

Digitale Bildverarbeitung

DHBW Stuttgart, Vorlesung „Computergraphik und Bildverarbeitung“

Praktische Übung

Projekte in dieser Vorlesung

Automotive

Spurerkennung



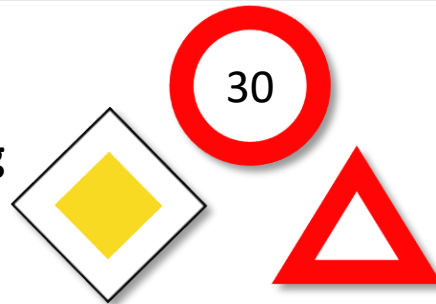
Farbräume

Bildanalyse (Morphologische Verfahren, Merkmalsextraktion, Kanten- und Flächenbestimmung)

Histogramme

Segmentierung

Verkehrszeichenerkennung



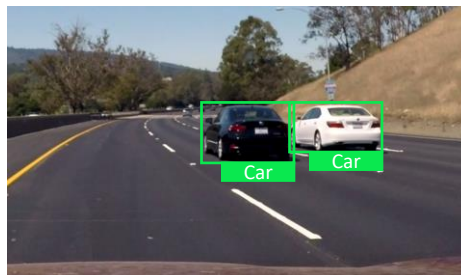
Farbräume

Kontrastverstärkung

Histogramme

Klassifizierung, Objekterkennung

Objekterkennung



Operationen im Ortsbereich (lokale Operatoren, Faltungsfilter)

Segmentierung

Optischer Fluss

Klassifizierung, Objekterkennung

Projektübersicht: Spurerkennung

Spurerkennung



Farbräume

Histogramme

Bildanalyse (Morphologische
Verfahren, Merkmalsextraktion,
Kanten- und Flächenbestimmung)

Segmentierung

- Auteilung der Note
 - 70 % Programm inkl. Quellcodes und Kommentare
 - Kommentare müssen sinnvoll eingesetzt werden
 - bedeutungsvolle Namensgebung für Variablen
 - gern Python-Bibliothek Sphinx verwenden (<https://www.sphinx-doc.org/>)
 - 30 % Dokumentation (mögliche Formate Word, Powerpoint, Markdown)
 - Gewählte Vorgehensweise
 - Warum haben Sie sich für die gewählte Vorgehensweise entschieden?
 - Welche Alternativen gab es?
 - Dokumentation und Diskussion der Ergebnisse
 - Welche Probleme traten auf? Welche Lösungswege haben Sie verfolgt?
 - Lessons Learned
 - Was nehmen Sie aus dem Projekt für sich mit?
 - Ausblick
 - Welche Probleme konnten Sie im Rahmen des Projektes nicht behandeln?

Projektübersicht: Spurerkennung

Spurerkennung



Farbräume

Histogramme

Bildanalyse (Morphologische
Verfahren, Merkmalsextraktion,
Kanten- und Flächenbestimmung)

Segmentierung

Mindestanforderungen (entspricht der Note 2,0)

- **Segmentierung des Bildes:** schränken Sie das Bild auf den Bereich ein, in dem sich die Spurmarkierungen befinden
- **Vorverarbeitung:** Führen Sie eine Kamerakalibrierung (für Udacity-Bildquellen) und die Perspektivtransformation durch
- **Farbräume, Histogramme:** erkennen Sie die Spurmarkierungen in den Farben der angegebenen Quellen
Falls weitere Spurmarkierungen auf dem Bild gefunden werden, müssen die der eigenen Fahrspur priorisiert werden
- **Allgemeines:** Die Verarbeitung von Bildern muss in Echtzeit stattfinden --> Ziel: > 20 FPS
- **Allgemeines:** Beschleunigen Sie die Verarbeitung durch weitere Maßnahmen (bspw. Erkennung der Spurmarkierung in den ersten Frames, Tracking der Spurmarkierung in weiteren Frames solange, bis sich Spurmarkierungspositionen zu stark ändern) → mind. eine Maßnahme im Projekt verwenden
- **Curve / Polynom Fitting:** Erkennen Sie die Krümmung der Fahrspur und geben Sie diese im Ausgabebild aus
- **Allgemeines:** relevante Spurmarkierungen werden in den Udacity-Bildern und im Video „project_video“ durchgehend erkannt

Projektübersicht: Spurerkennung

Spurerkennung



Farbräume

Histogramme

Bildanalyse (Morphologische
Verfahren, Merkmalsextraktion,
Kanten- und Flächenbestimmung)

Segmentierung

Zusatzaufgaben (Mindestanforderungen + 2x Zusatzaufgaben = 1,3)

- relevante Spurmarkierungen werden im Video "challenge_video" und "harder_challenge_video" (nahezu) durchgehend erkannt
- relevante Spurmarkierungen werden auf den Datensatz KITTI angewendet. Welche Anpassungen müssen vorgenommen werden, damit Ihr Algorithmus übertragen werden kann?
- erkennen Sie Objekte im Bild und visualisieren Sie diese (z.B. weitere Fahrzeuge, Motorräder, etc.)
Die Objekterkennung bitte so implementieren, dass sie deaktivierbar ist und nicht in FPS-Berechnung einzahlt.
- nutzen Sie alternative Möglichkeiten der Spurerkennung (z.B. mit Neuronalen Netzen)
- ergänzen Sie Ihren Algorithmus um eine Kennzeichenerkennung
- Gerne können Sie eigene Zusatzaufgaben zur Verbesserung Ihres Algorithmus einführen. (Aufwand sollte vergleichbar sein zu o.g. Punkten).
- **Alle durchgeführten Aufgaben müssen dokumentiert, kommentiert und abgegeben werden.**

Projektübersicht: Spurerkennung

Spurerkennung



Farbräume

Histogramme

Bildanalyse (Morphologische
Verfahren, Merkmalsextraktion,
Kanten- und Flächenbestimmung)

Segmentierung

Zusatzaufgabe Android (Mindestanforderungen + 2x Zusatzaufgaben + Android-Portierung = 1,0)

- entwickelter Algorithmus wurde auf Android übertragen
- Dokumentation der erkannten Fahrspuren
- Diskussion über die Herausforderungen bei der Portierung der Python-Umsetzung mit der Java-Umsetzung
- **Alle durchgeführten Aufgaben müssen dokumentiert, kommentiert und abgegeben werden.**

Projektübersicht: Spurerkennung

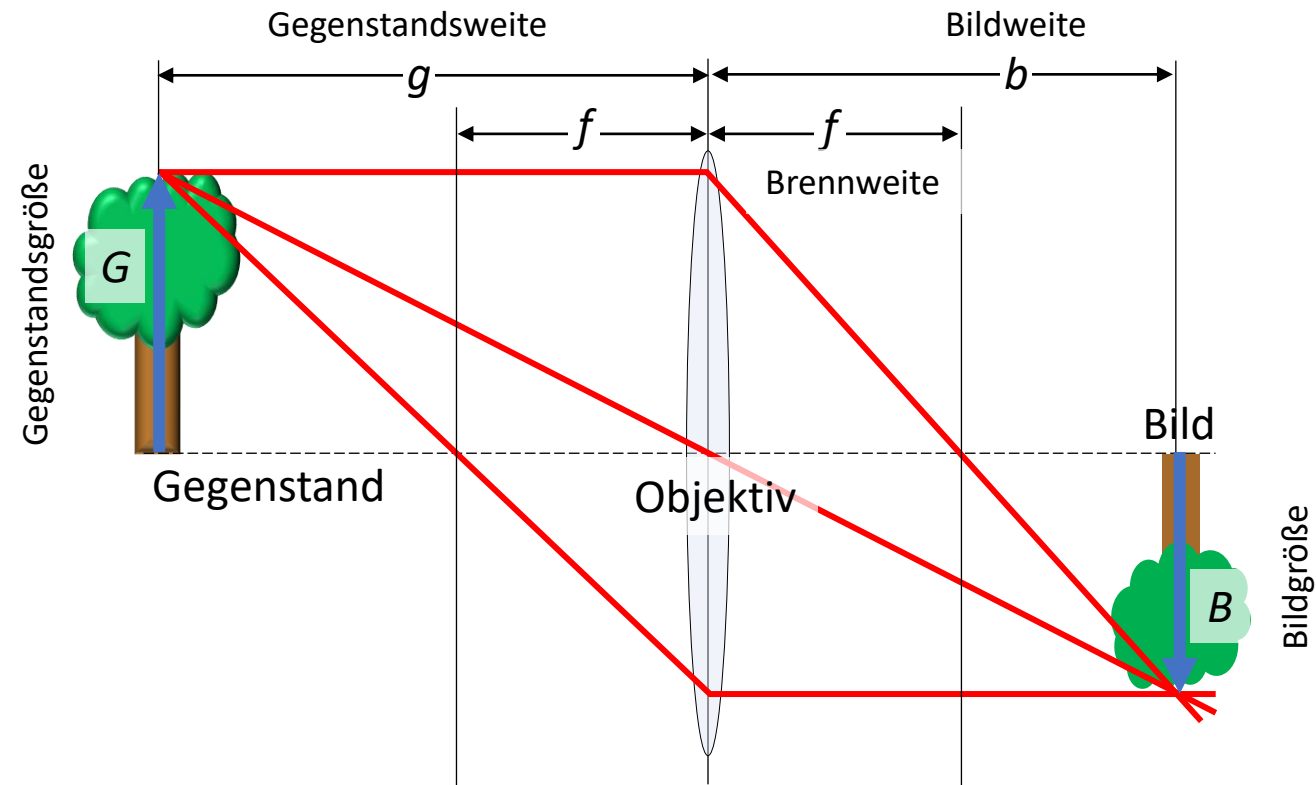
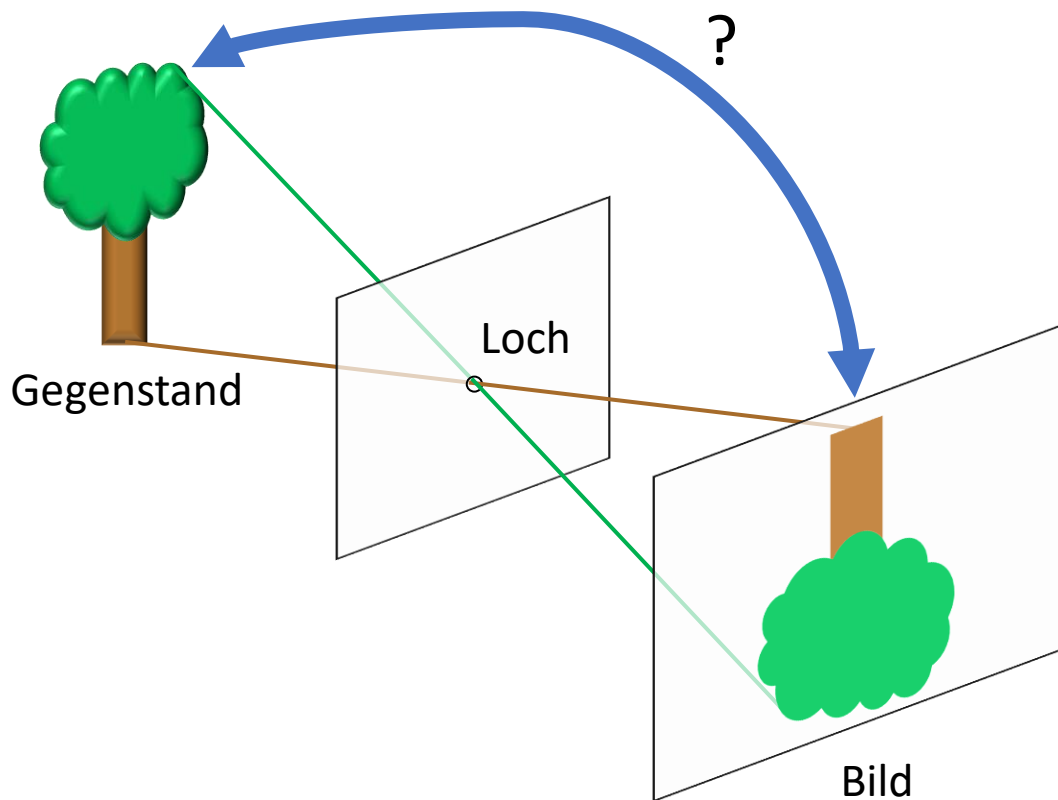
- Erwartete Abgabe
 - Quellcodes inkl. Kommentare
 - Jupyter Notebook o.ä. zur prototypischen Implementierung
 - Python Quellcode zur performanten Implementierung
 - Android Studio Quellcode
 - Bilder und Videos inkl. erkannter Linien und Objekte
 - Dokumentation
 - Word, Powerpoint, Markdown

Projektübersicht: Spurerkennung

- 12x Gruppen, Vorstellung am 12.12. pro Gruppe:
 - 7 Minuten Projektvorstellung (Live-Vorstellung, Methoden, Vorgehen, Probleme)
 - 3 Minuten offene Fragen aus der Gruppe
- Visualisierung in Zeitmessung einfließen lassen

Exkurs 1: Kamerakalibrierung

siehe eigene PDF

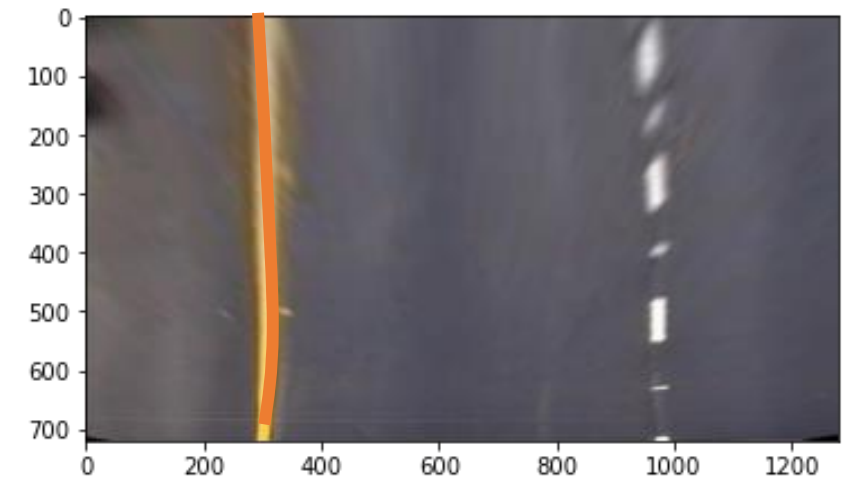


Exkurs 2: Perspektivtransformation

- Krümmung der Linien im ursprünglichen Kamerabild entspricht nicht der realen Fahrspurkrümmung
→ Perspektivtransformation in Vogelperspektive

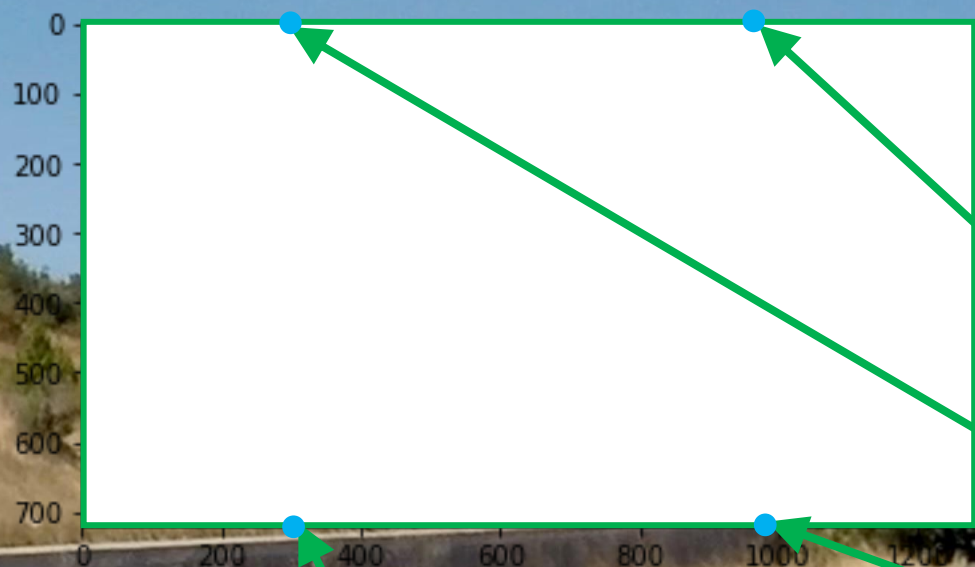


$$\begin{pmatrix} t_i u_i' \\ t_i v_i' \\ t_i \end{pmatrix} = A \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix}$$





Perspektivtransformation



1. Ermitteln Sie 4x Punkte auf dem Bild, die nach der Transformation in einer rechteckigen Beziehung zueinander stehen
2. Definieren Sie die 4x Zielpunkte, an denen die zu transformierenden Punkte nach der Projektion sein sollen

| ID | Original | | Transformiert | |
|----|----------|-----|---------------|-----|
| 1 | 598 | 448 | 300 | 0 |
| 2 | 684 | 448 | 980 | 0 |
| 3 | 1026 | 668 | 980 | 720 |
| 4 | 278 | 668 | 300 | 720 |