

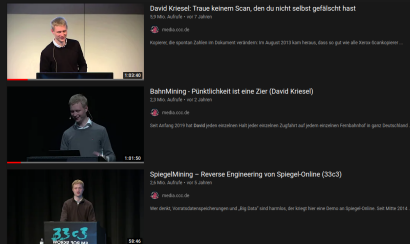
Entwicklung eines autonomen Fahrzeugs

Sven Thomas & Maximilian Biebl

Technische Hochschule Mittelhessen

Anmerkung

Präsentationsstil/Gestaltung soll an David Kriesel angelehnt sein.
Seine Vorträge haben meiner Meinung nach eine gute Balance
zwischen Humor und Sachlichkeit.



Wer ihn nicht kennt: angucken!

Inhalt

- 1 Linienerkennung und Lenkung
 - Erste „naive“ Idee und Probleme
 - Weiterentwicklung der ersten Idee
 - Lenkwinkelbestimmung
- 2 Überholmanöver
 - Einleitung des Manövers
 - 5 Phasen des Manövers
- 3 Parken
 - Finden der Parklücke
 - Einparken
- 4 Geschwindigkeitsregler

Erste dumme „naive“ Idee

Idee: Aus den Vektoren „Druschnittsvektor“ bilden und anhand dessen Lenkwinkel bestimmen

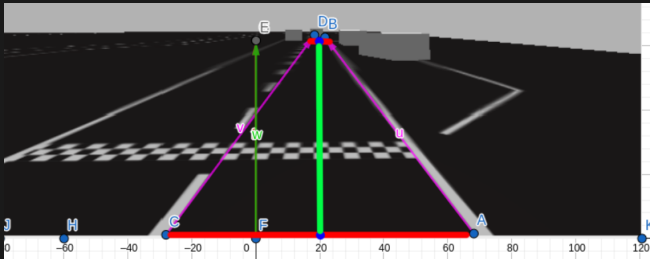


Abbildung: Erster Gedanke

Problem

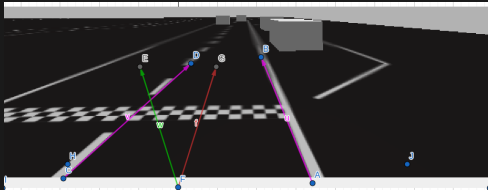


Abbildung: wenn zu weit rechts

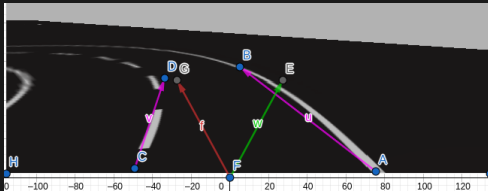
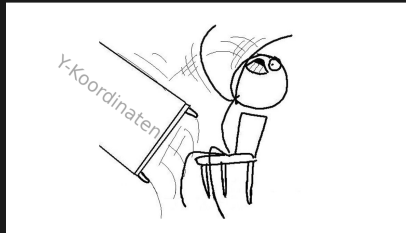


Abbildung: in der Kurve

Wer brauch Y?



- Erkenntnis: wir brauchen nur X-Koordinaten um Mitte der Linien zu bestimmen
- Anhand des „Durchschnitts-X“ und der Bildmitte/Automitte wissen wir in welche Richtung wir müssen.
- Wie weiter die beiden X-Koordinaten getrennt sind um so stärker müssen wir lenken

Probleme mit Mittellinie



- Houghline zu empfindlich \Rightarrow zuviel „Beifang“
- Houghline zu grob \Rightarrow Probleme bei Kurven
- einfach Außenlinie nehmen und $1,25\times$ des Durchschnitts-X als Soll-Fahrbahn

”Discopumper-Algorithmus” für Region-of-Interest

- RoI bekommt festen Startbereich
- wir nehmen erstmal alles was wir bekommen an X-Koordinaten
- wenn wir nichts finden müssen wir breiter werden
- wenn wir immer noch nichts finden nehmen wir das letzte was wir hatten

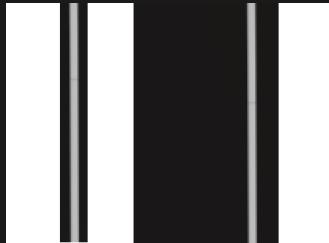
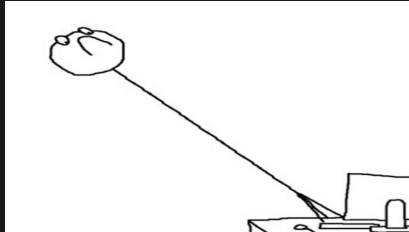


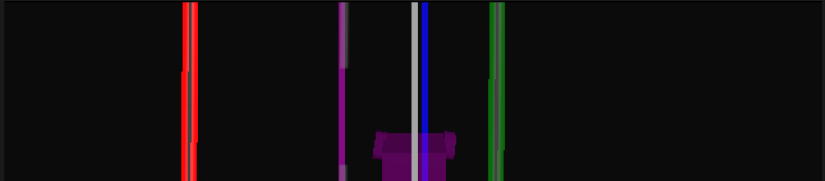
Abbildung: Unterschiedlich große roi

Verbesserung durch Top-Down-View



- Optimierung durch Top-Down-View
- Kamera schaut senkrecht vor dem Auto nach unten
- könnte realistischer werden, durch Bildtransformation zu einem "pseudo" Top-Down

Top-Down-View

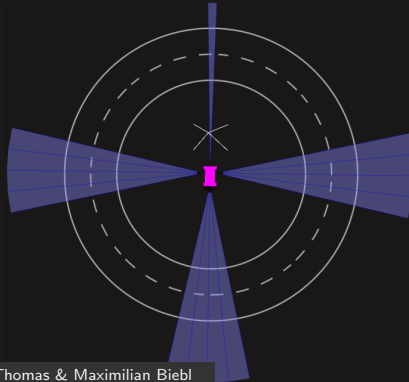


- linke Linie
- rechte Linie
- errechnete Mittellinie
- Bildmitte \Leftrightarrow Automitte
- 1.25-Fache der Mittellinie \Rightarrow Soll-Fahrbahn

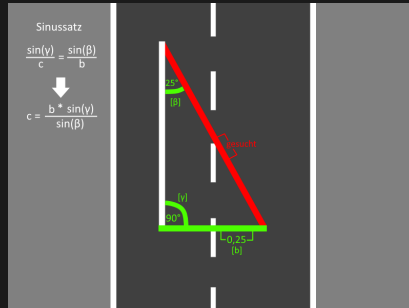
$\Rightarrow \text{Lenkwinkel} = \text{SollFahrbahn}_x - \text{Bildmitte}_x$

Einleitung des Manövers

- schmaler ToF-Sensor an Front
- erkennt an Schwellenwert ob ein Überholmanöver nötig ist
- links breiter ToF für 'Schulterblick'
- Schulterblick ok \Rightarrow starte mit Phase 1 und blockiere zunächst die linienbasierte Lenkung



1. Phase Spurwechsel



- Spurwechsel sei vereinfacht eine Gerade
- Überholgerade
- Spurwechseldistanz
- β = Lenkwinkel

aus Strecke c und Geschwindigkeit \Rightarrow Dauer des Spurwechsels

2.Phase Rol-Wechsel + Warten auf die zu überholende Box

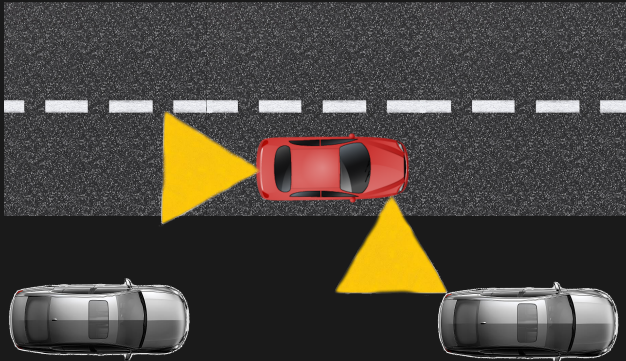
- versetzen der Rol für Lane Detection
- setzen eines neuen Faktors für Soll-Fahrbahn
- starte wieder die linienbasierte Lenkung
- warten bis rechts die Box ist

Phasen 3 bis 5

- ③ Wagen ist auf der linken Spur und wartet bis er an der Box vorbei ist
- ④ Spiegelverkehrt zu Phase 2 wechsel der Rol
- ⑤ Spiegelverkehrt zu Phase 1 Spurwechsel zurück

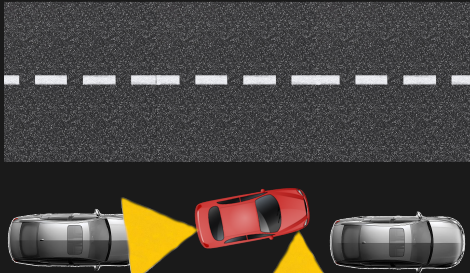
Erkennen einer Parklücke in 4 Phasen

- 1 Erkennen der ersten Box
- 2 Erkennen der Lücke
- 3 Erkennung der zweiten Box
- 4 Einleiten des Parkmanövers



Einparken in 3 Phasen

- 1 Führt noch ein Stück grade aus und führt anschließen rückwärts ein Spurwechsel durch, gleich zum Überholmanöver
- 2 In der Parklücke \Rightarrow fährt Rückwärts bis hinterer ToF Schwellenwert erreicht.
- 3 Ausparken \Rightarrow Spurwechsel vorwärts und anschließend übergang in normalen Modus



Geschwindigkeitsregler

sehr primitiv:

- nur außerhalb von Manövern aktiv
- bestimmung anhand Lenkwinkel
- Lenkwinkel kleiner als Schwellwert \Rightarrow beschleunigen
- nach Delay immernoch kleiner als Schwellwert \Rightarrow weiter beschleunigen
- Schwellwert überschritten \Rightarrow zurück auf Minimalgeschwindigkeit

DANKE FÜR IHRE AUFMERKSAMKEIT!