

GALAXIS



Unterstützt durch

BASLER
the power of sight

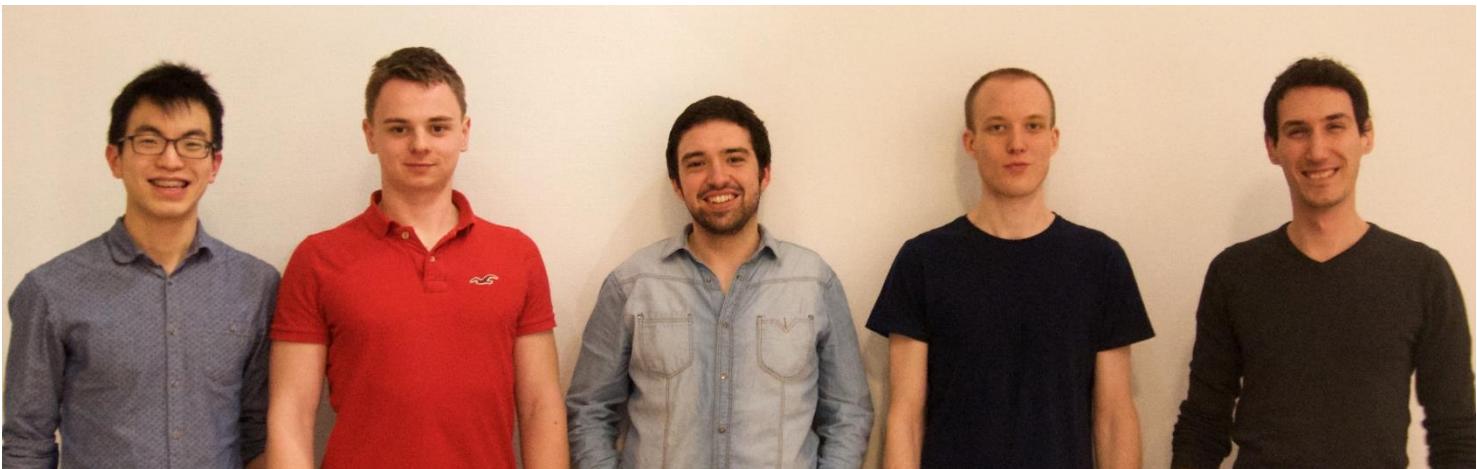
ika


Informatik 11
Embedded Software

RWTHAACHEN
UNIVERSITY

Team GalaXIs

- ▶ Gegründet 2007, zuerst partizipiert 2008
- ▶ 5 Mitglieder (Informatik und Elektrotechnik)
- ▶ Betreut vom Lehrstuhl i11, Software für eingebettete Systeme und Institut für Kraftfahrzeuge

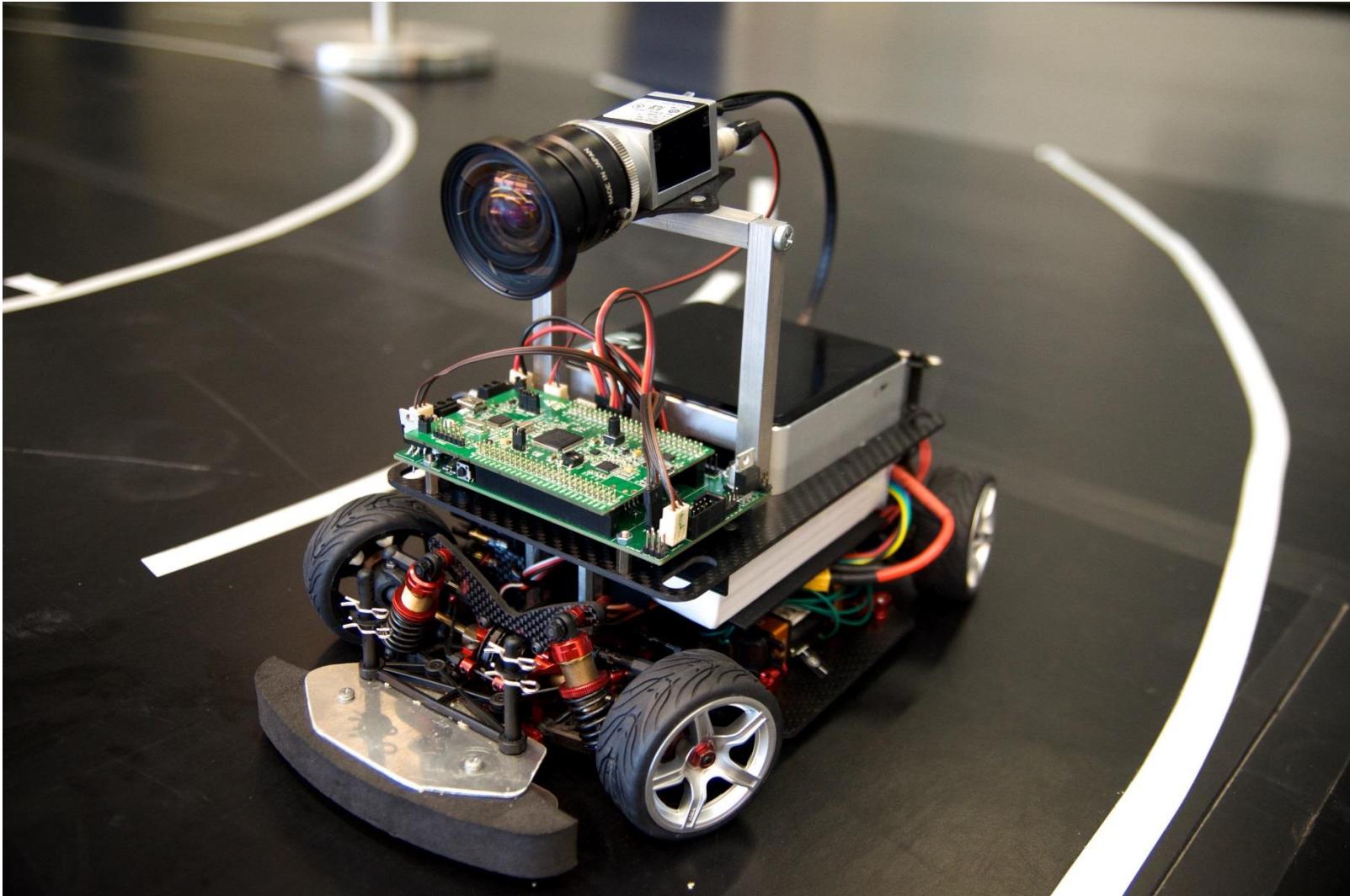


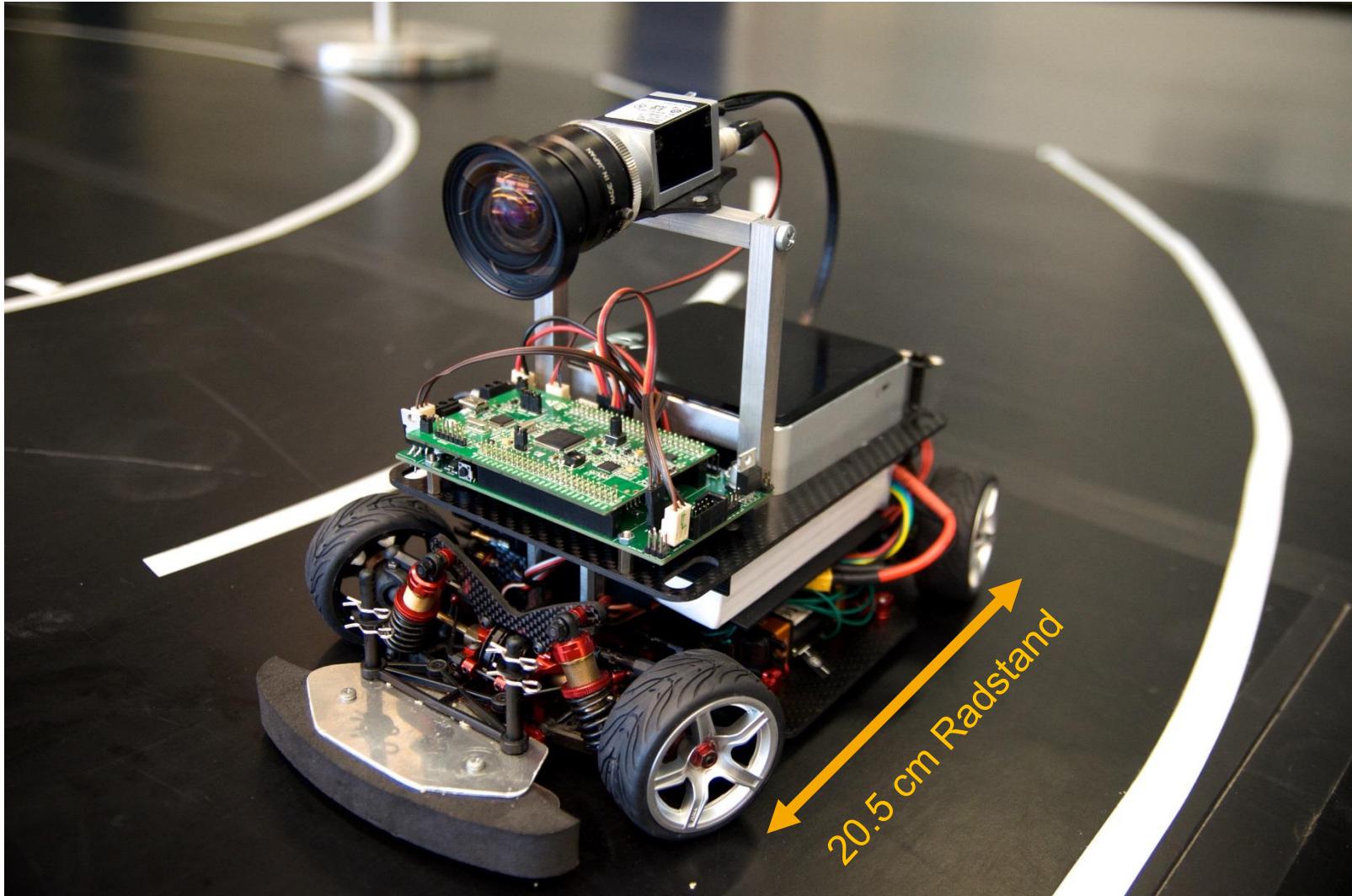
Gliederung

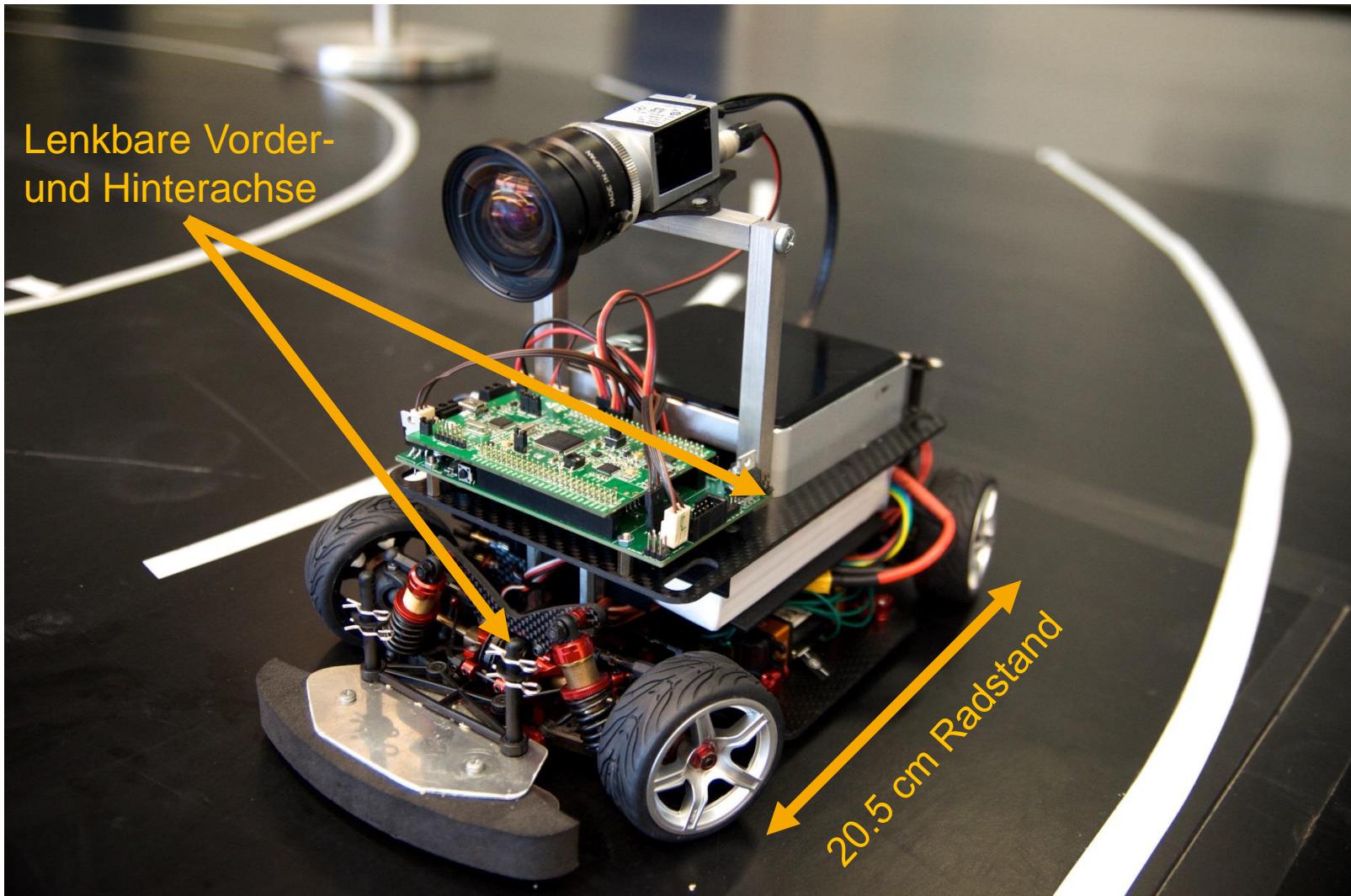
- ▶ Fahrzeugaufbau
- ▶ Hardwarearchitektur
- ▶ Softwarearchitektur
- ▶ Energieverbrauch
- ▶ Kosten
- ▶ Rundfahrt
- ▶ Hindernisfahrt
- ▶ Einparken

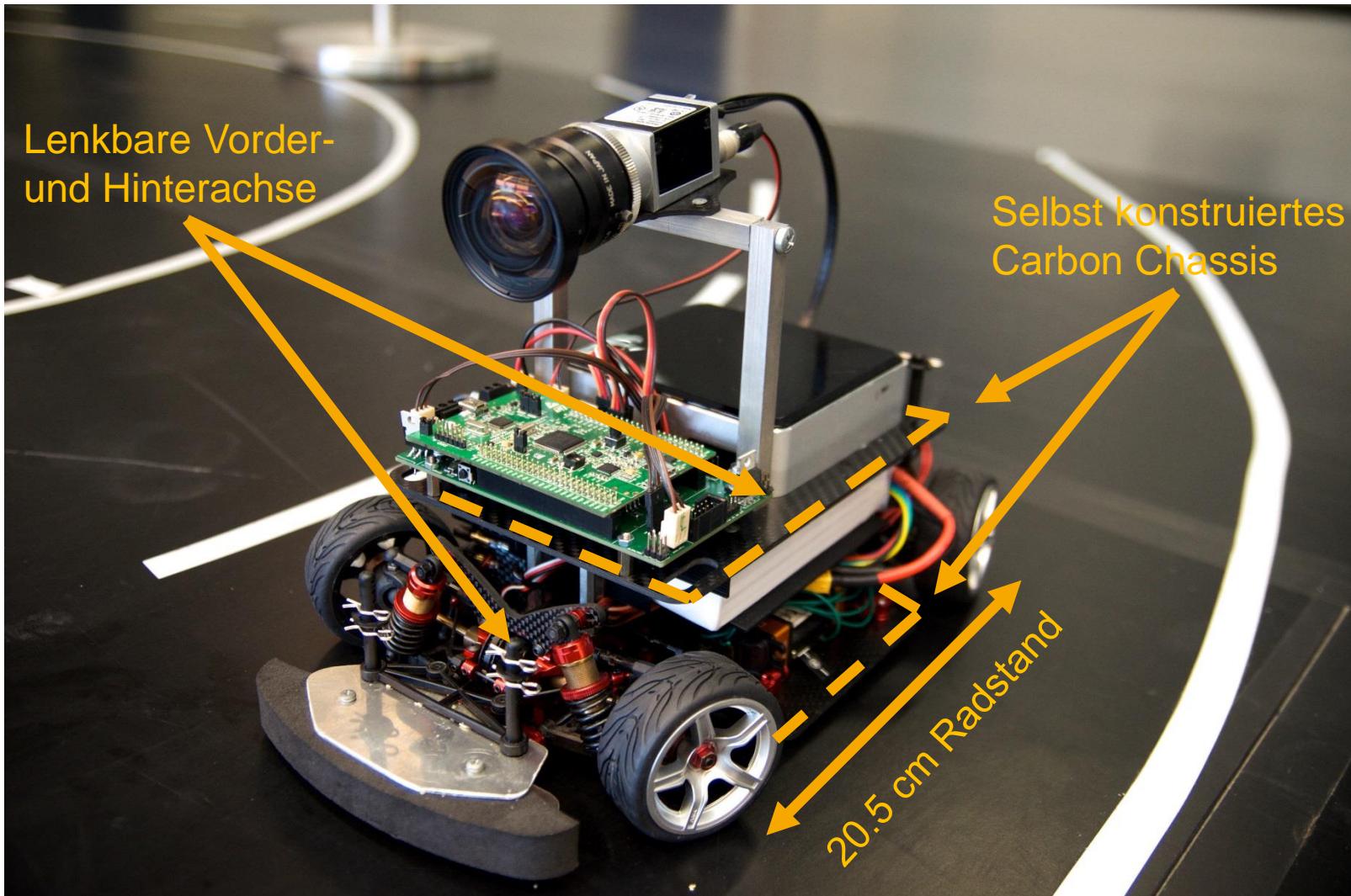
Fahrzeugaufbau

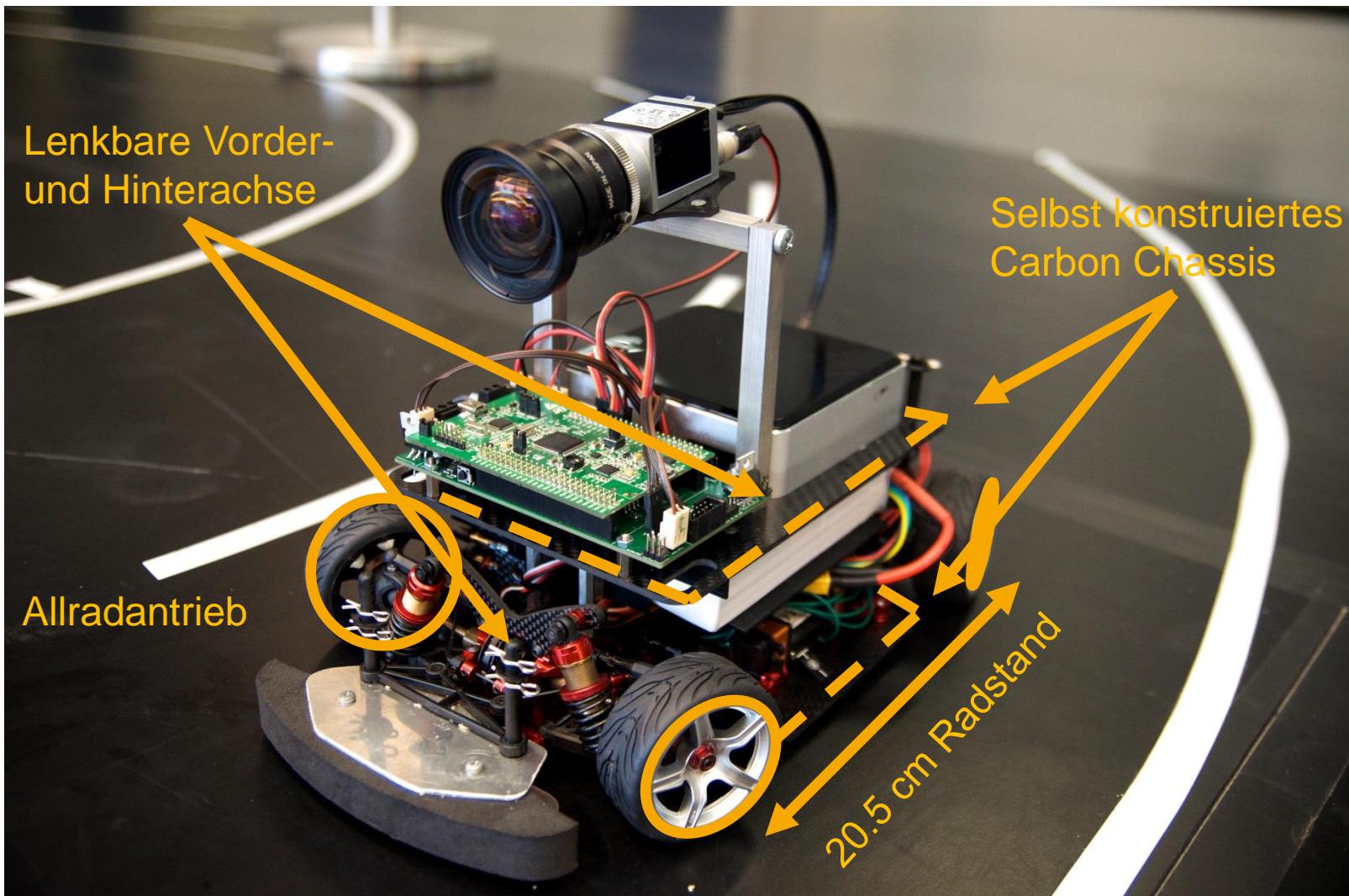
Team GalaXIs: Hyperion





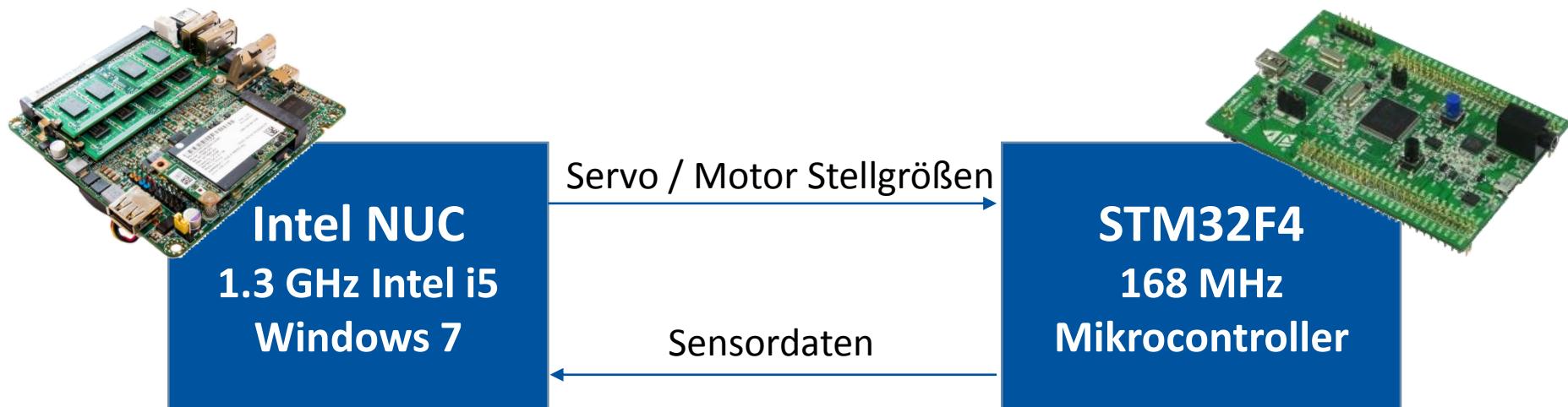




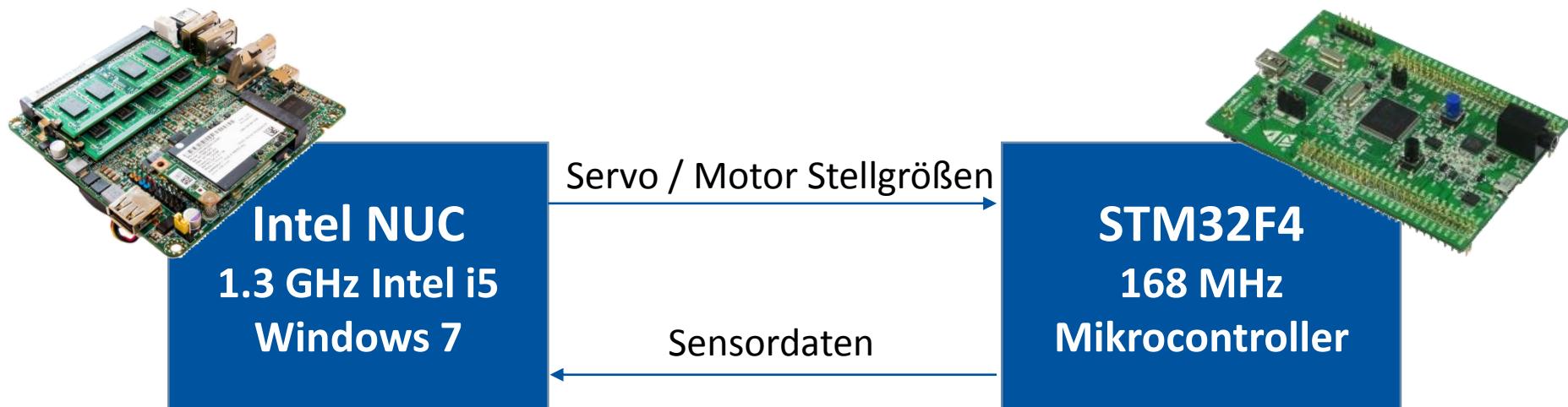


Hardwarearchitektur

Team GalaXIs: Hyperion



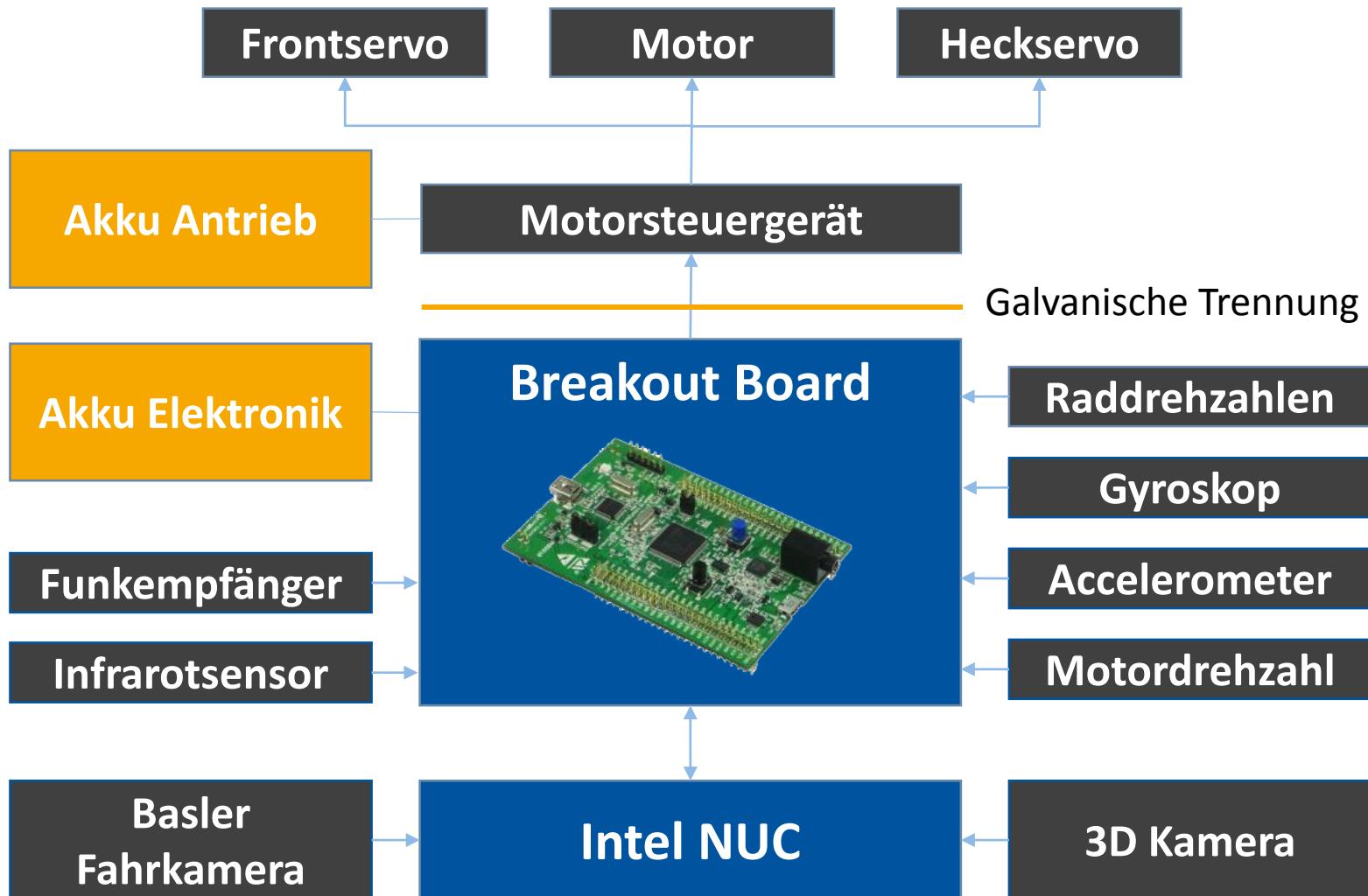
- ▶ Starke Rechenleistung und Schnittstelle zur Hardware erforderlich
- ▶ Intel NUC für Aufgaben wie Straßenerkennung, Hinderniserkennung
- ▶ STM für Ansteuerung der Motoren und Auslesen der Sensoren



- ▶ Aufgabe erfordert sowohl starke Rechenleistung als auch IO Kommunikation
- ▶ Intel NUC für Aufgaben wie Straßenerkennung, Hinderniserkennung
- ▶ STM für Ansteuerung der Motoren und Auslesen der Sensoren

- ▶ Kommunikation über RS232/USB - 3 MBit/s, 1 ms Latenz

Sensorik/Aktuatorik



Softwarearchitektur

Team GalaXIs: Hyperion

Softwarearchitektur

- ▶ Programmiersprache C/C++
- ▶ Externe Libraries frei verfügbar
 - OpenCV (Bilderkennung)
 - OpenNI (3D-Kamera)
 - NLOpt (Optimierungsalgorithmen)
 - Point Cloud Library (Parklückenerkennung)
 - FLANN (Fast Library for Approximate Nearest Neighbors)
- ▶ Versionskontrolle GIT
- ▶ Entwicklungsumgebung Visual Studio 2013

Softwarearchitektur

- ▶ Lose gekoppelte Libraries
- ▶ Möglichst einfach gehalten
- ▶ Kein Framework, Middleware oder spezielles Scheduling
- ▶ Ermöglicht separate Entwicklung und Testen

Netzwerkkamera
auslesen

Sensorik/Aktuatorik

Parklücken-
erkennung

Remote Telemetrie

Straßenerkennung

Hinderniserkennung

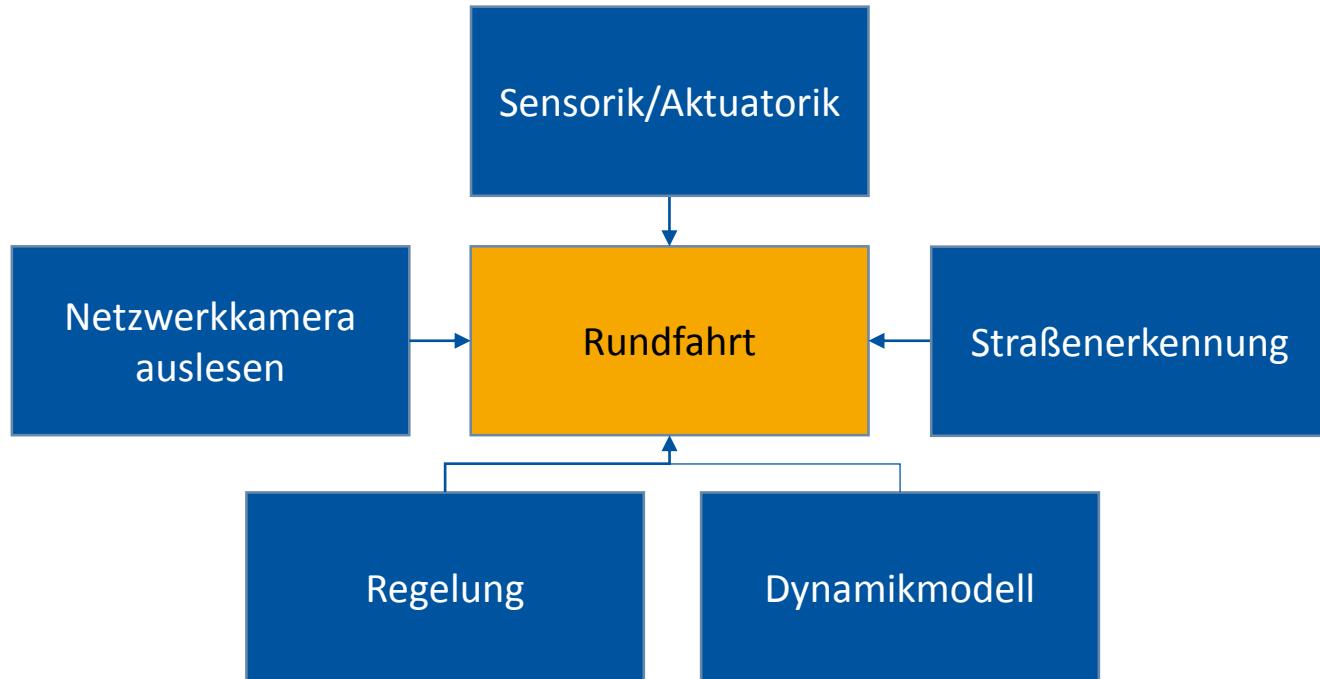
Stoplinienerkennung

Regelung

Dynamikmodell

Softwarearchitektur

- ▶ Verknüpfung der Libraries in **Szenarios**
- ▶ Keine Echtzeitfähigkeit notwendig



Energiebilanz

Team GalaXIs: Hyperion

Energiebilanz

Steuerstromkreis: 14.8 V, 3200 mAh

Intel NUC	15	W
STM32F4	0.625	W
Gesamt	15.625	W

Fahrstromkreis: 7.4 V, 6000 mAh

Aktorik	20	W
---------	----	---

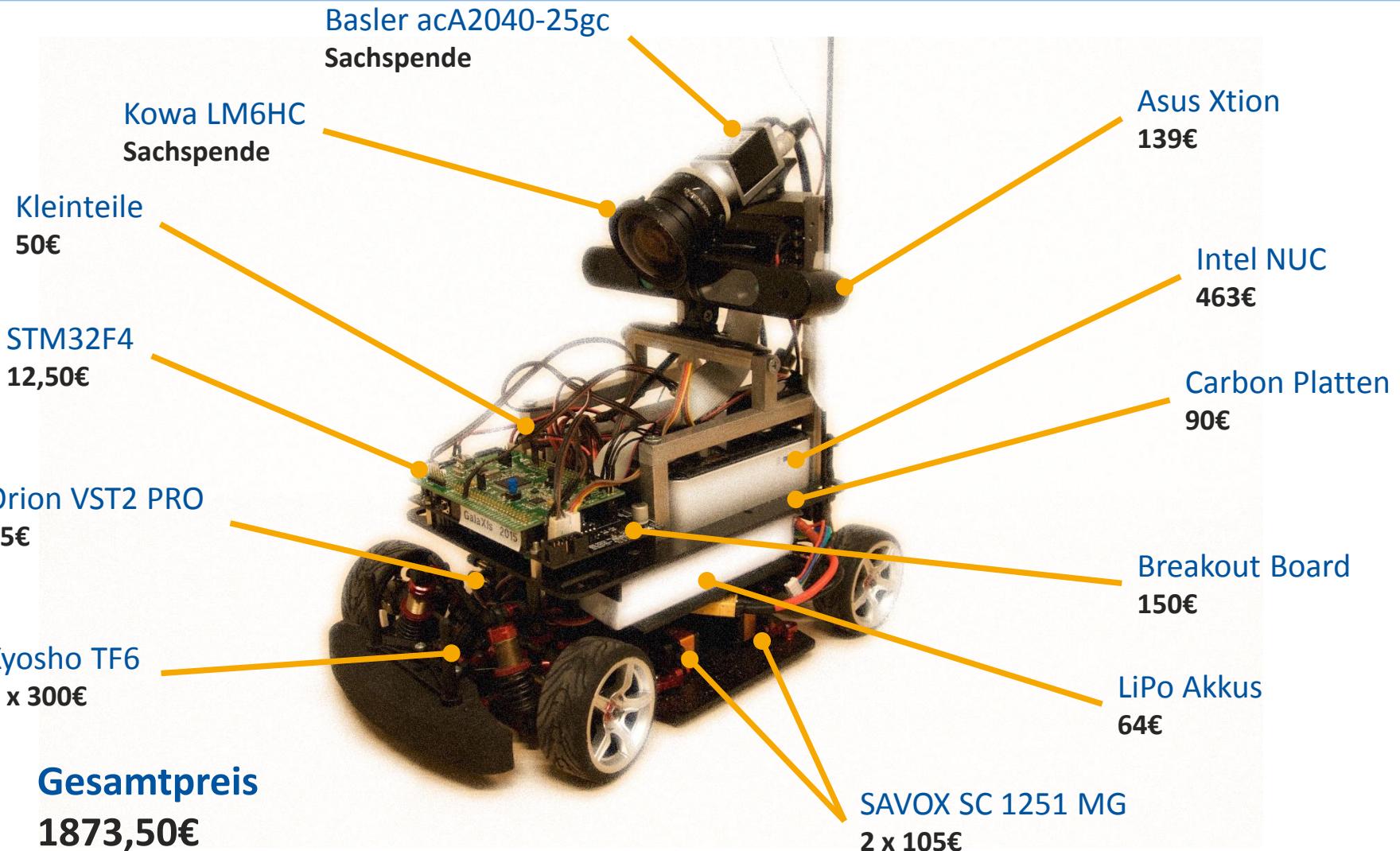


Mögliche Fahrzeit:
ca. 2 h

Kostenbilanz

Team GalaXIs: Hyperion

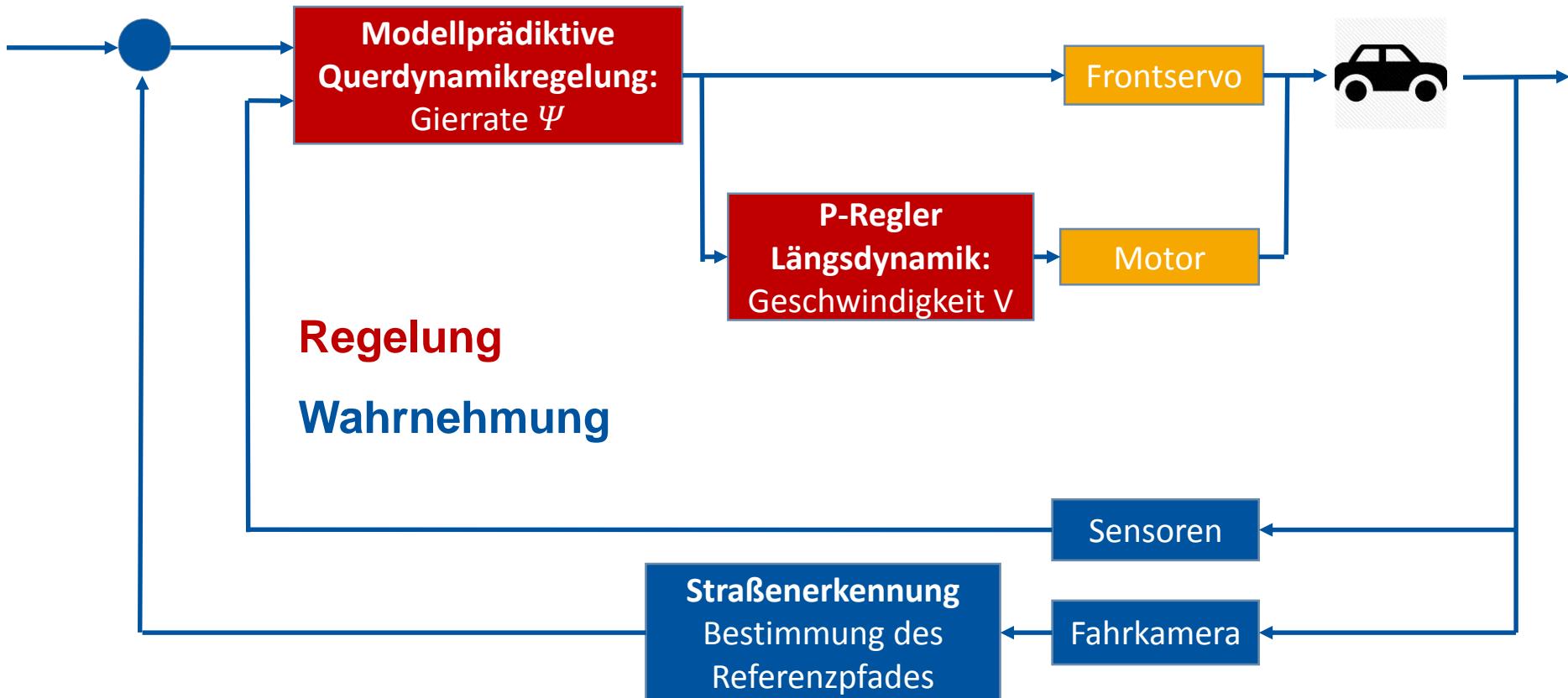
Kostenbilanz



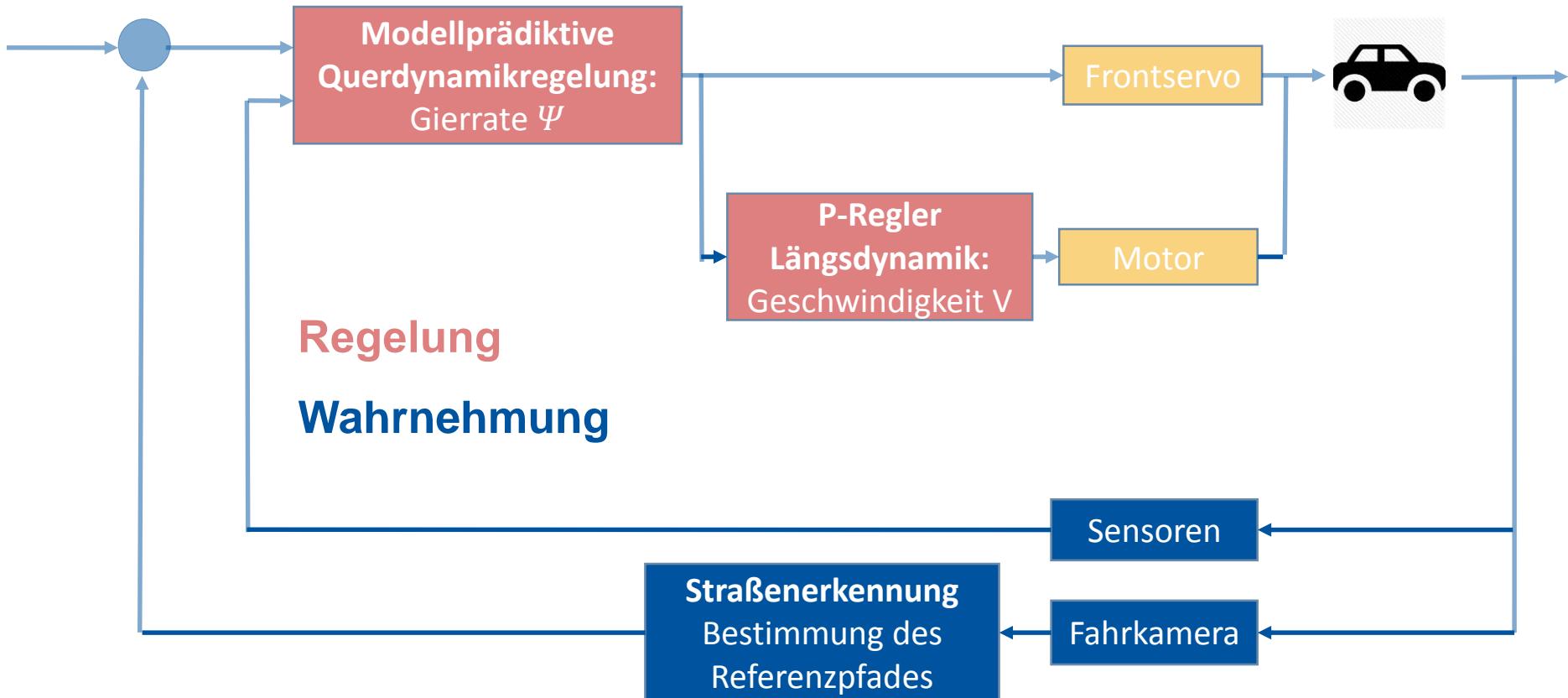
Systemkonzept Rundfahrt

Team GalaXIs: Hyperion

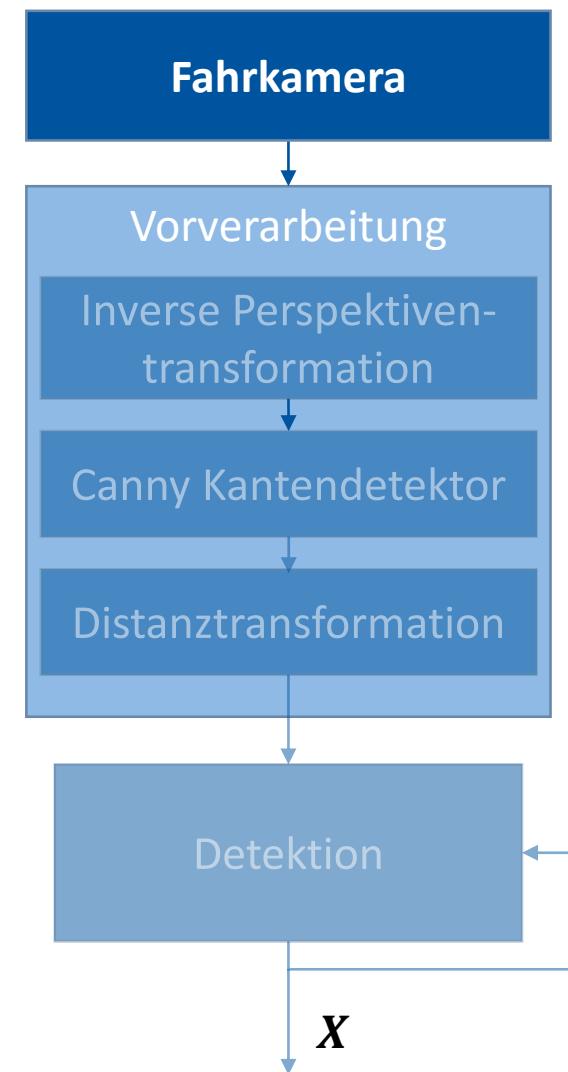
Systemkonzept Rundfahrt



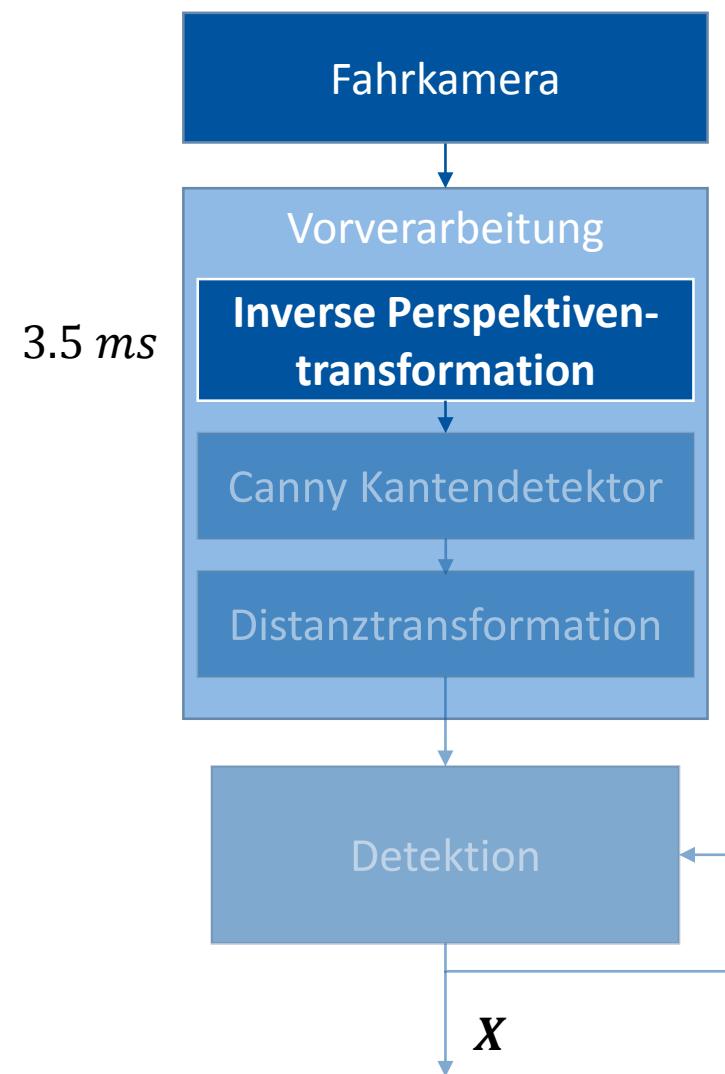
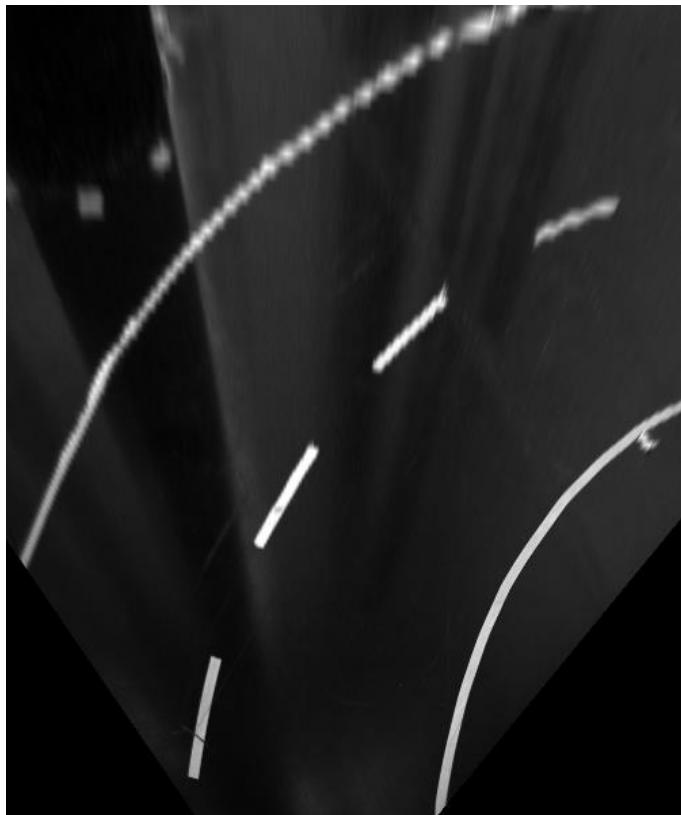
Systemkonzept Rundfahrt



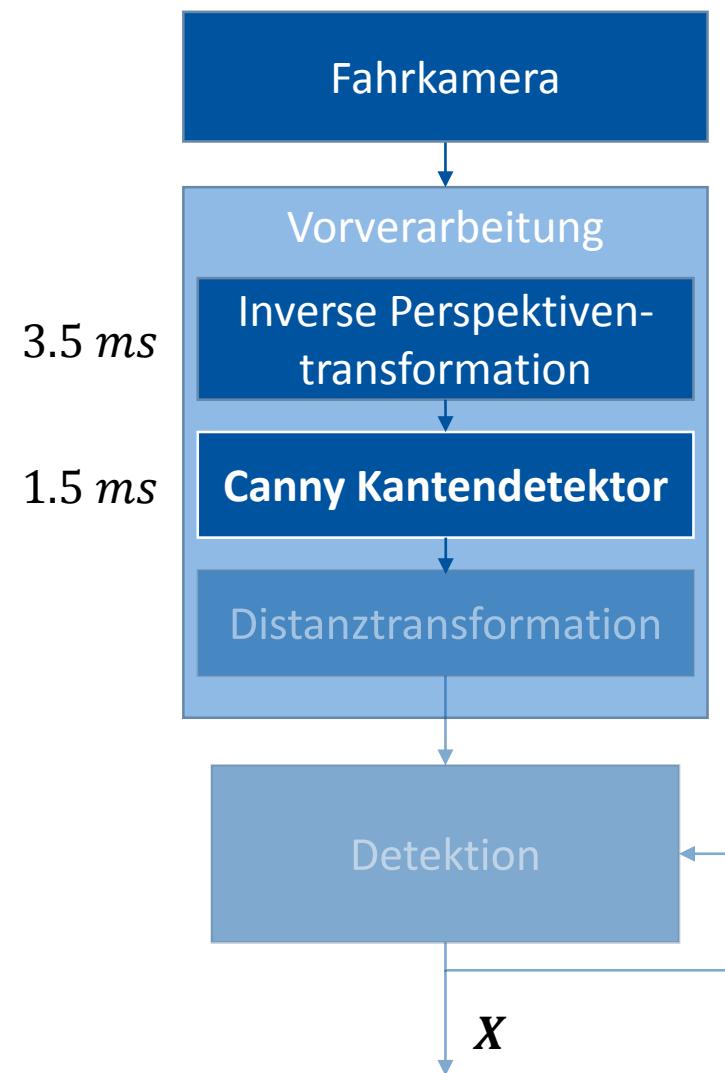
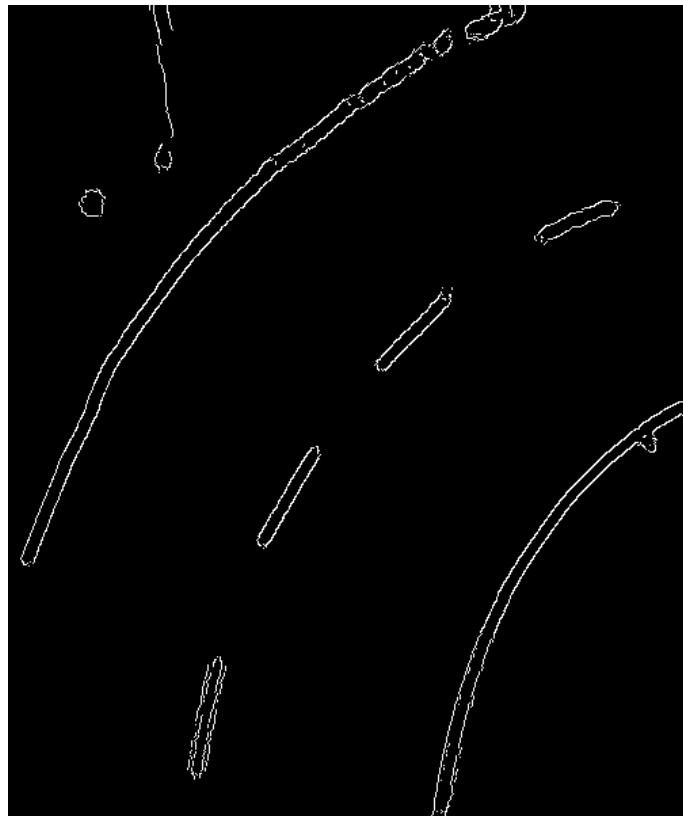
Straßenerkennung – Bestimmung des Referenzpfades



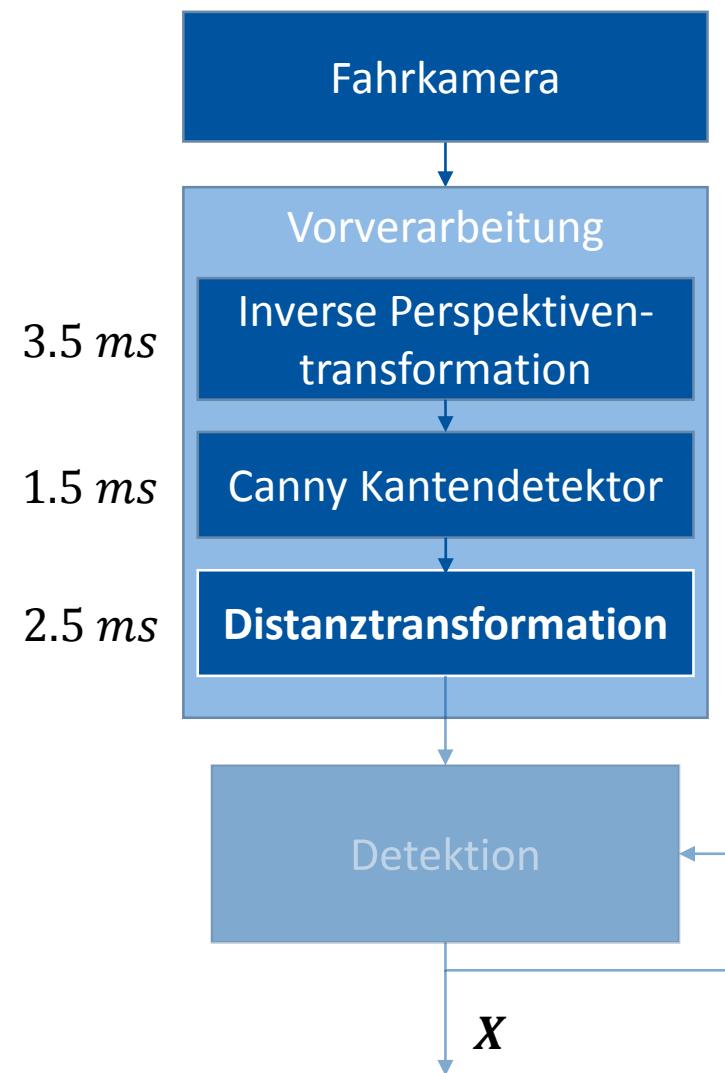
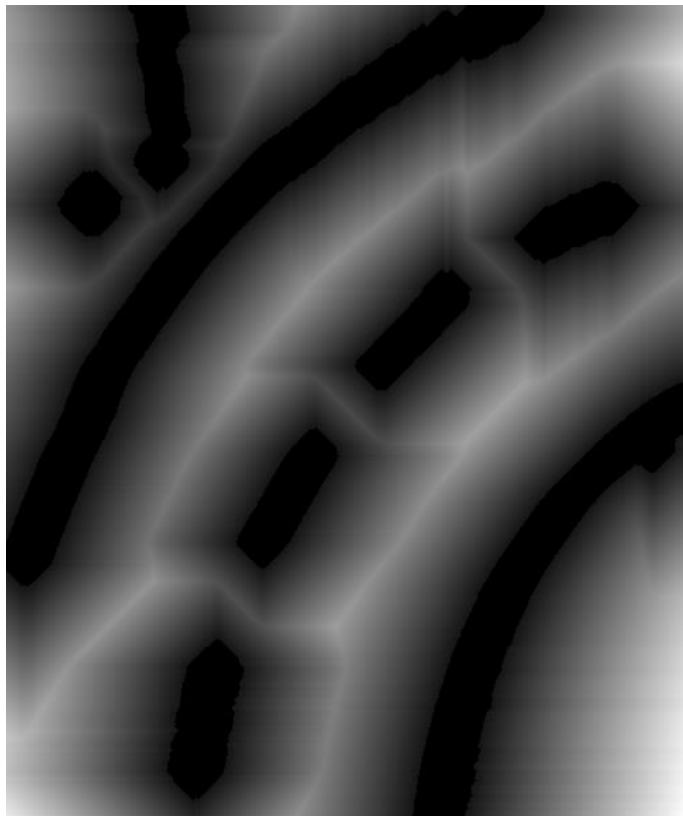
Straßenerkennung – Bestimmung des Referenzpfades



Straßenerkennung – Bestimmung des Referenzpfades

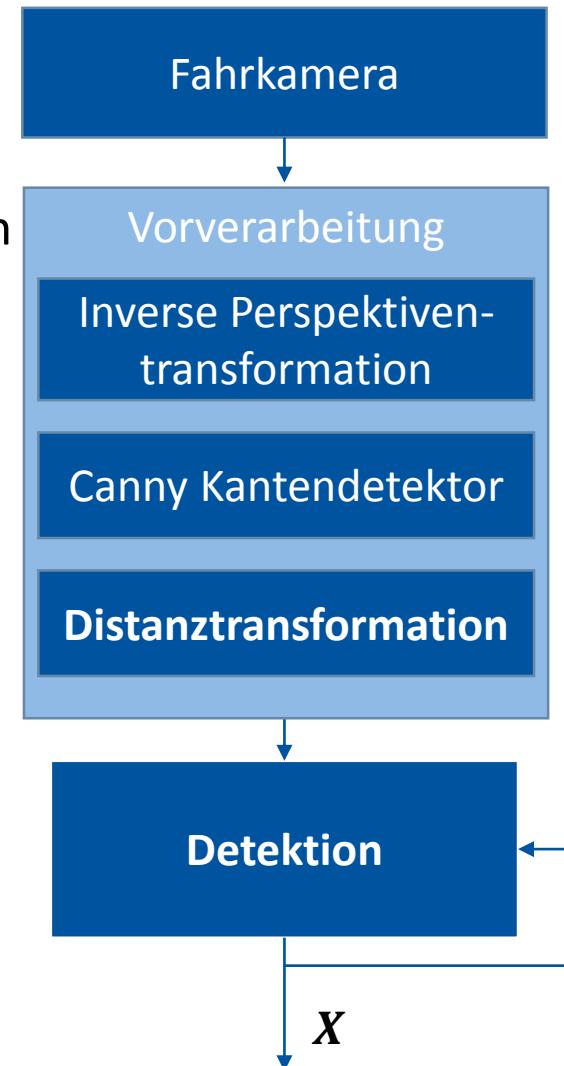
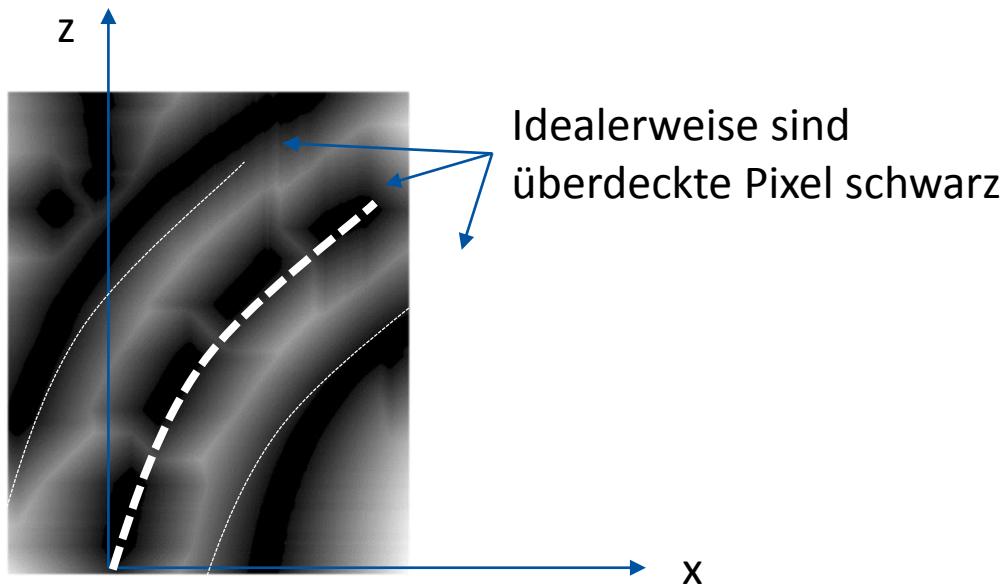


Straßenerkennung – Bestimmung des Referenzpfades



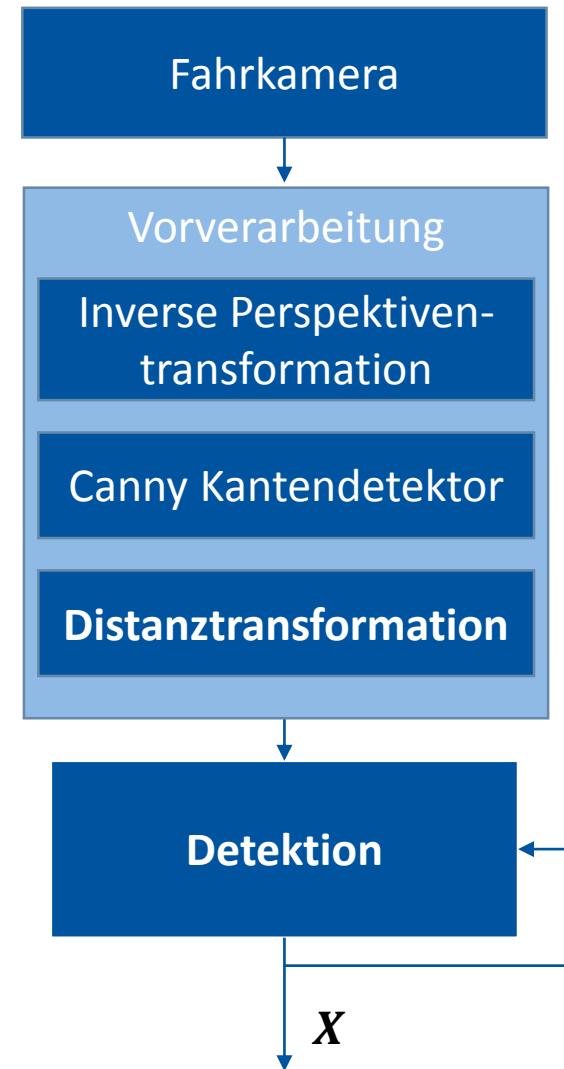
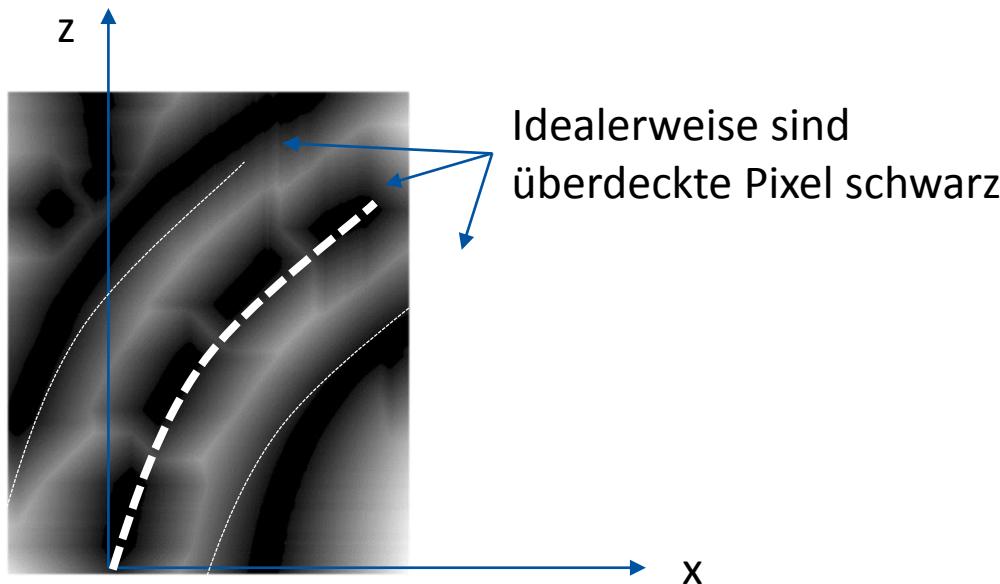
Straßenerkennung – Bestimmung des Referenzpfades

- ▶ Polynom modelliert mittlere Fahrspur
- ▶ Gütefunktional $F(X, DT)$
- ▶ Generierung hypothetischer Fahrbahnmarkierungen
- ▶ **Gütekriterium** basierend auf überdeckten Pixeln

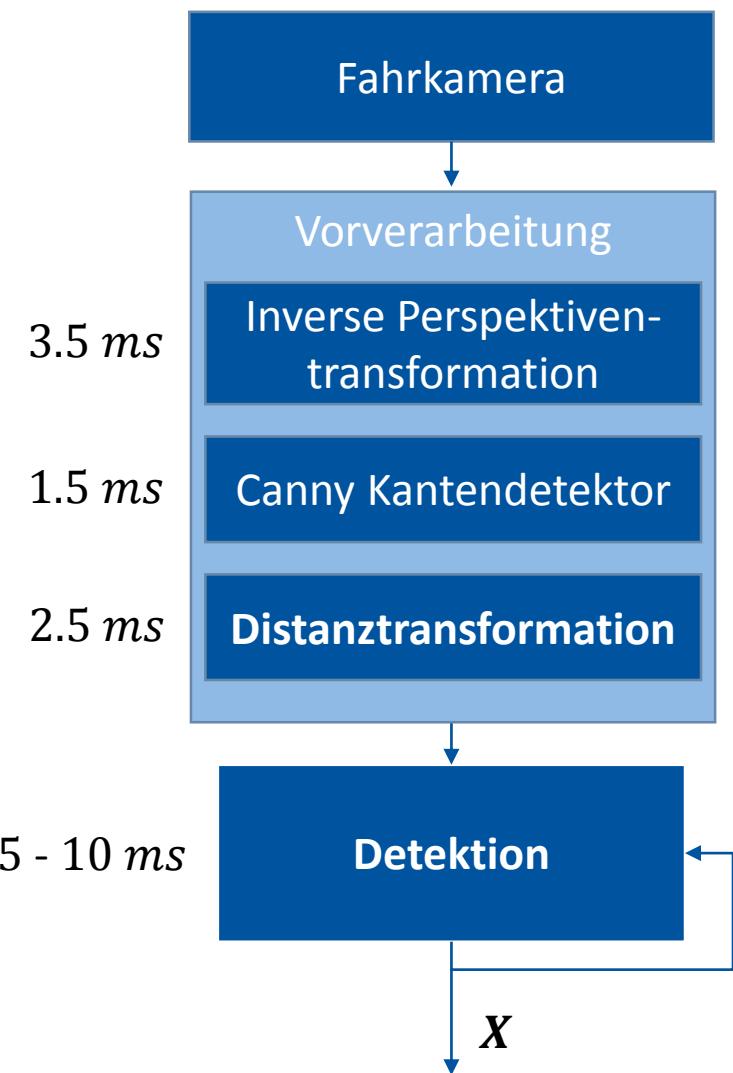
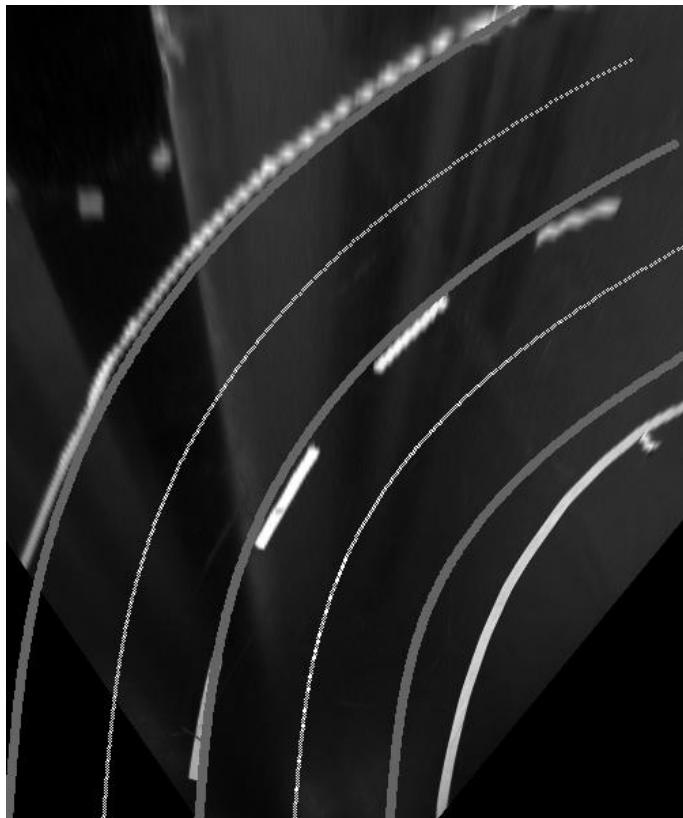


Straßenerkennung – Bestimmung des Referenzpfades

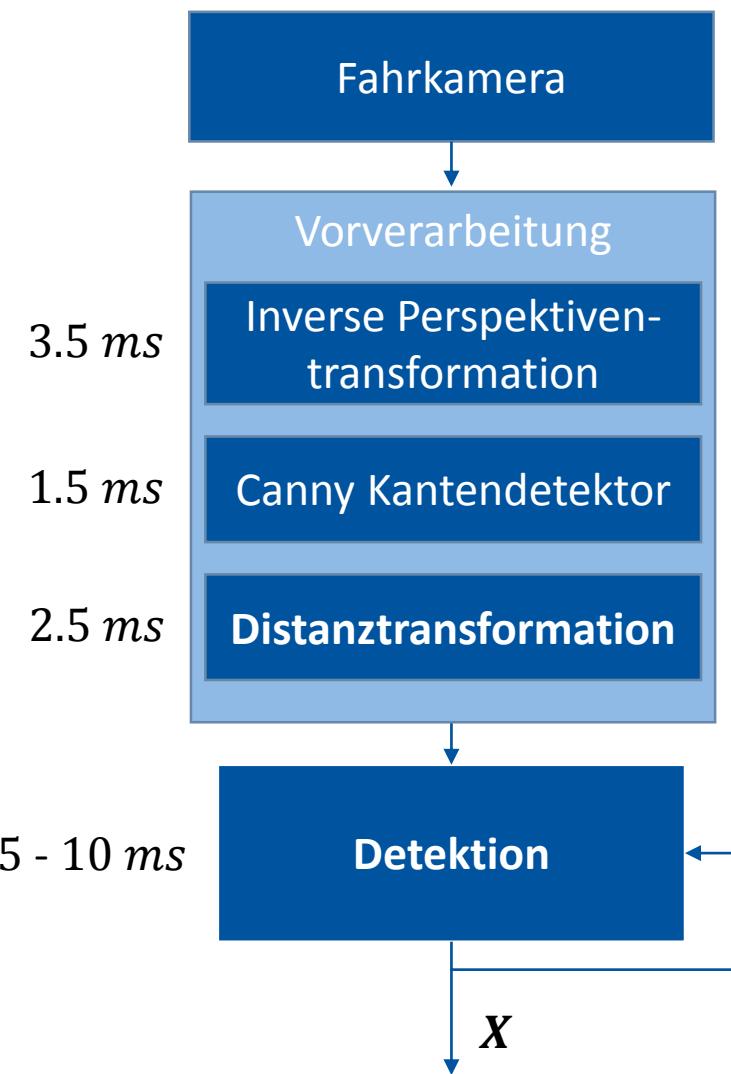
- ▶ Maximierung von $F(X, DT)$ mittels CRS
- ▶ Beschränkter Parameterraum
- ▶ **Tracking:** Ergebnis fungiert als neuer Startwert



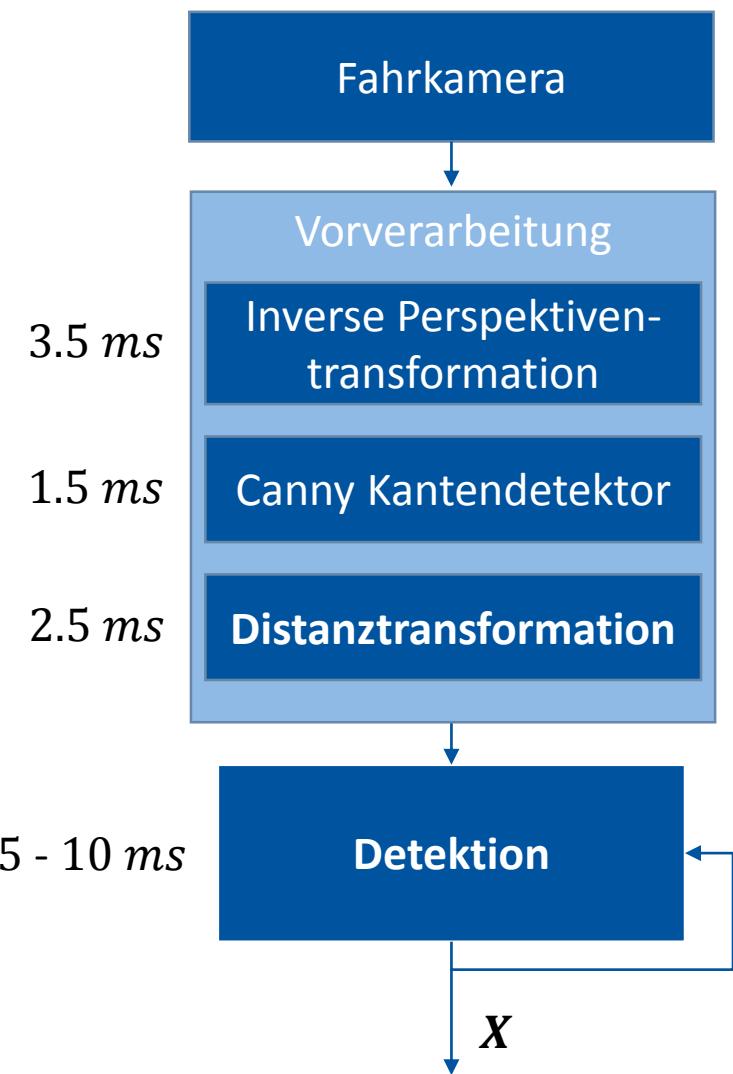
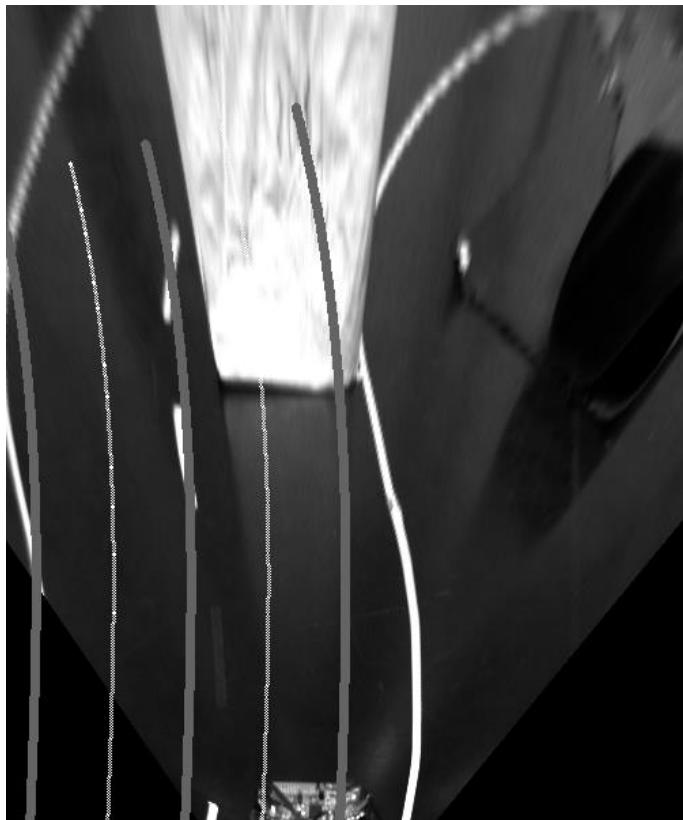
Straßenerkennung – Bestimmung des Referenzpfades



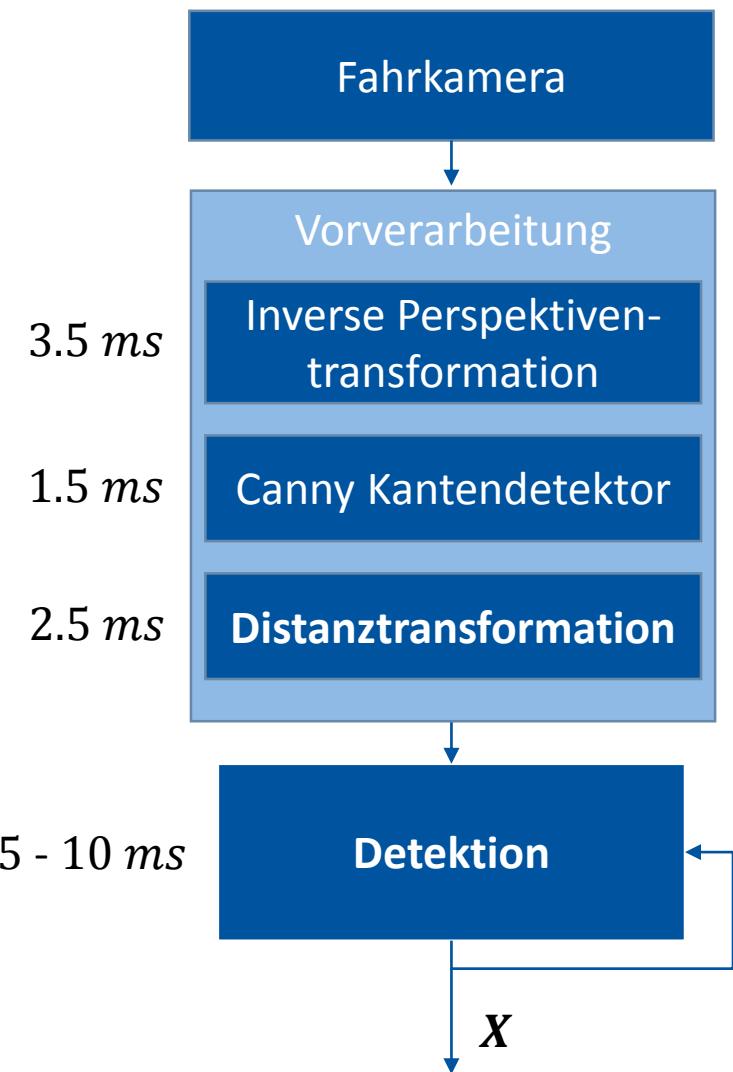
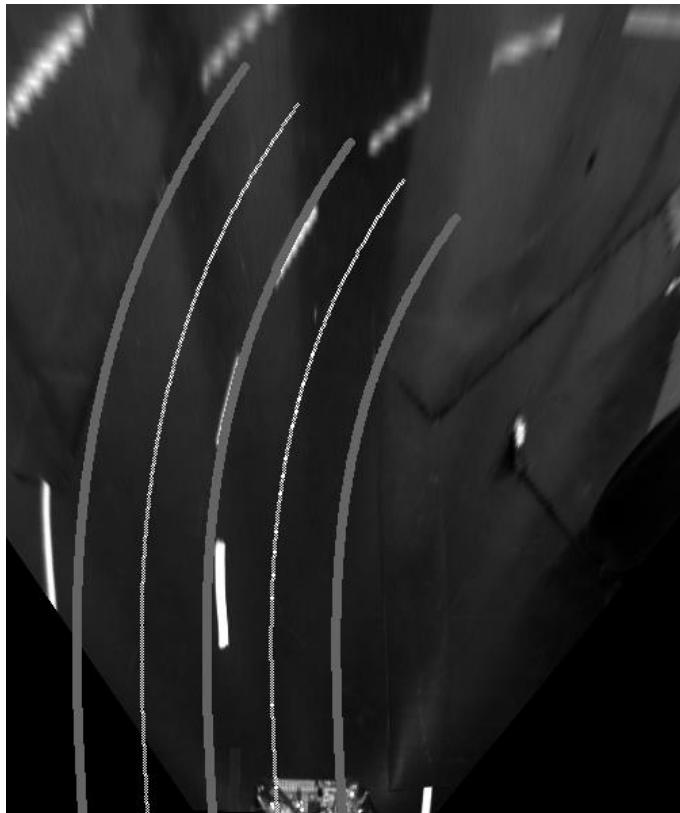
Straßenerkennung – Bestimmung des Referenzpfades



Straßenerkennung – Bestimmung des Referenzpfades



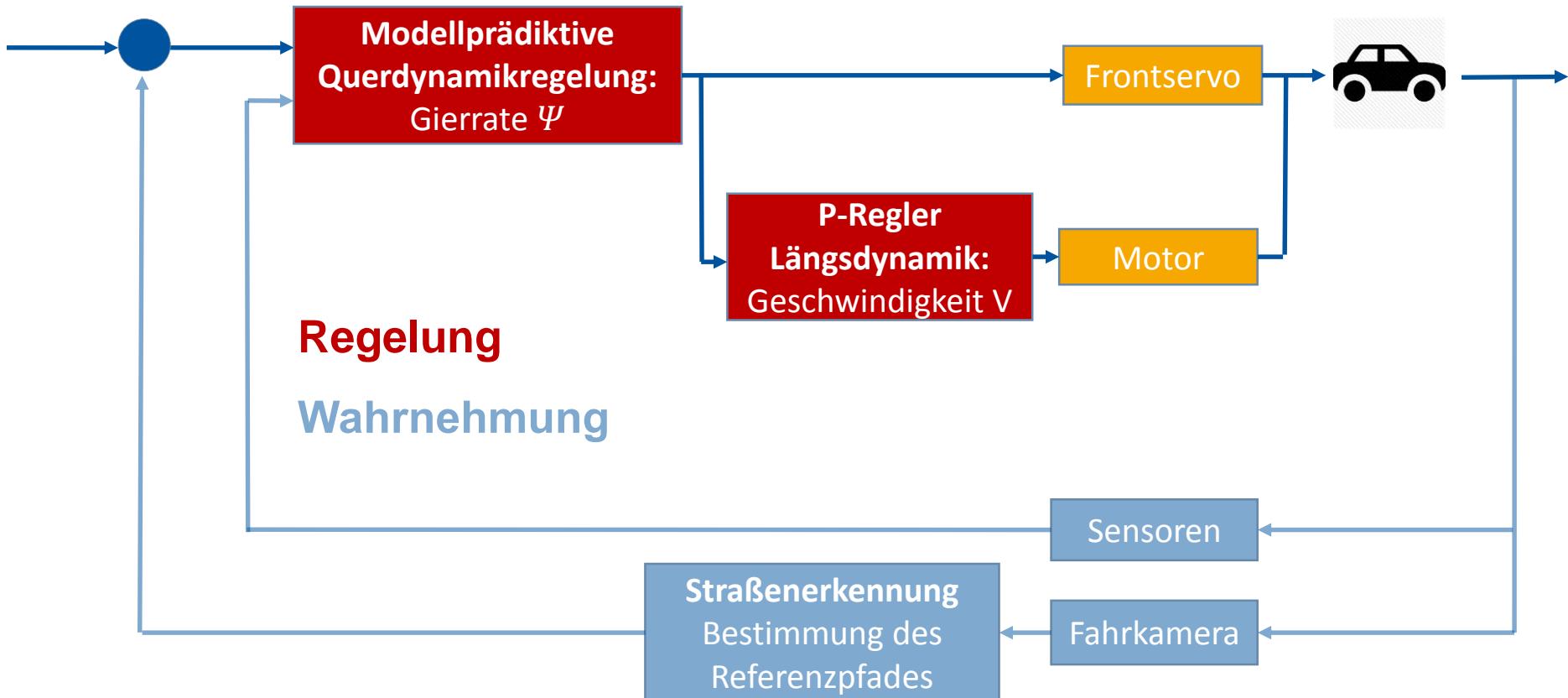
Straßenerkennung – Bestimmung des Referenzpfades



Straßenerkennung – Bestimmung des Referenzpfades

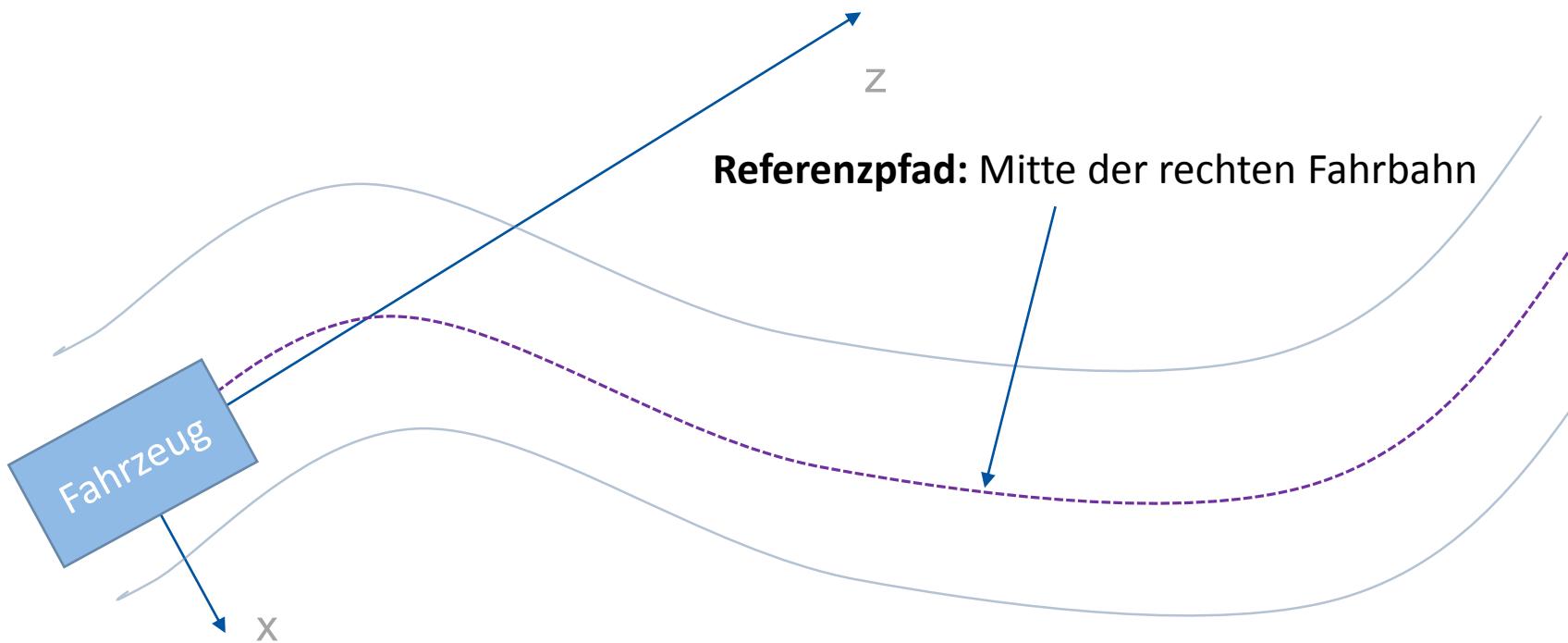
- ▶ Sehr schneller Ansatz (**55-80 FPS**) bei einer Bildgröße von **2046x600**
 - Mit Partikelfilter vergleichbare Güte, gleichzeitig effizienter
- ▶ Robust gegenüber fehlenden Markierungen
- ▶ Informationen zur Straßengeometrie implizit eingebunden
 - Parallelität der Markierungen
 - Fahrspurbreite
 - Mögliche Straßenkrümmungen

Systemkonzept Rundfahrt



Modellprädiktive Querdynamikregelung

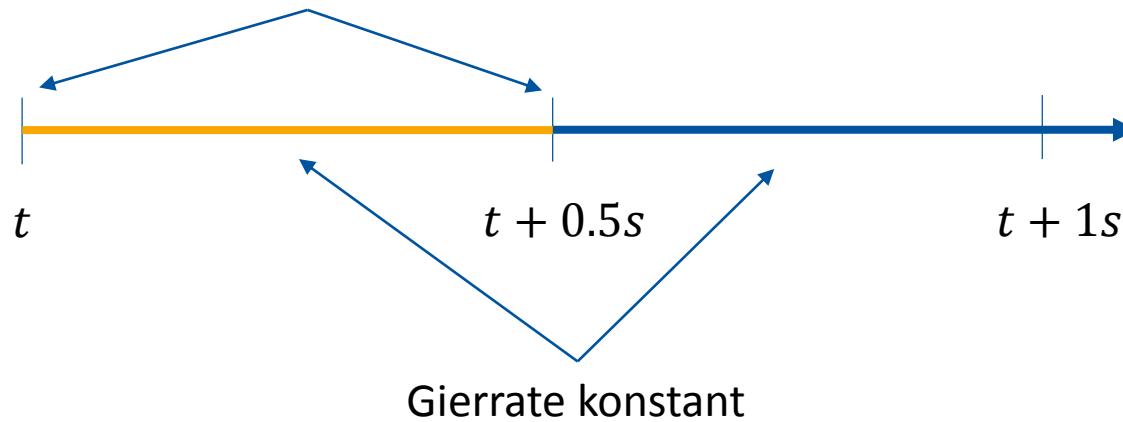
- ▶ Eingaben
 - Referenzpfad als Punktmenge im lokalen Fahrzeugkoordinatensystem
 - Momentane Geschwindigkeit
- ▶ Resultat: optimale Gierrate



Modellprädiktive Querdynamikregelung

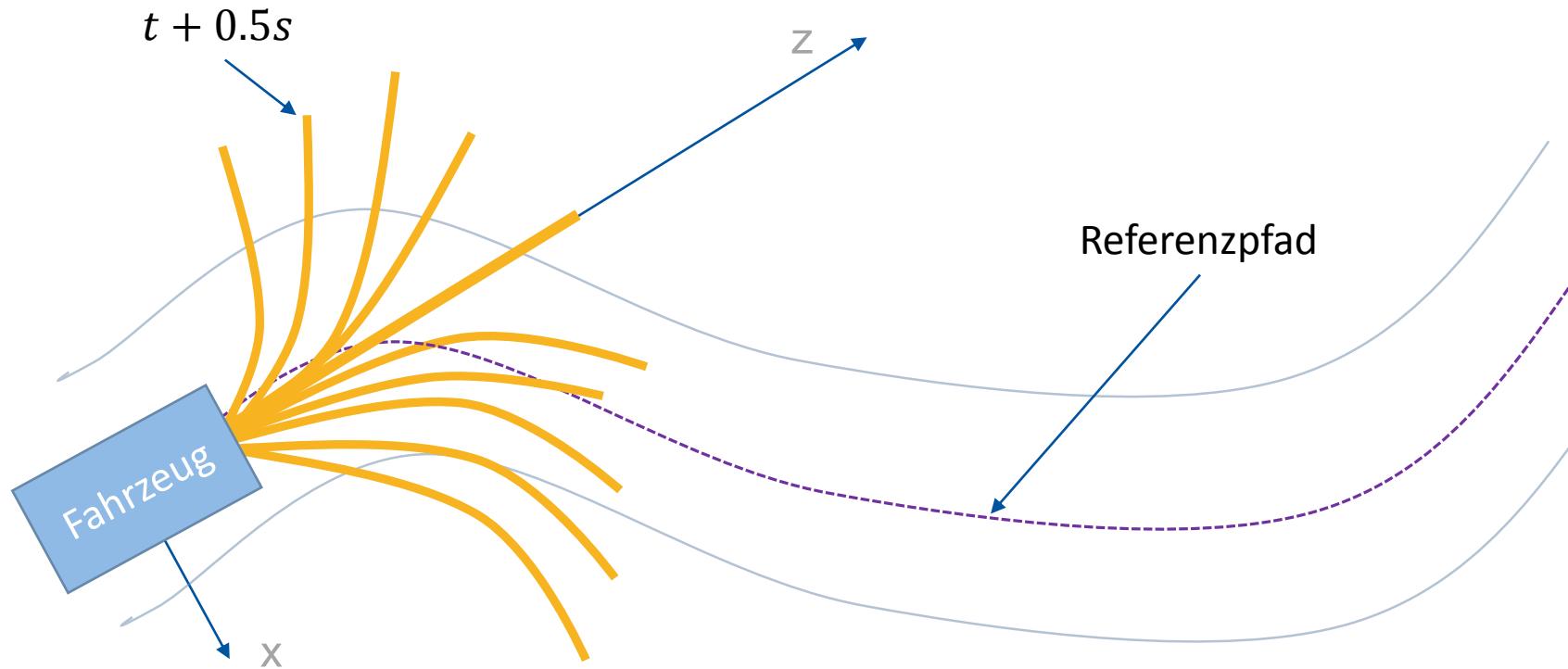
- ▶ Zeithorizont 1s
- ▶ Prädiktion der Fahrzeugposition für verschiedene Gierraten $\dot{\psi} \in \Psi$
- ▶ Ψ Diskretisierung der fahrbaren Gierraten
- ▶ Stellzeitpunkte t und $t + 0.5s$ mit $(\dot{\psi}_t, \dot{\psi}_{t+0.5s})$
- ▶ Gierraten zwischen Stellzeitpunkten **konstant**

Stellzeitpunkte (Gierratenänderung erlaubt)



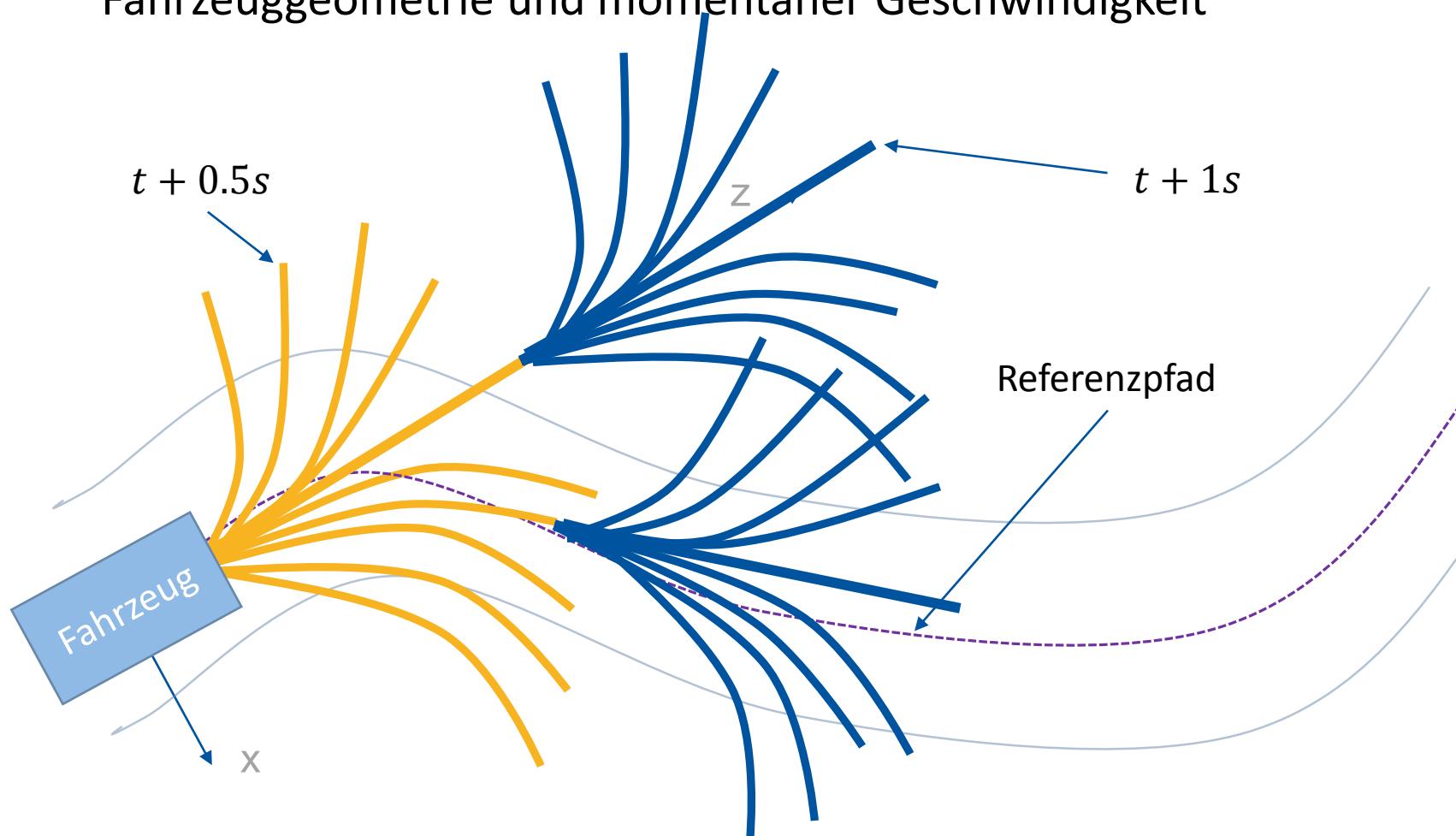
Modellprädiktive Querdynamikregelung

- Vorausberechnung der zukünftigen Position auf Basis der Fahrzeuggeometrie und momentaner Geschwindigkeit



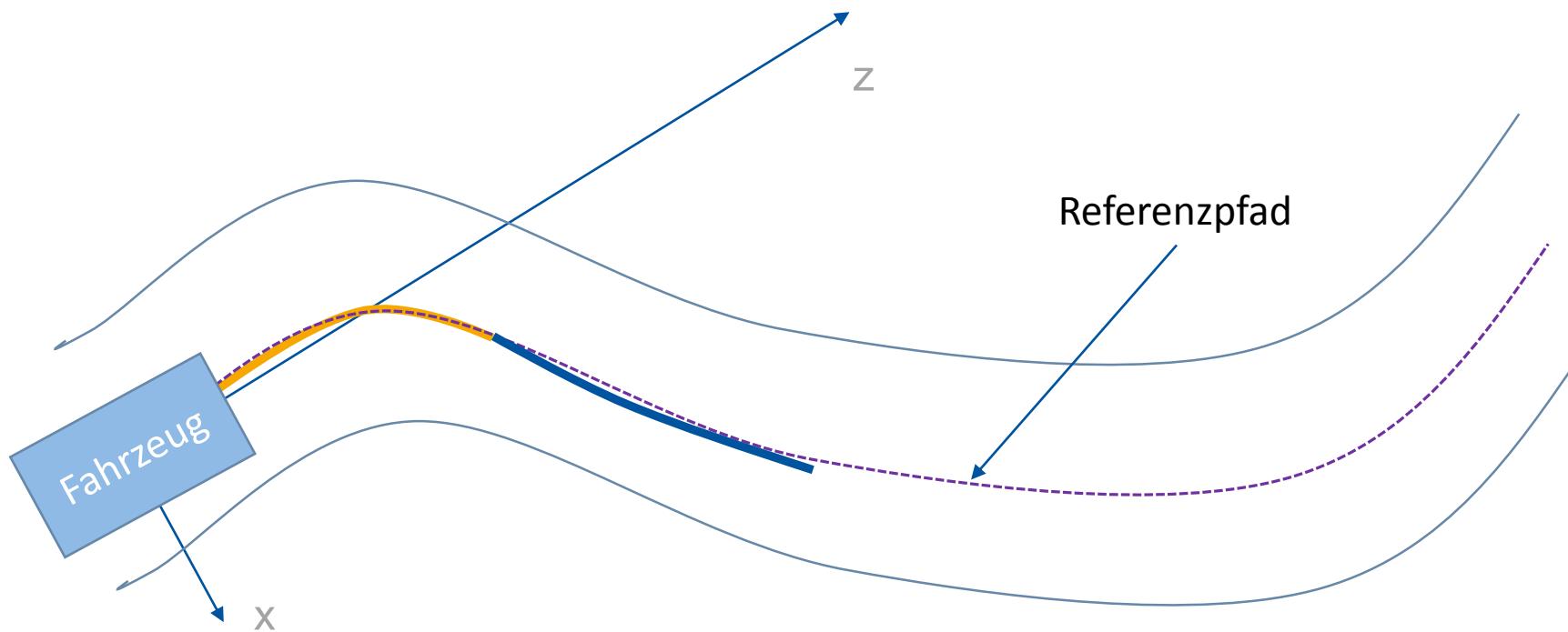
Modellprädiktive Querdynamikregelung

- Vorausberechnung der zukünftigen Position auf Basis der Fahrzeuggeometrie und momentaner Geschwindigkeit

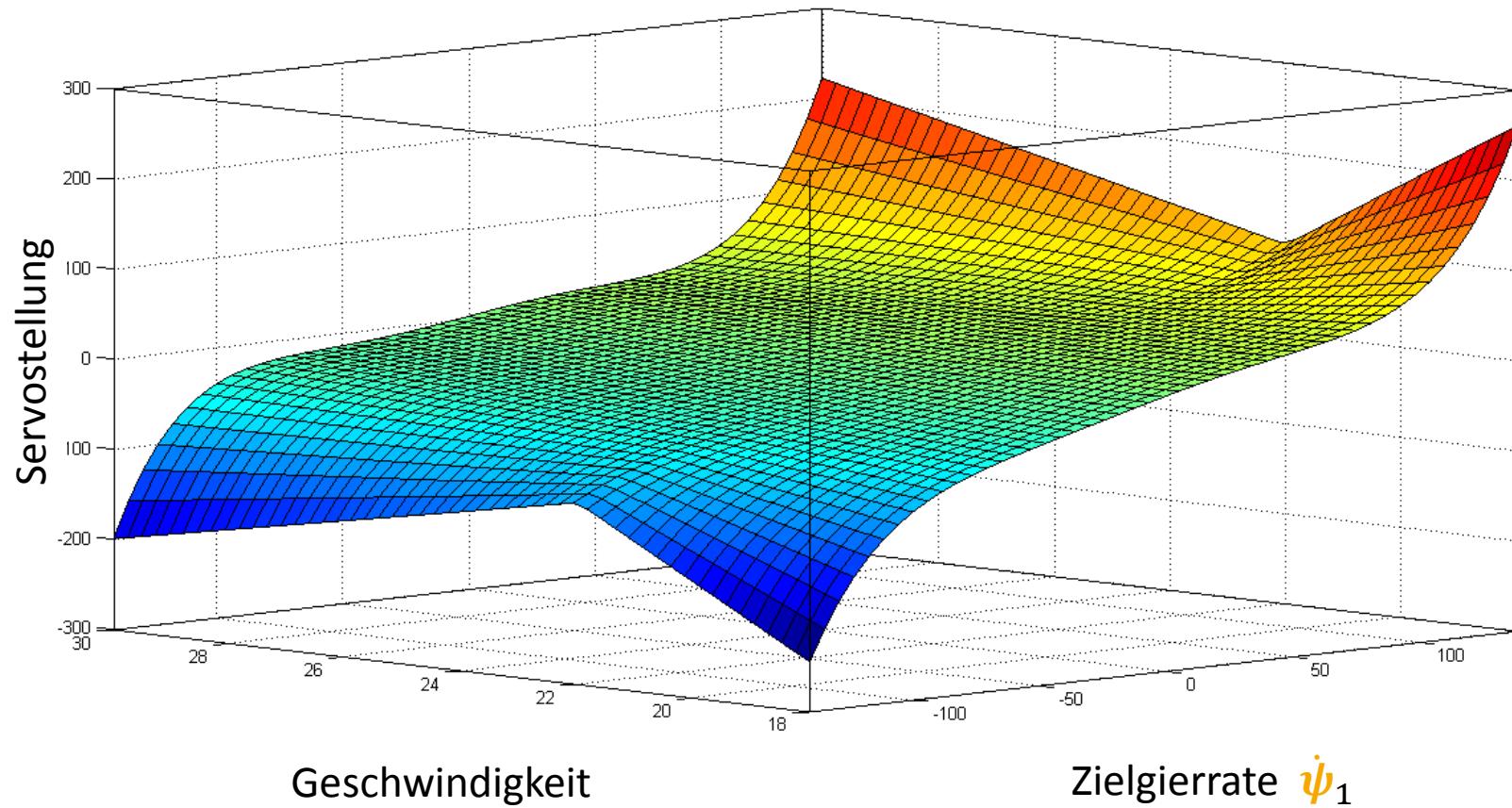


Modellprädiktive Querdynamikregelung

- Ergebnis: Variante $(\dot{\psi}_1, \dot{\psi}_2)$ mit minimalem Fehler zum Referenzpfad
- Evaluation von 5500 Varianten in $\sim 3 \text{ ms}$
- Berücksichtigt wird nur $\dot{\psi}_1$



Modellprädiktive Querdynamikregelung



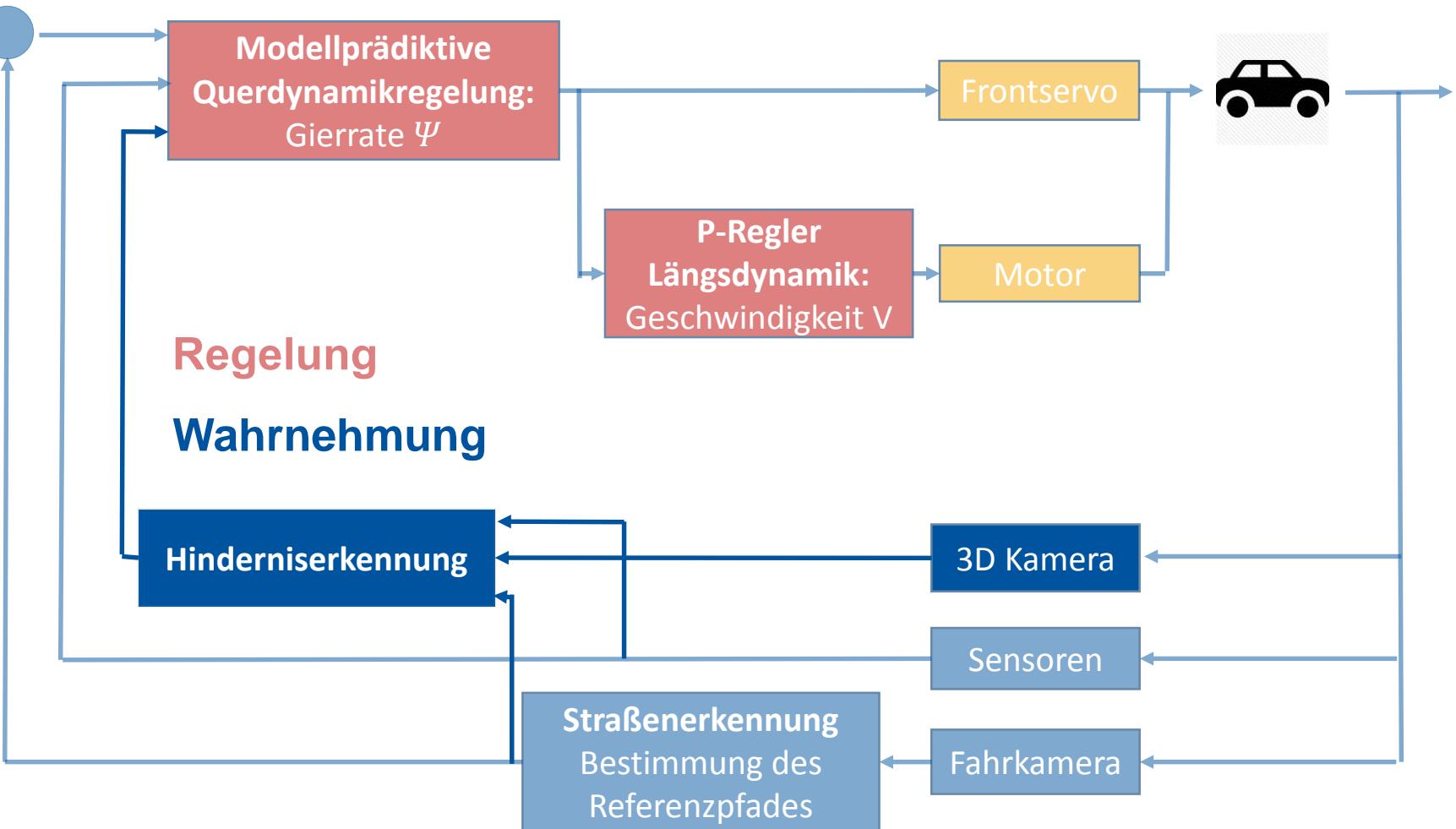
Modellprädiktive Querdynamikregelung

- ▶ Umgang mit Nichtlinearität und Multivariabilität
 - ▶ Implizite Berücksichtigung des zukünftigen Straßenverlaufs
 - ▶ Intuitive Auslegung
 - ▶ Nachteil: Berechnungsaufwand
-
- ▶ Alternativlösung PID Regler mit Gainscheduling
 - Komplexere Auslegung
 - Parameterwerte haben keine klare Bedeutung
 - Kurvenradius als Vorsteuerung nötig, problematisch bei Kurven Ein- und Ausfahrten
 - Ausweichmanöver erfordert synthetisierten Kurvenradius

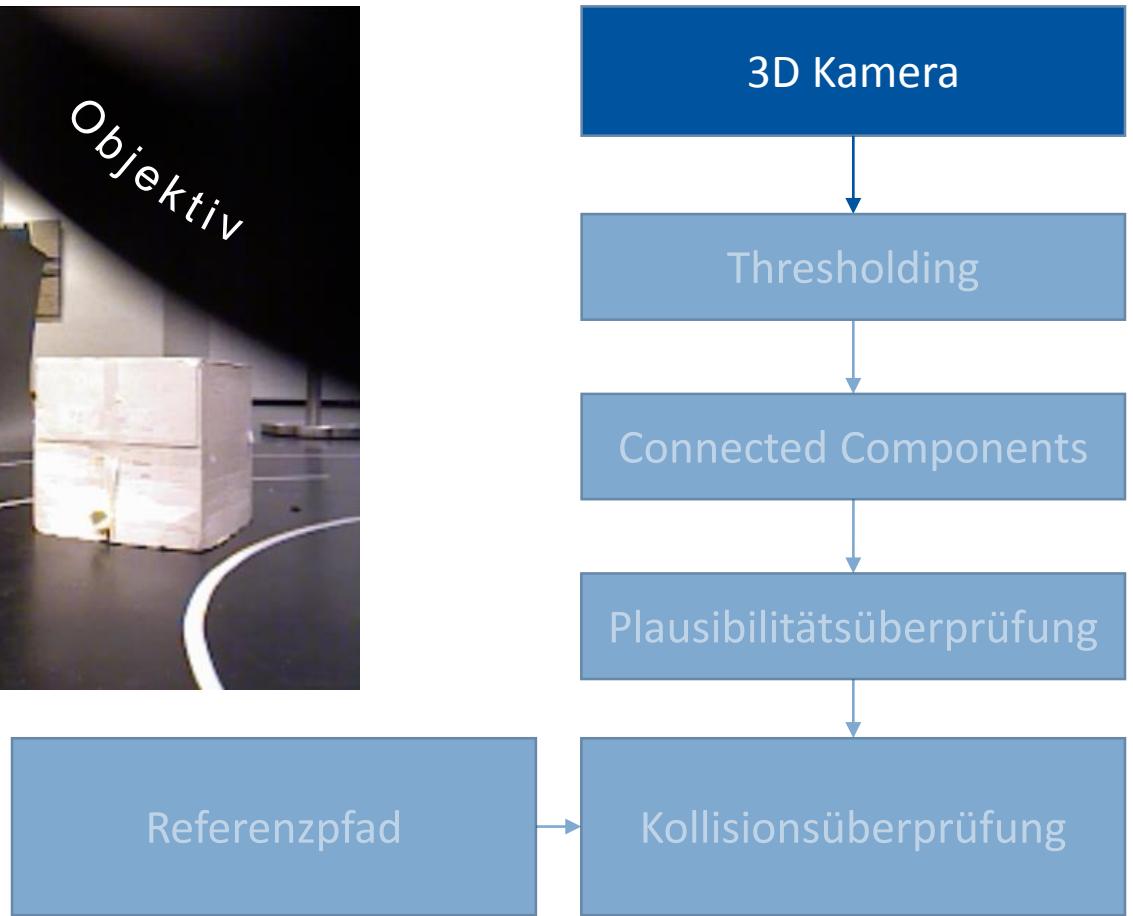
Systemkonzept Hindernisfahrt

Team GalaXIs: Hyperion

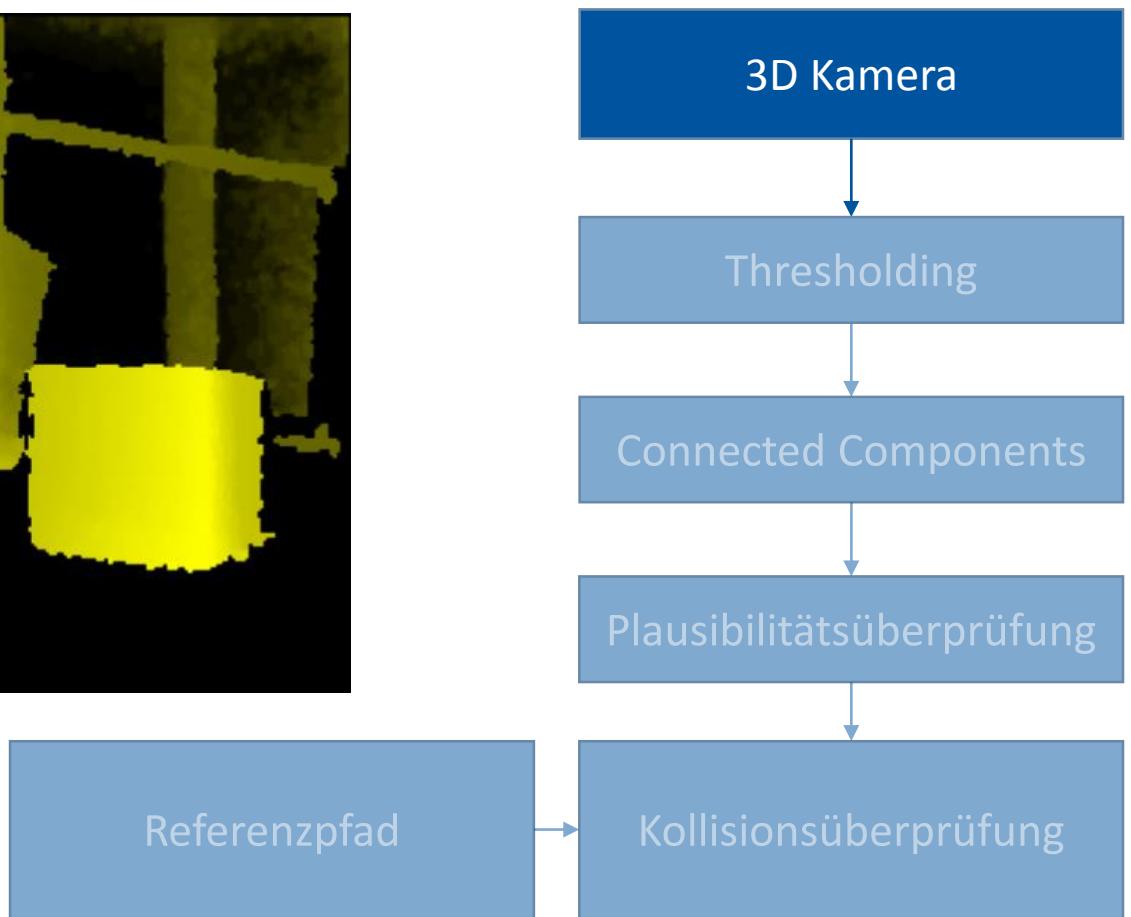
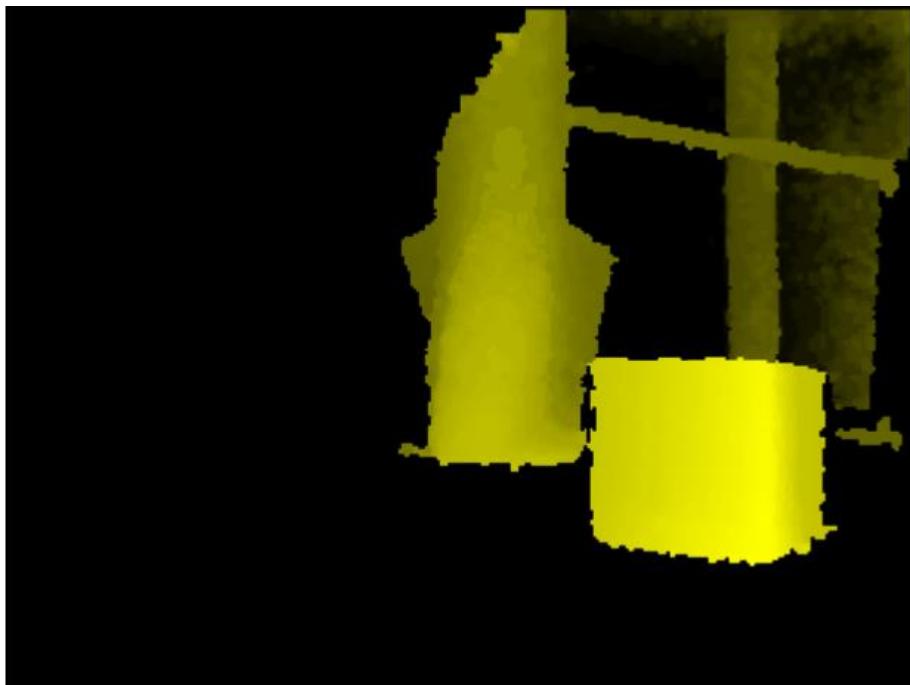
Systemkonzept Hindernisfahrt



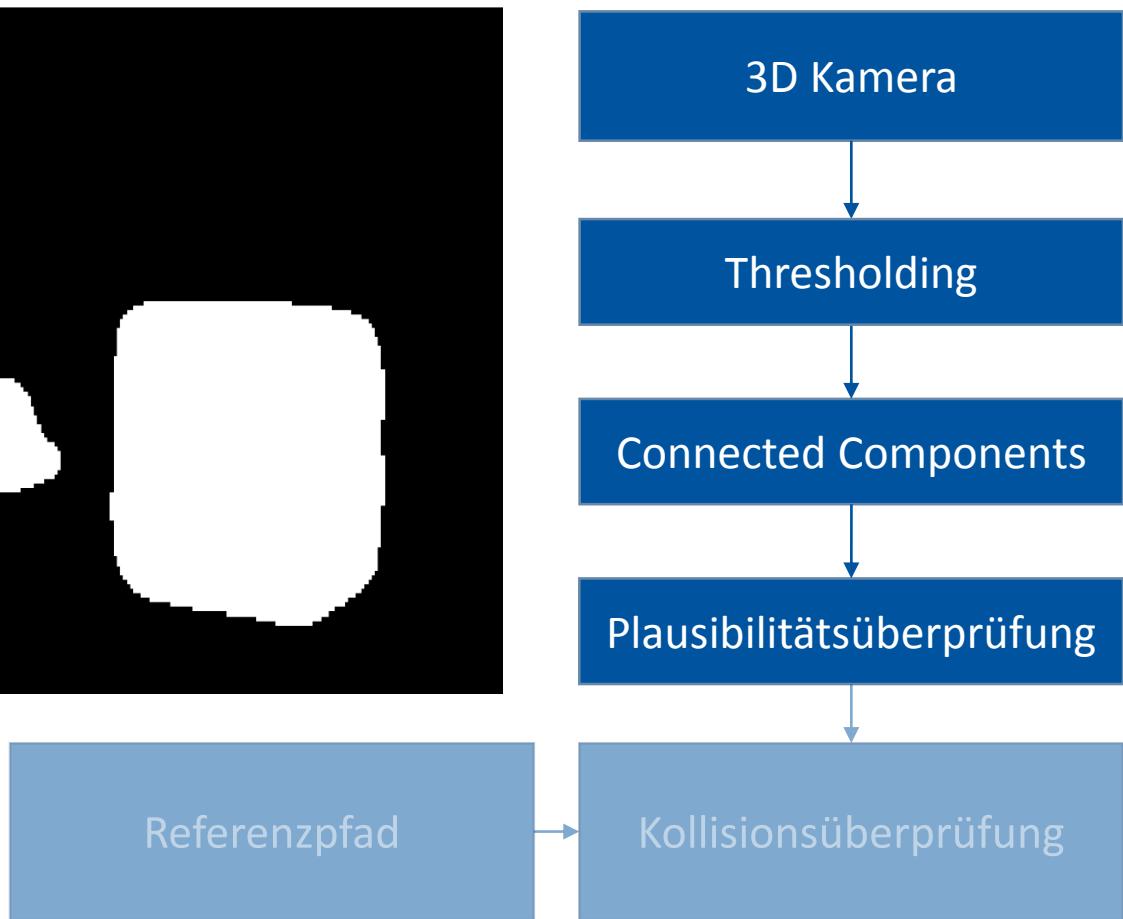
Hinderniserkennung mittels 3D Kamera



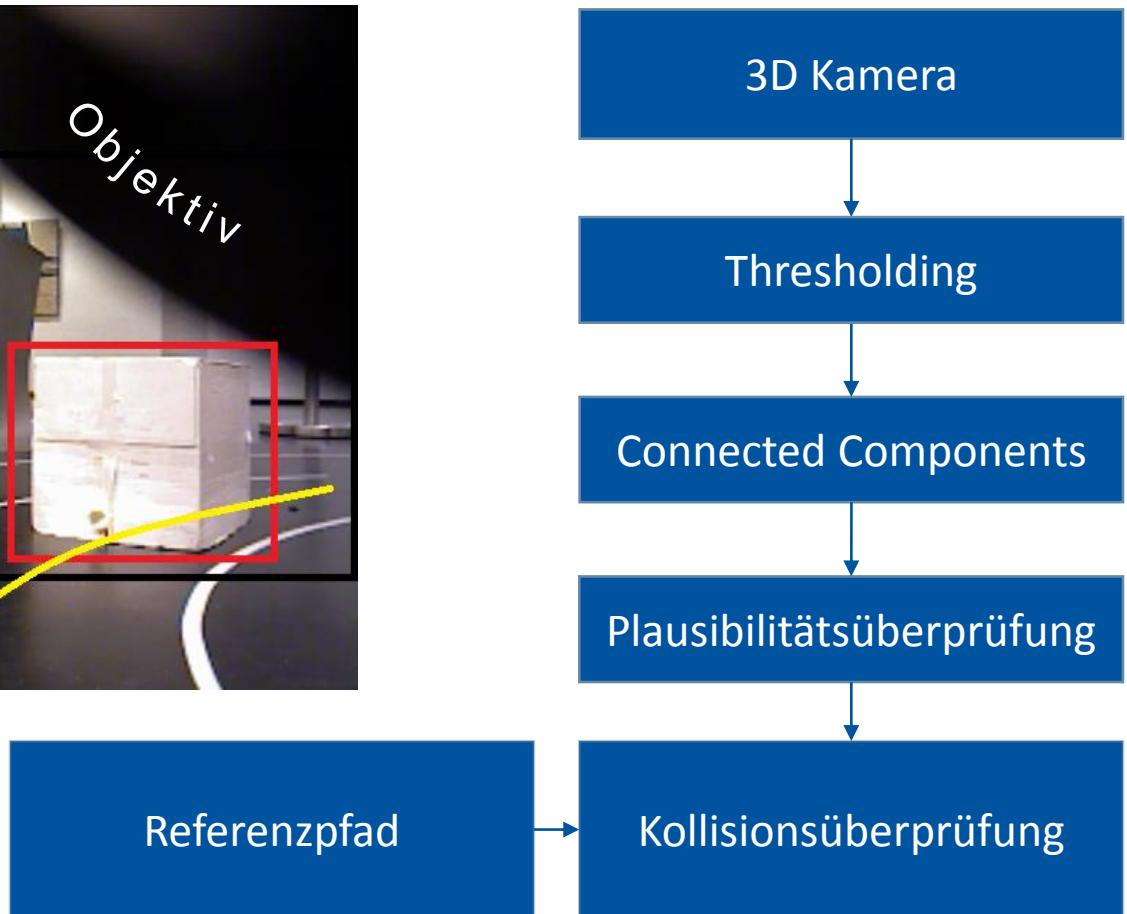
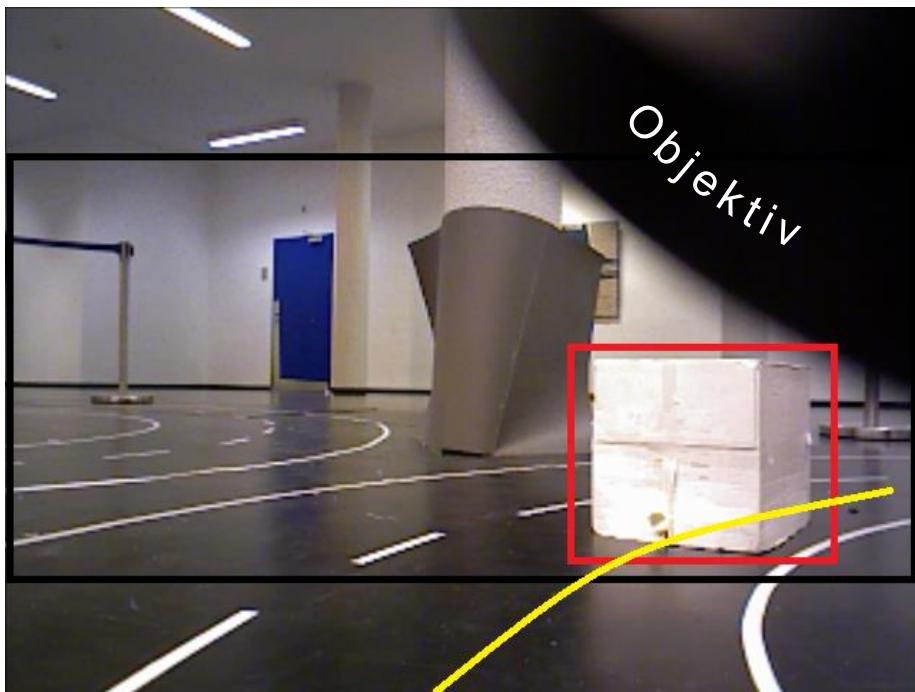
Hinderniserkennung mittels 3D Kamera



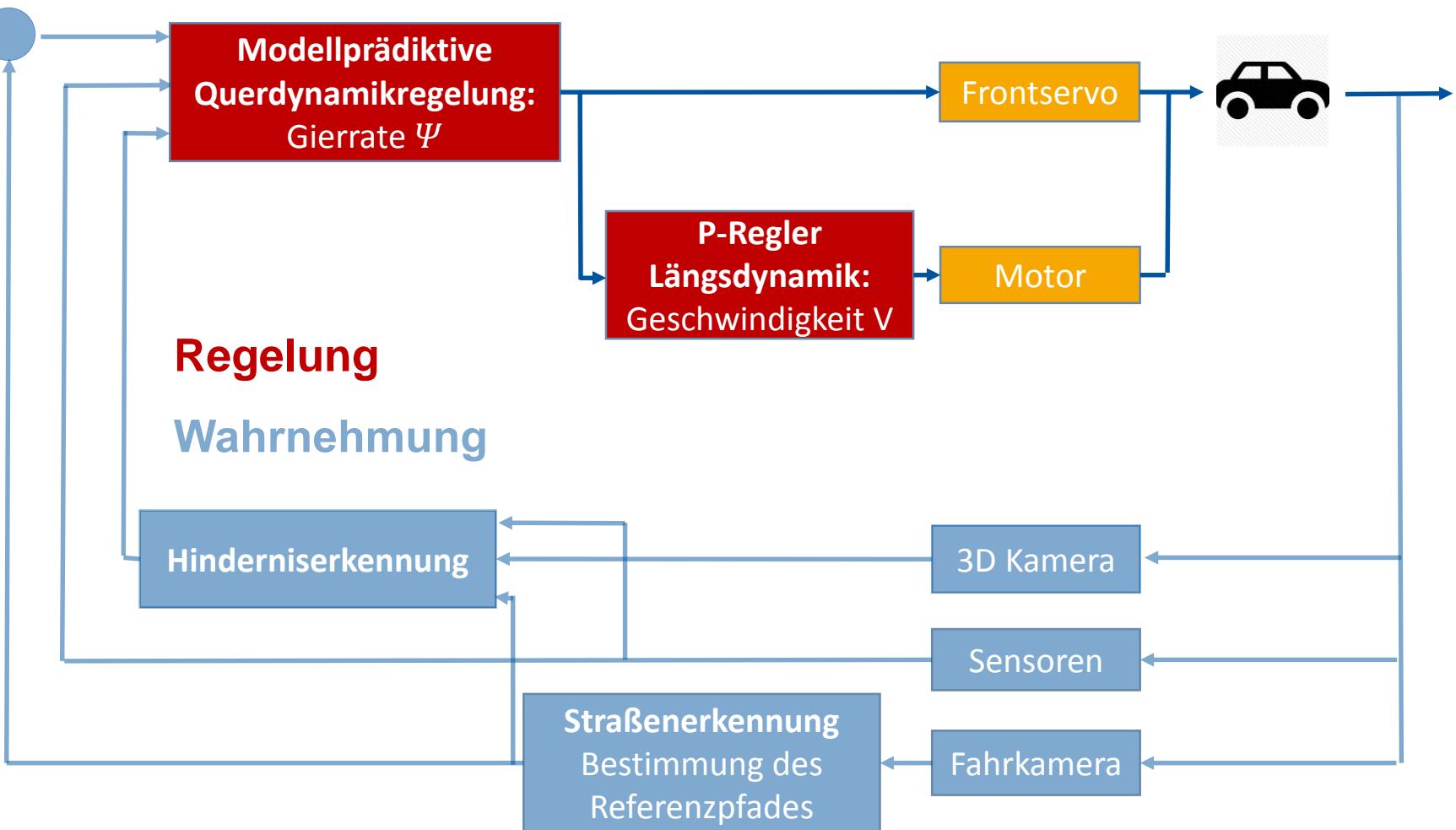
Hinderniserkennung mittels 3D Kamera



Hinderniserkennung mittels 3D Kamera

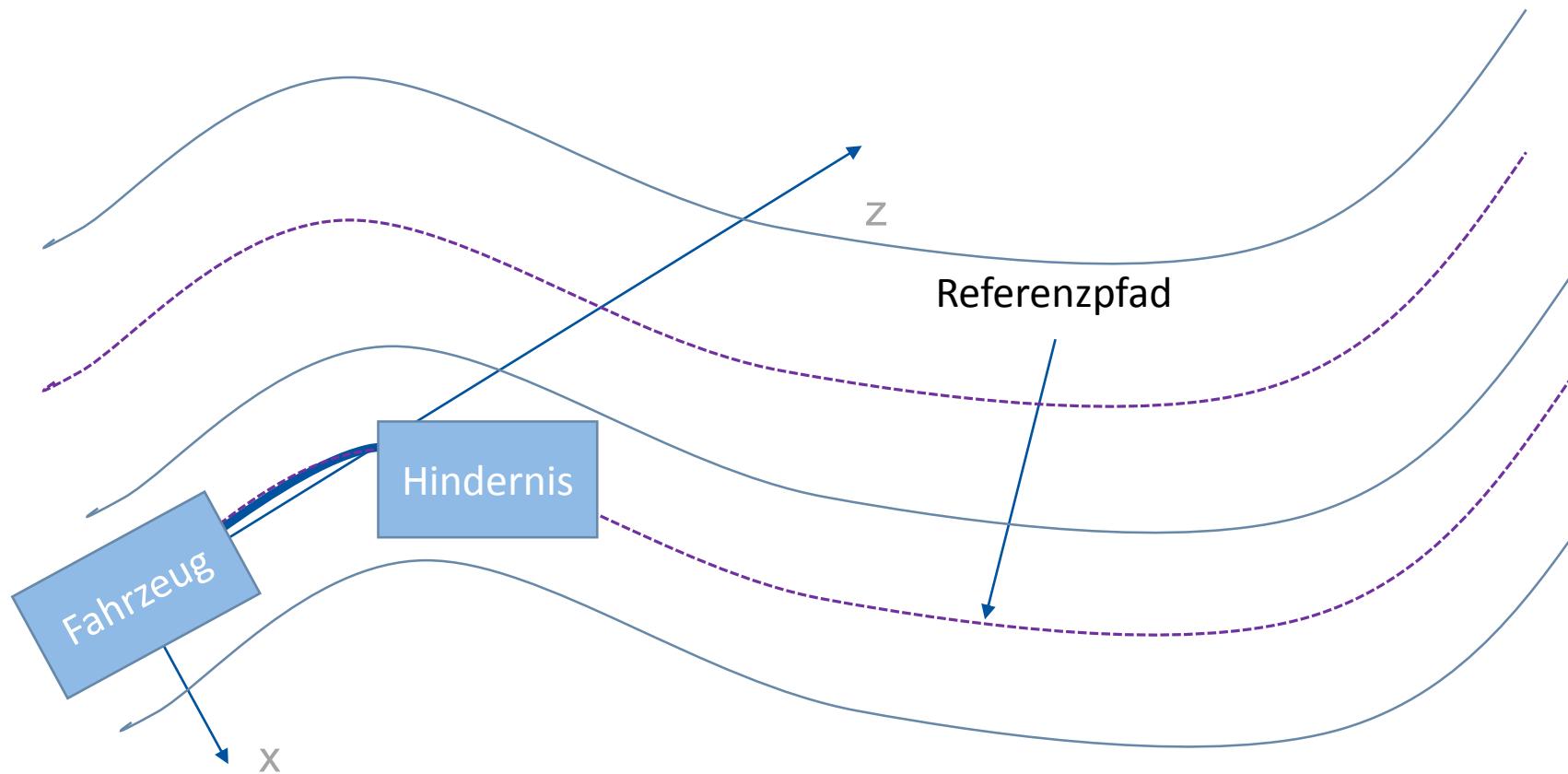


Systemkonzept Hindernisfahrt



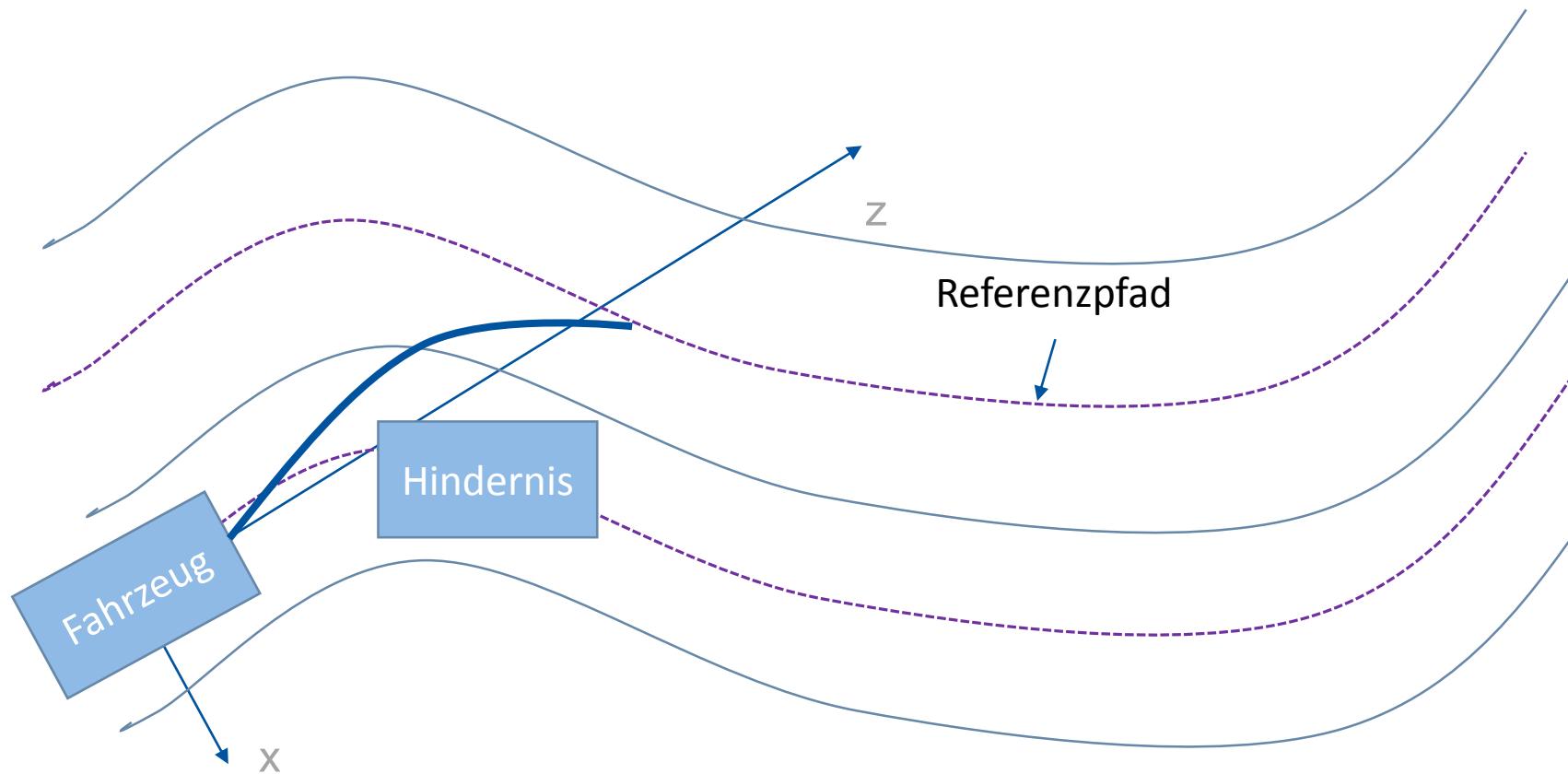
Fahrspurselection

- ▶ Kollision erkannt → linke Fahrspur wird zum Referenzpfad
- ▶ Querdynamikregelung führt Spurwechsel implizit durch



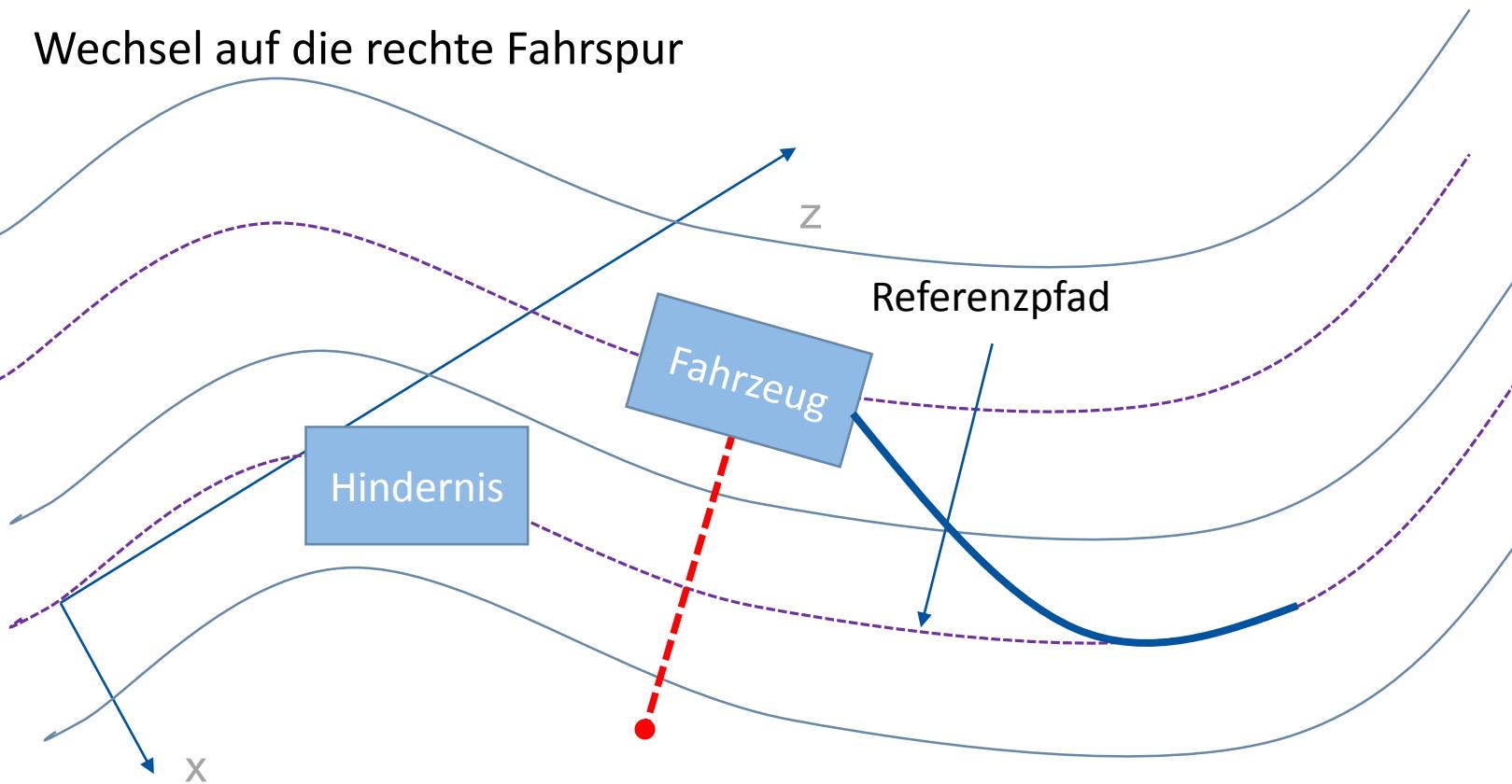
Fahrspurselection

- ▶ Kollision erkannt → linke Fahrspur wird zum Referenzpfad
- ▶ Querdynamikregelung führt Spurwechsel implizit durch



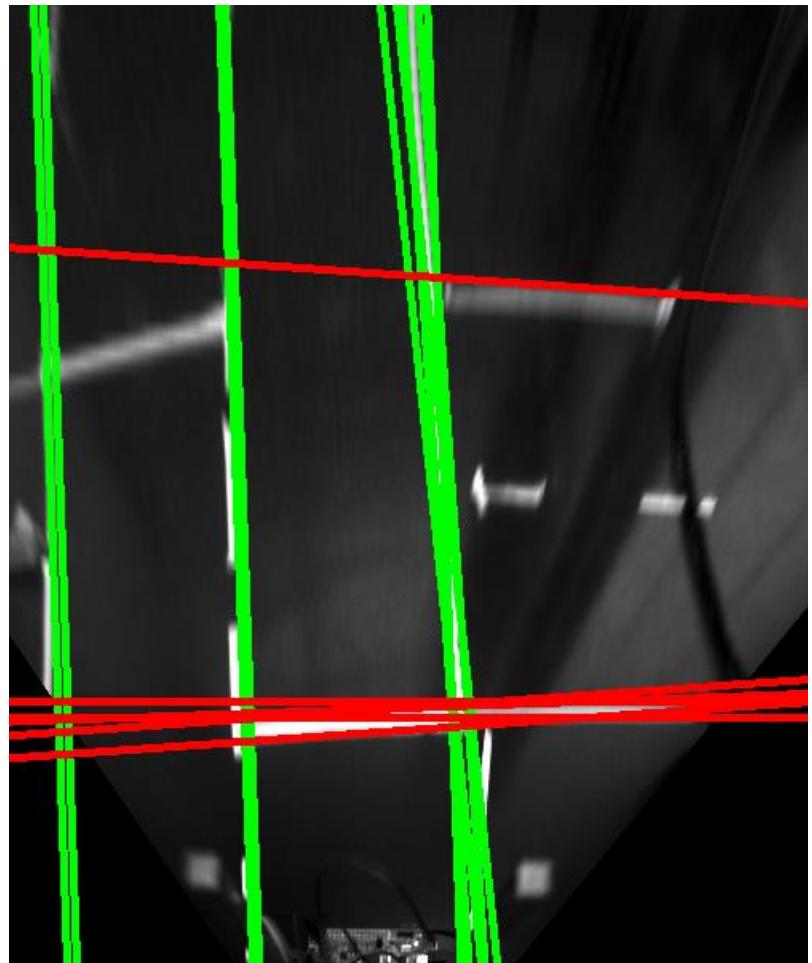
Fahrspurselektion

- ▶ Kollision erkannt → linke Fahrspur wird zum Referenzpfad
- ▶ Querdynamikregelung führt Spurwechsel implizit durch
- ▶ Wechsel auf die rechte Fahrspur



Kreuzungen

- ▶ Erkennung mit Hough Transformation
- ▶ Parameterraum für bessere Performance eingeschränkt



Alternativkonzepte Hindernisfahrt

Alternative Sensorik

- ▶ Infrarotsensor liefert nur punktuelle Information
- ▶ Ultraschall ermöglicht keine exakte Lokalisation

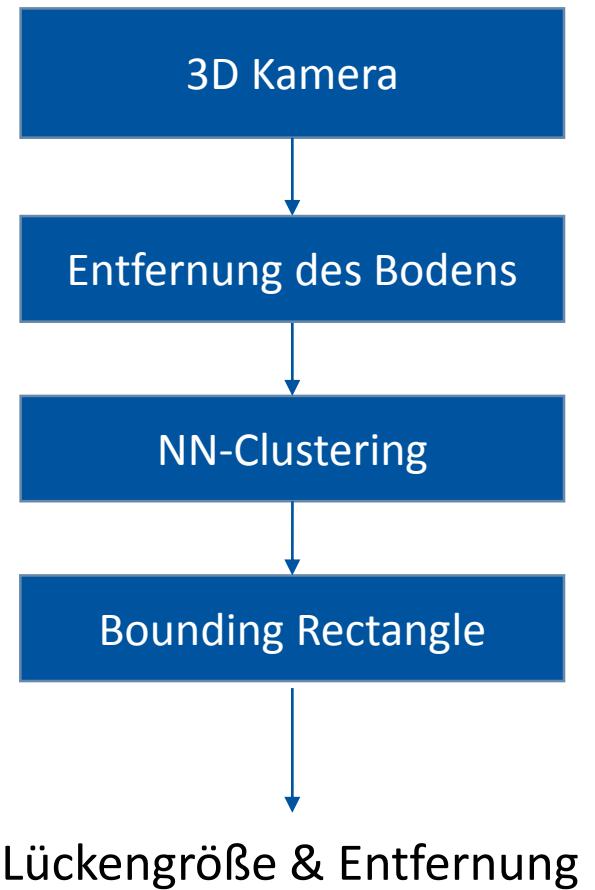
Geplantes Ausweichmanöver

- ▶ Zusätzlicher Rechenaufwand
- ▶ Selbstlokalisierung auf geplantem Pfad notwendig → Fehlerquelle

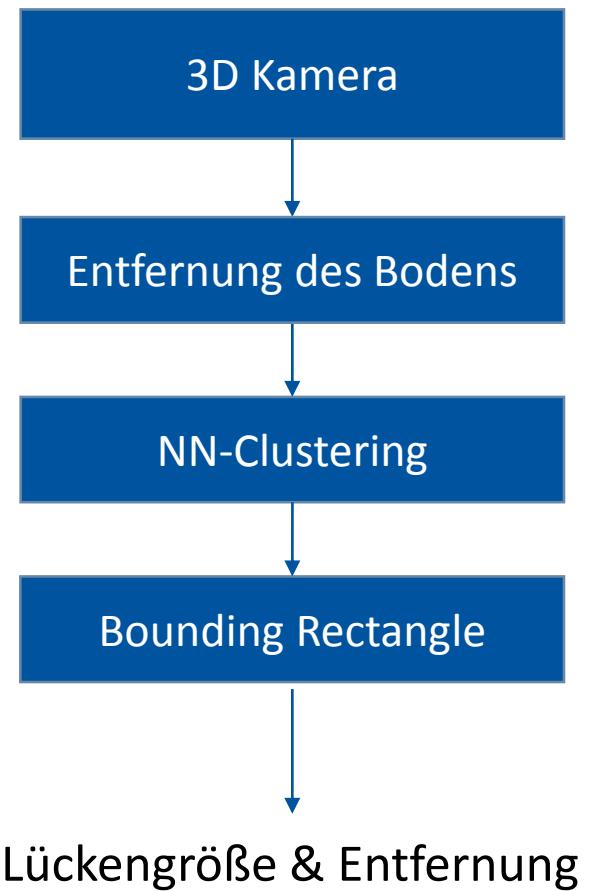
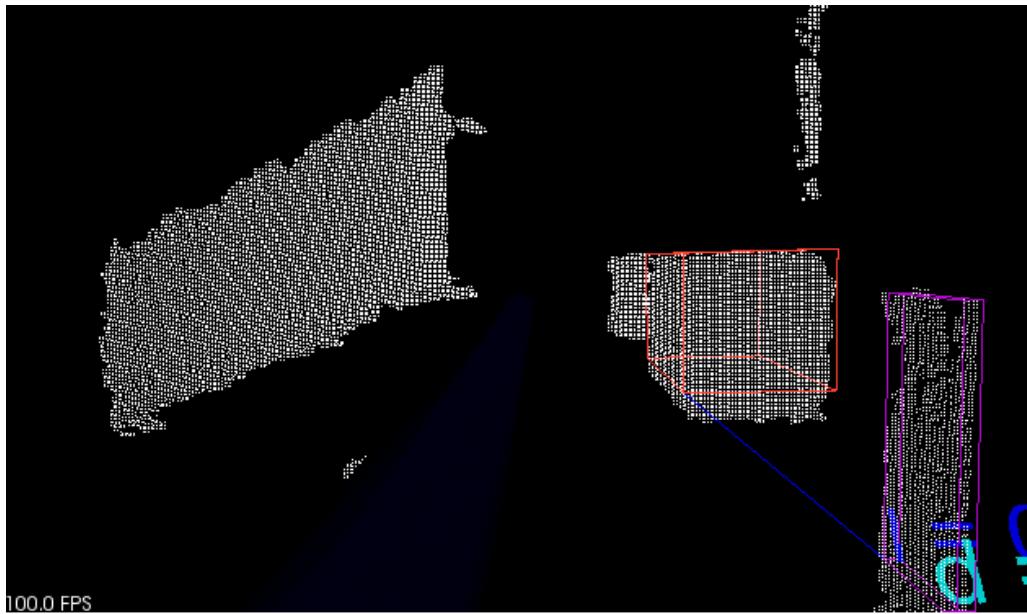
Systemkonzept Vorwärts-Einparken

Team GalaXIs: Hyperion

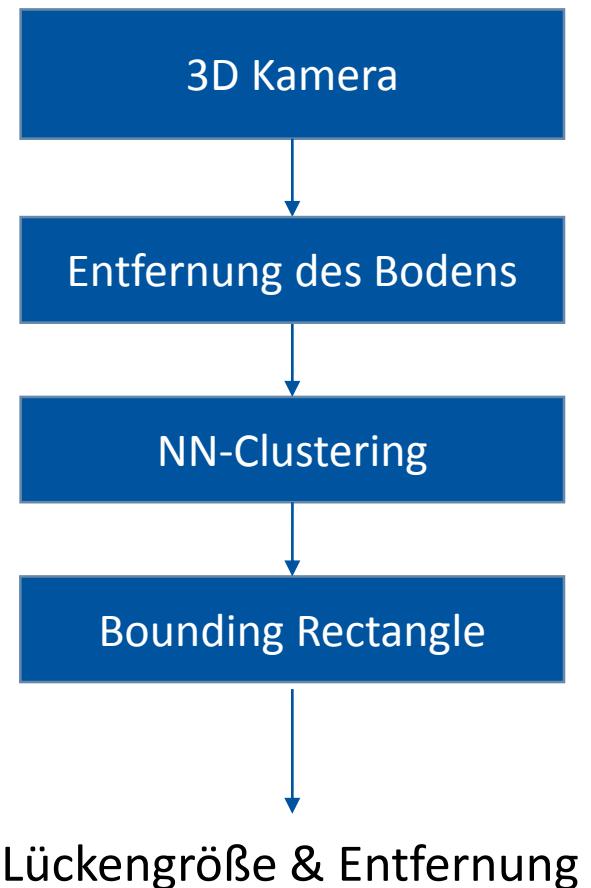
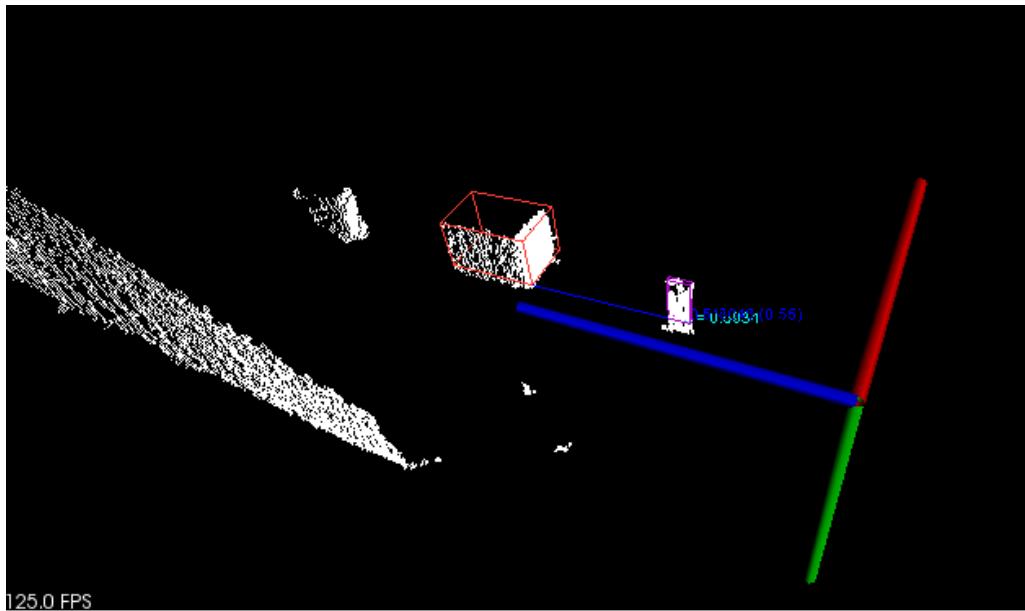
Parklückenerkennung mittels 3D Kamera



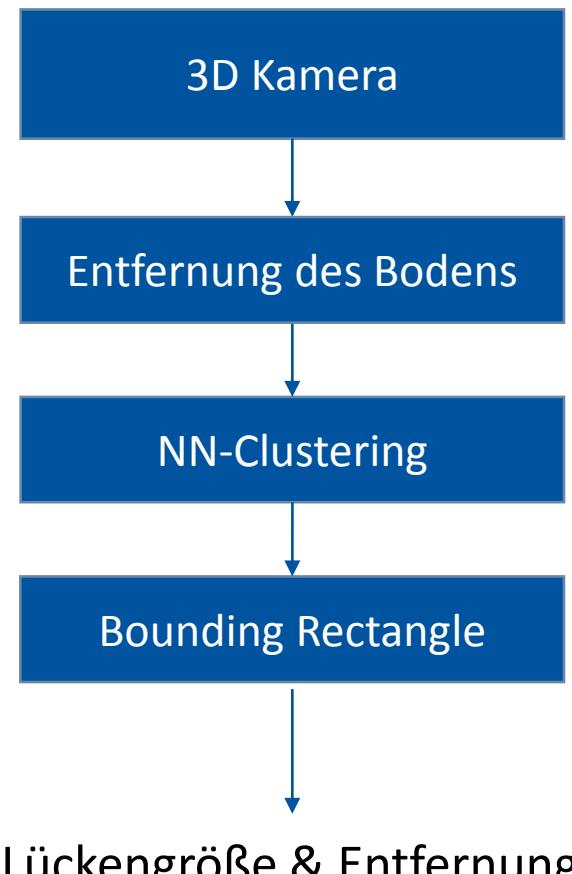
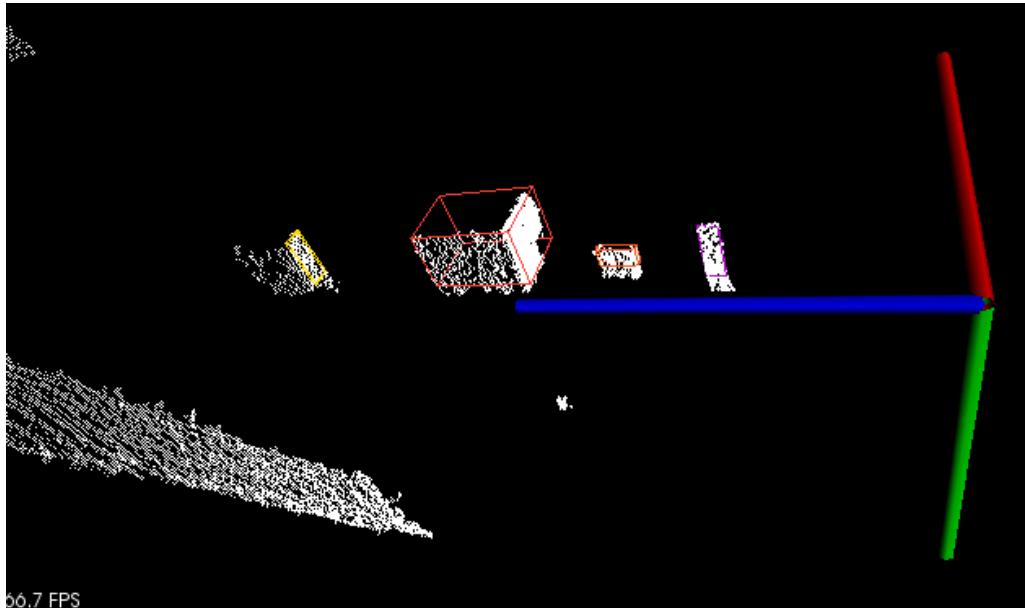
Parklückenerkennung mittels 3D Kamera



Parklückenerkennung mittels 3D Kamera



Parklückenerkennung mittels 3D Kamera



- ▶ 30 FPS
- ▶ Hindernisse werden detektiert

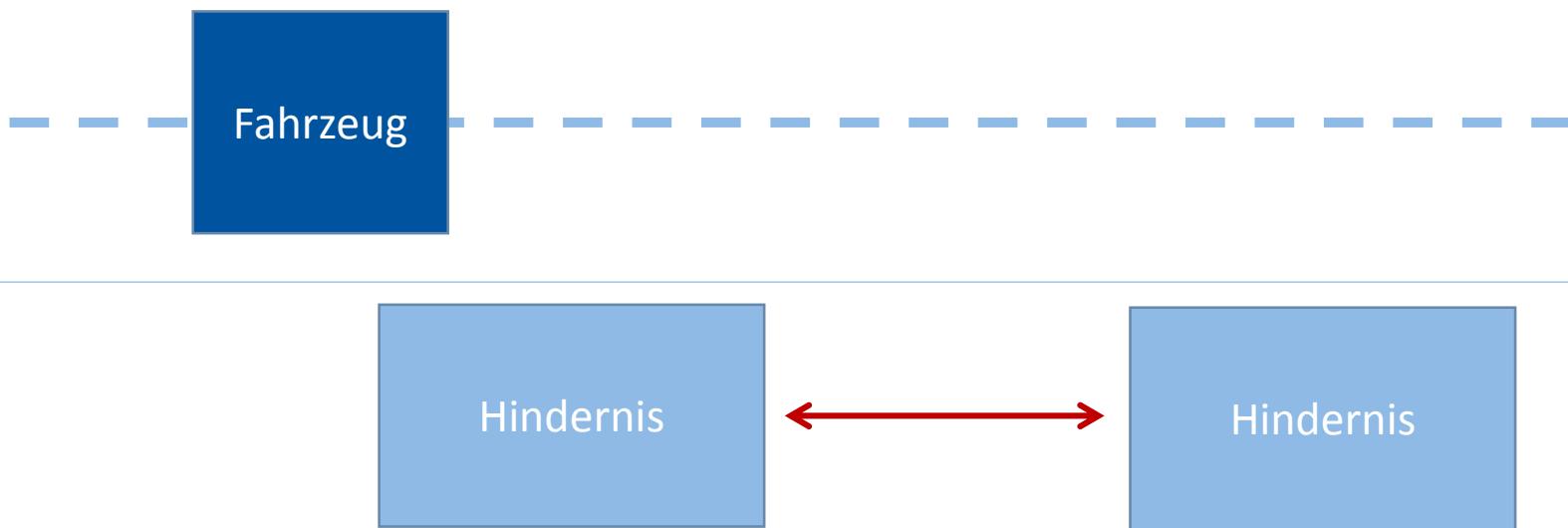
Einparkvorgang

- ▶ Fahrzeug folgt der Fahrspur mittels dem Rundfahrtkonzept
 - ▶ Parallelle Auswertung der 3D Kamera
-



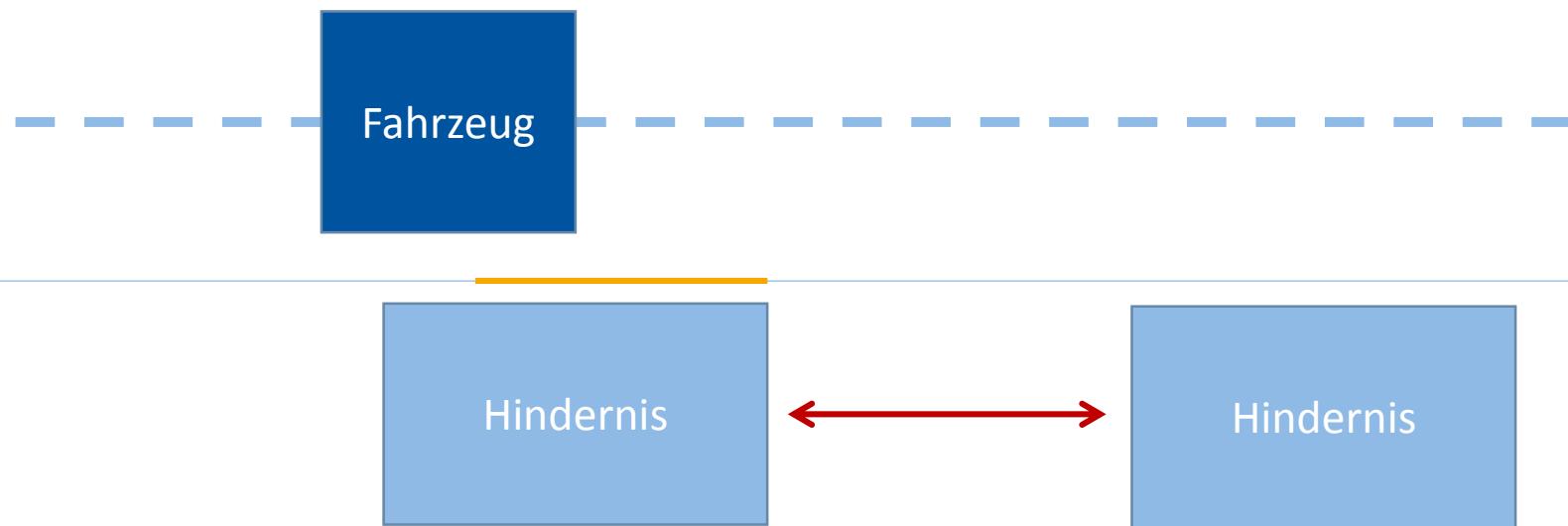
Einparkvorgang

- ▶ Fahrzeug folgt der Fahrspur mittels dem Rundfahrtkonzept
 - ▶ Parallelle Auswertung der 3D Kamera
 - ▶ Parklücke detektiert → Abbremsen
-



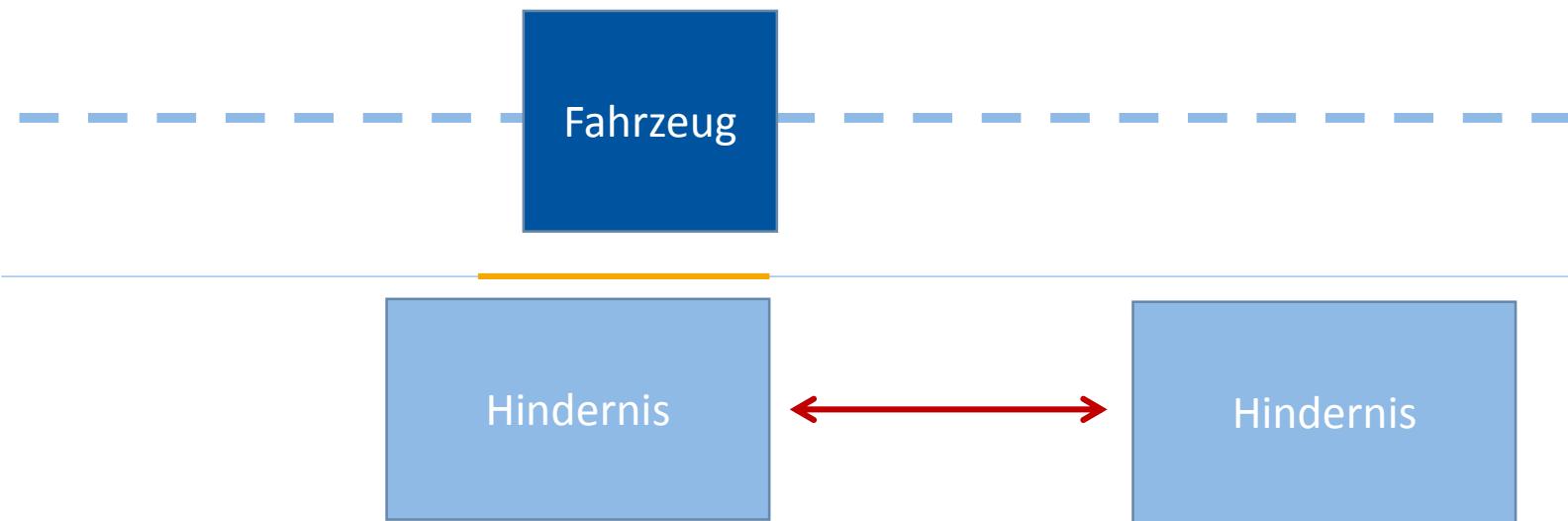
Einparkvorgang

- ▶ Fahrzeug folgt der Fahrspur mittels dem Rundfahrtkonzept
- ▶ Parallelle Auswertung der 3D Kamera
- ▶ Parklücke detektiert → Abbremsen
- ▶ **Entfernung** zum Einschlagzeitpunkt abfahren



Einparkvorgang

- ▶ Fahrzeug folgt der Fahrspur mittels dem Rundfahrtkonzept
- ▶ Parallele Auswertung der 3D Kamera
- ▶ Parklücke detektiert → Abbremsen
- ▶ **Entfernung** zum Einschlagzeitpunkt abfahren
- ▶ Gesteuertes Einparkmanöver unter Verwendung der **Allradlenkung**



Einparkvorgang

- ▶ Fahrzeug folgt der Fahrspur mittels dem Rundfahrtkonzept
 - ▶ Parallele Auswertung der 3D Kamera
 - ▶ Parklücke detektiert → Abbremsen
 - ▶ **Entfernung** zum Einschlagzeitpunkt abfahren
 - ▶ Gesteuertes Einparkmanöver unter Verwendung der **Allradlenkung**
-



Alternativkonzepte Einparken

Alternativkonzept: Manöverplanung & geregeltes Manöver

- ▶ Dubins Car / Continuous Curvature Steering
- ▶ Regelung erfordert exakte Lokalisation auf geplantem Pfad

- ▶ Kurzer Radstand & lenkbare Hinterachse erleichtern Manöver
- ▶ Lückengröße im Vorfeld bekannt & konstant

Vielen Dank für Ihre Aufmerksamkeit

