschieben der Region-of-Interests nach links nach erstem Wechsel und nach dem Zurückfahren wieder nach rechts

Einparkmanöver

1. Parklücke suchen: Rechter Sensor sucht zwei Hindernisse in Folge
2. Beim zweiten Hindernis abbremsen und Einparksequenz starten
   - 2.1 Festgelegte Einlenkungswinkel beim Rückwärtsfahren
   - 2.2 Optimierung der Position zwischen den Hindernissen $\rightarrow$ mittig
3. Nach kurzer Parkzeit wird die Ausparksequenz gestartet
   - 3.1 Zurückfahren bis zum Wunschabstand zum hinteren Hindernis
   - 3.2 Festgelegtes Ausparkverhalten und Reaktivierung des autonomen Fahrens

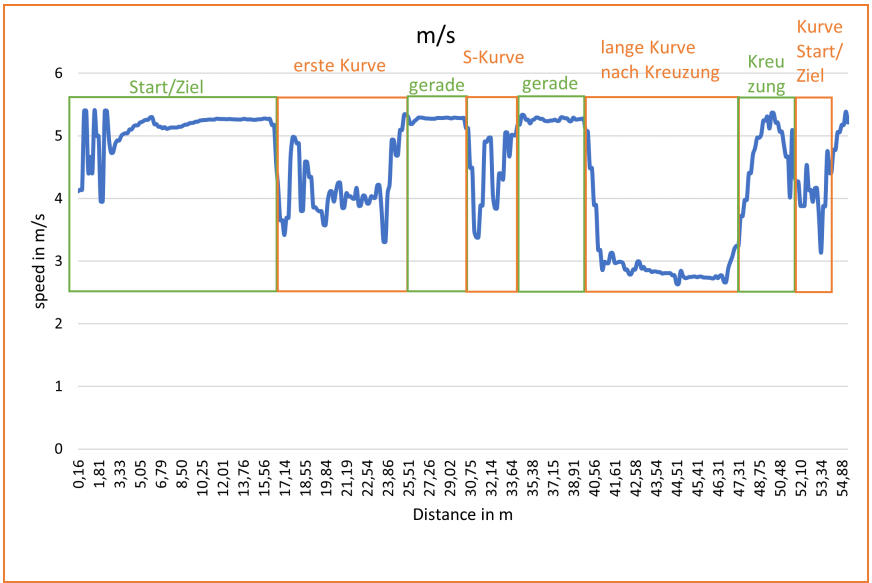
Abbildung: Skizziertes Einparkmanöver



„State-Machine“ & Remote Controller

- Grundkonzept: Stack — durch *pop* von einem Stack in den nächsten
- Zustände werden mithilfe eines Enums definiert
- Initialbelegung des Stacks mit allen gewünschten Fahrabschnitten, die sich ggf. selbst entfernen können
- Kommunikation mit State-Machine mithilfe vom Remote Controller
  - `w, a, s, d`: Manuelles Fahren
  - `q` Remote Controller wird de/aktiviert, indem der entsprechende State oben drauf gelegt wird bzw. entfernt wird
  - `r` Zurücksetzen der PID-Werte
  - `x` Zurücksetzen der Zustandsmaschine durch *clear* und erneute Belegung des Stacks
  - `p` Entfernung des obersten Zustands des Stacks

Speed Graph


m/s
Start/Ziel
erste Kurve
gerade
S-Kurve
gerade
lange Kurve
nach Kreuzung
Kreu
zung
Kurve
Start/
Ziel
speed in m/s
0
1
2
3
4
5
6
0,16
1,81
3,33
5,05
6,79
8,50
10,25
12,01
13,76
15,56
17,14
18,55
19,84
21,19
22,54
23,86
25,51
27,26
29,02
30,75
32,14
33,64
35,38
37,15
38,91
40,56
41,61
42,58
43,54
44,51
45,41
46,31
47,31
48,75
50,48
52,10
53,34
54,88
Distance in m

Quellen

- *Least Squared Graph*: Krishnavedala, CC BY-SA 3.0 via Wikimedia Commons, modifiziert (Quadrate)