



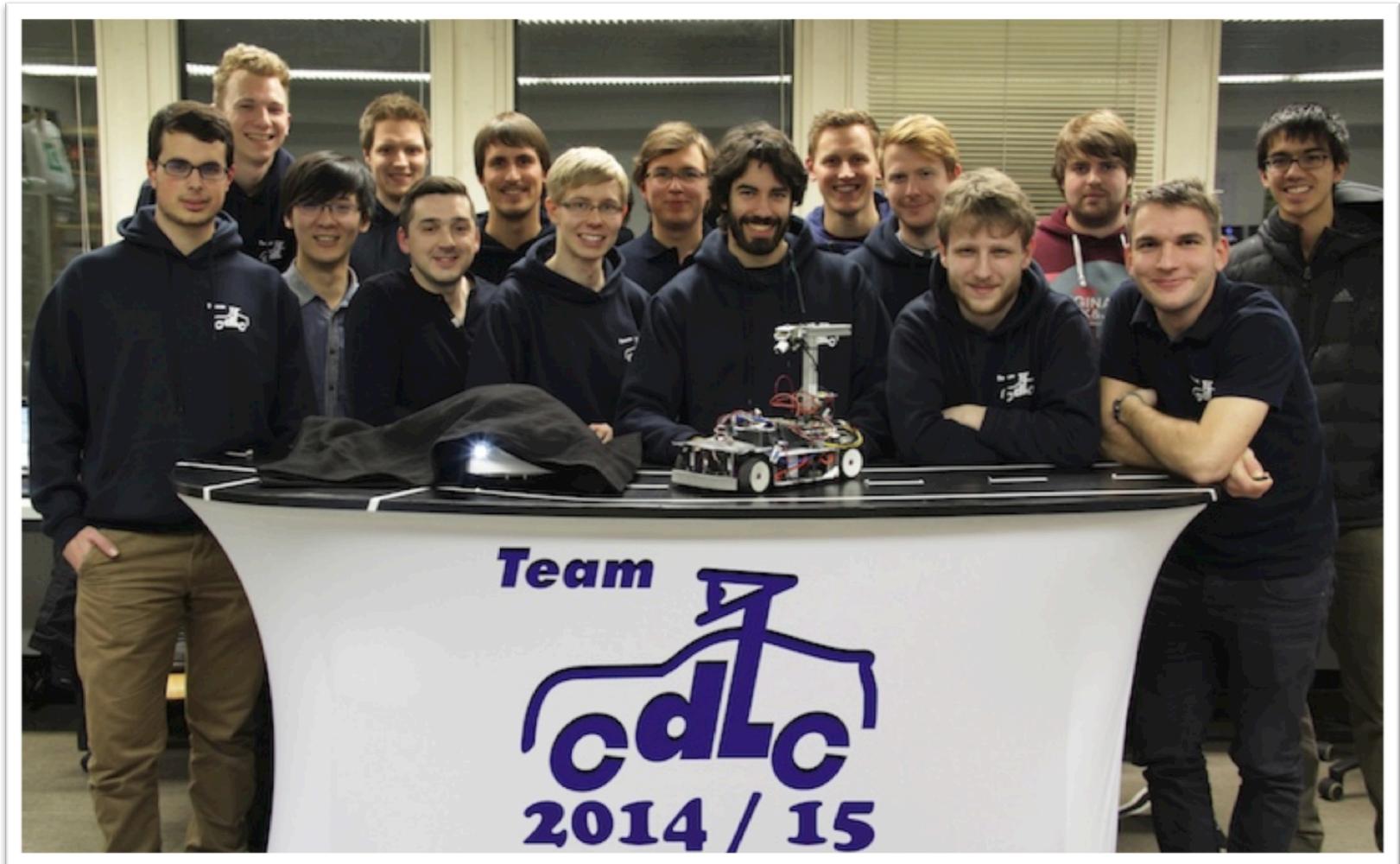
Technische
Universität
Braunschweig



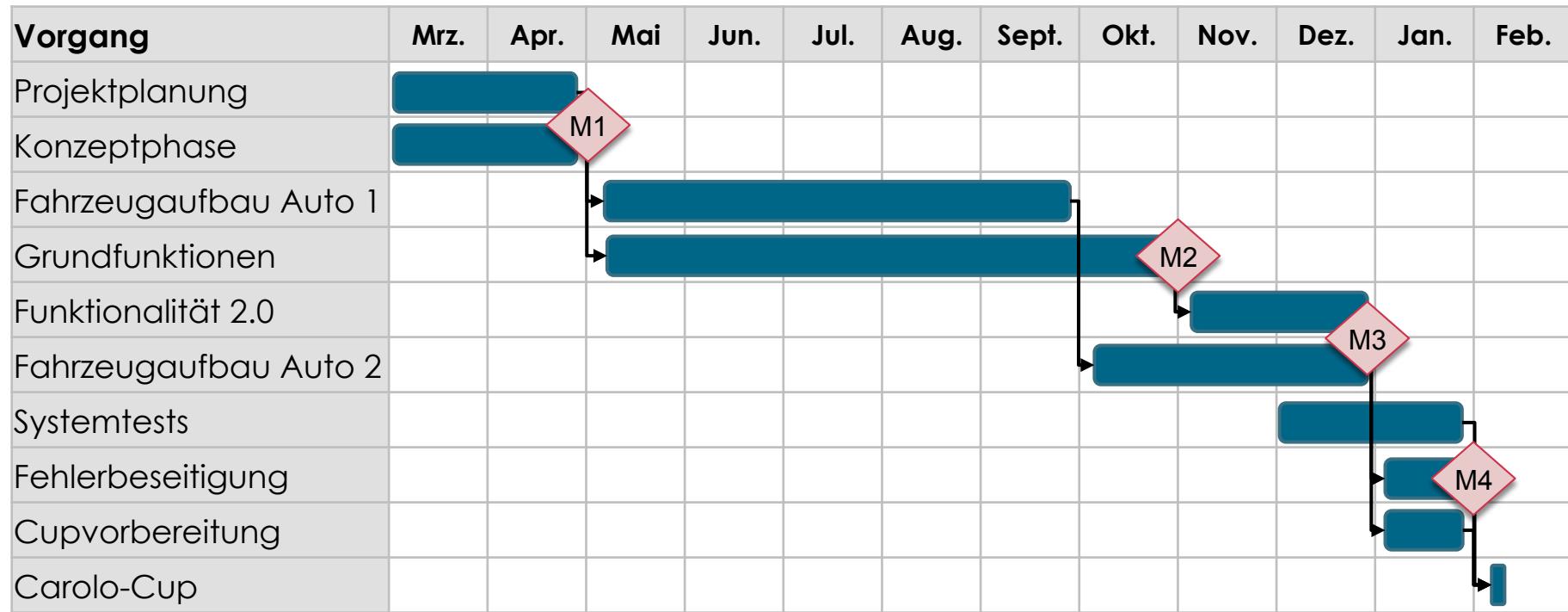
Carolinchen VII



Das Team



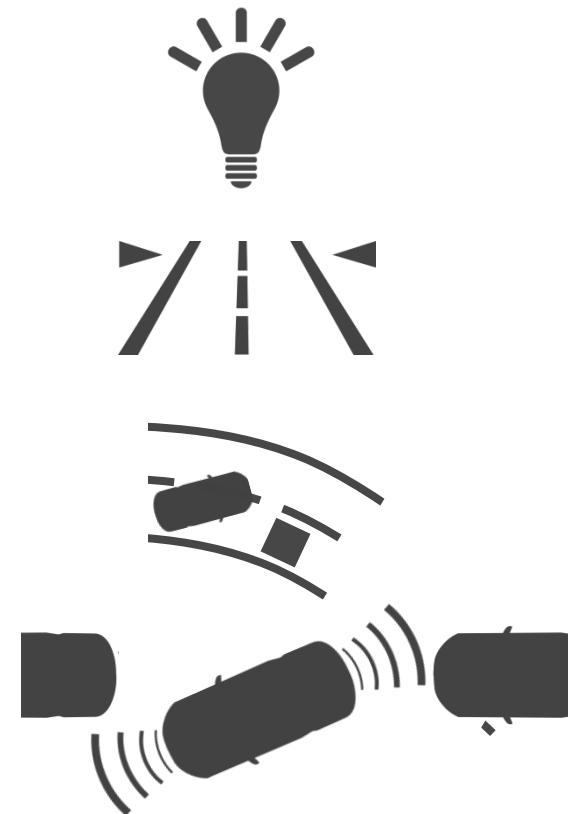
Projektmanagement



Meilensteine:

- M1: Konzeptfreigabe
- M2: Prototyp
- M3: Entwicklungsabschluss
- M4: Cup-Freigabe

Gliederung



Gliederung

Konzeption



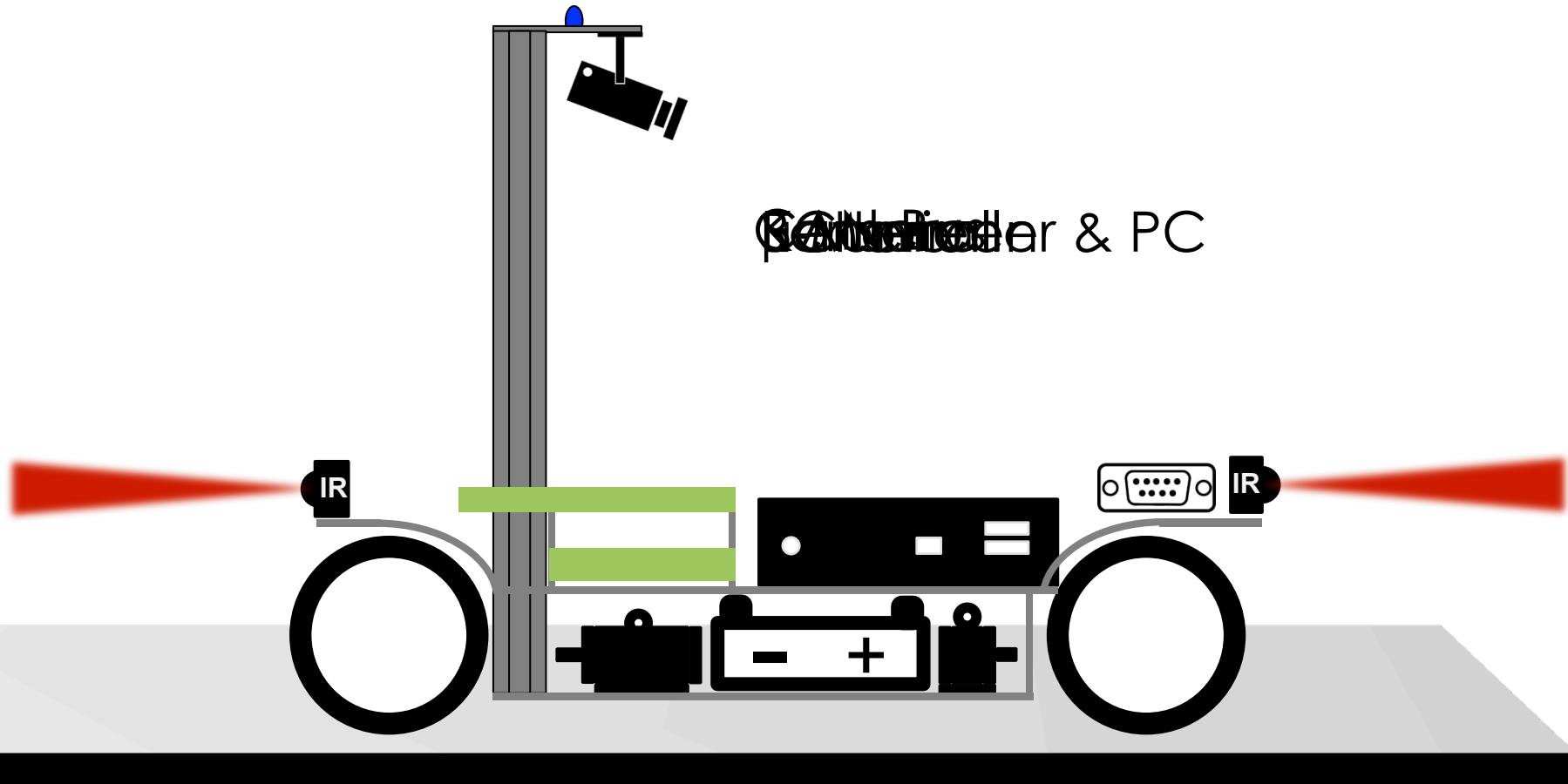
Freie Fahrt

Hindernisfahrt

Parken

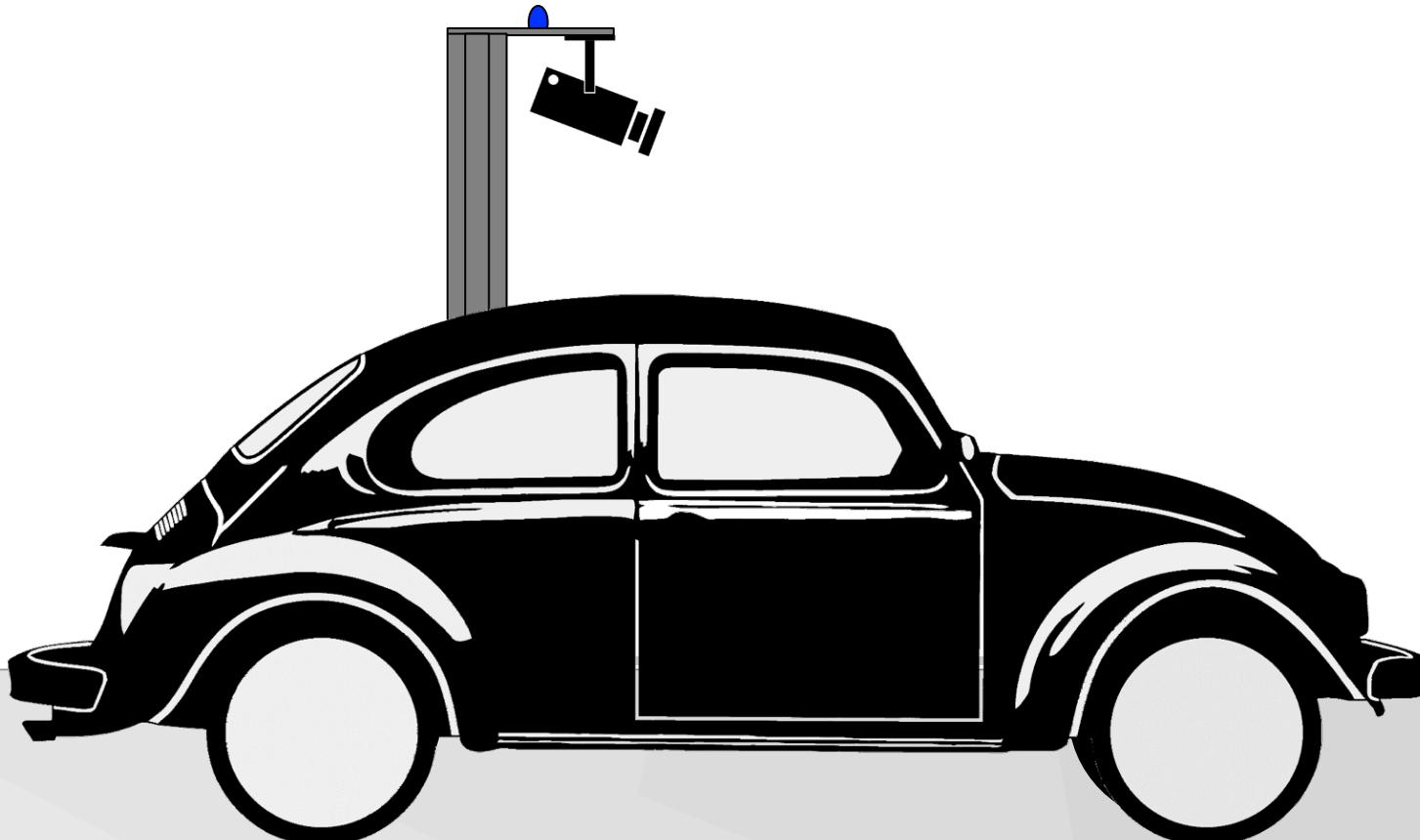


Hardware - Fahrzeugaufbau



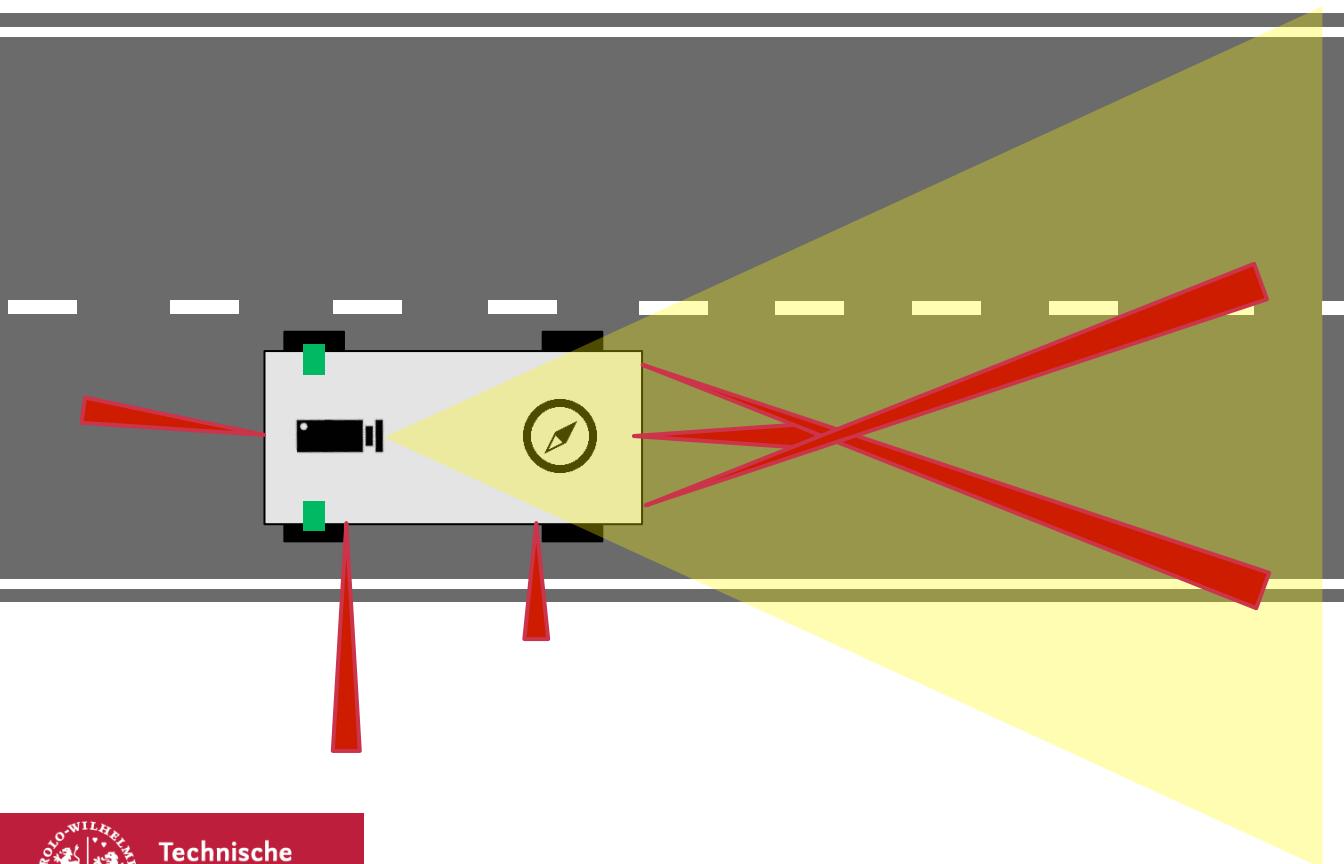


Hardware - Fahrzeugaufbau





Hardware - Sensorik

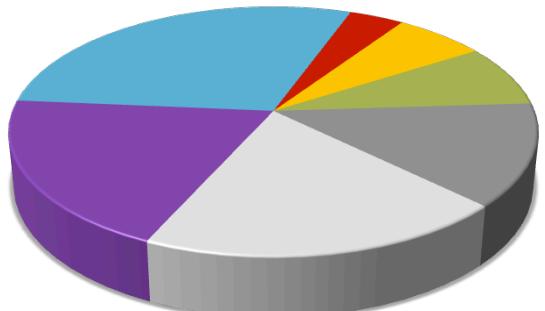


- Kamera
- Inkrementalgeber
- Inertialsensor
- Infrarot

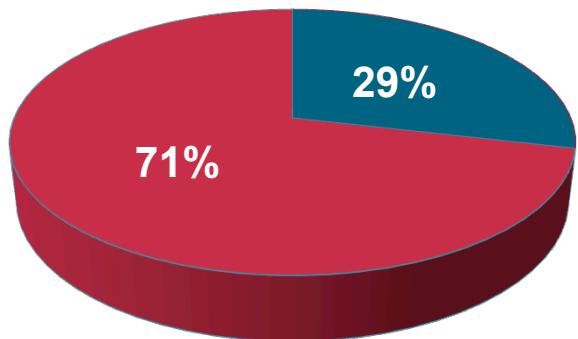


Kosten

Gesamtkosten



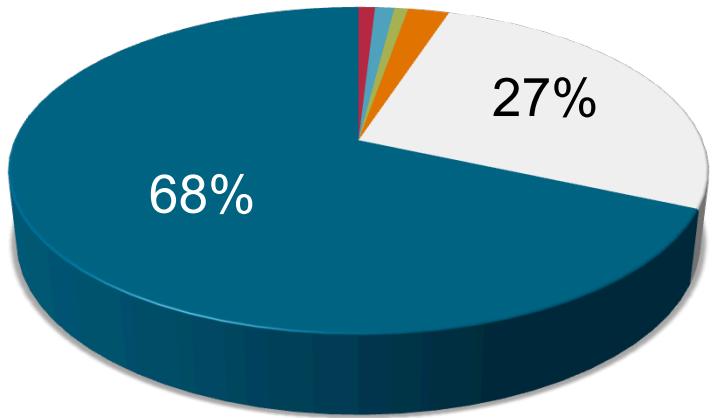
Förderung



Komponente	Preis
μC-Platine	87 €
Energieversorgung	145 €
Aktoren	180 €
Sonstiges	295 €
Chassis	435 €
Sensoren	438 €
Computer	650 €
Summe	2.230 €
Unterstützung	1.590 €
Eigenleistung	640 €



Energiebilanz

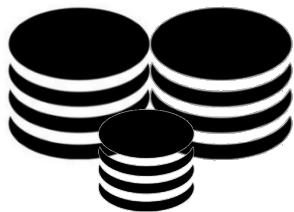


Komponente	Leistung
■ µController	1,2 Watt
■ IR-Sensoren	1,4 Watt
■ Kamera	1,0 Watt
■ Licht	3,0 Watt
□ Hauptrechner	35 Watt
■ Aktorik	90 Watt
Σ Fahrt	131,6 Watt



Evolution

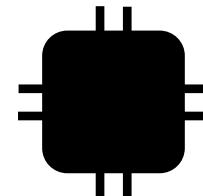
VI



3.700€



3.084g



8 bit

-35 %



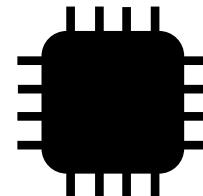
VII



2.350€



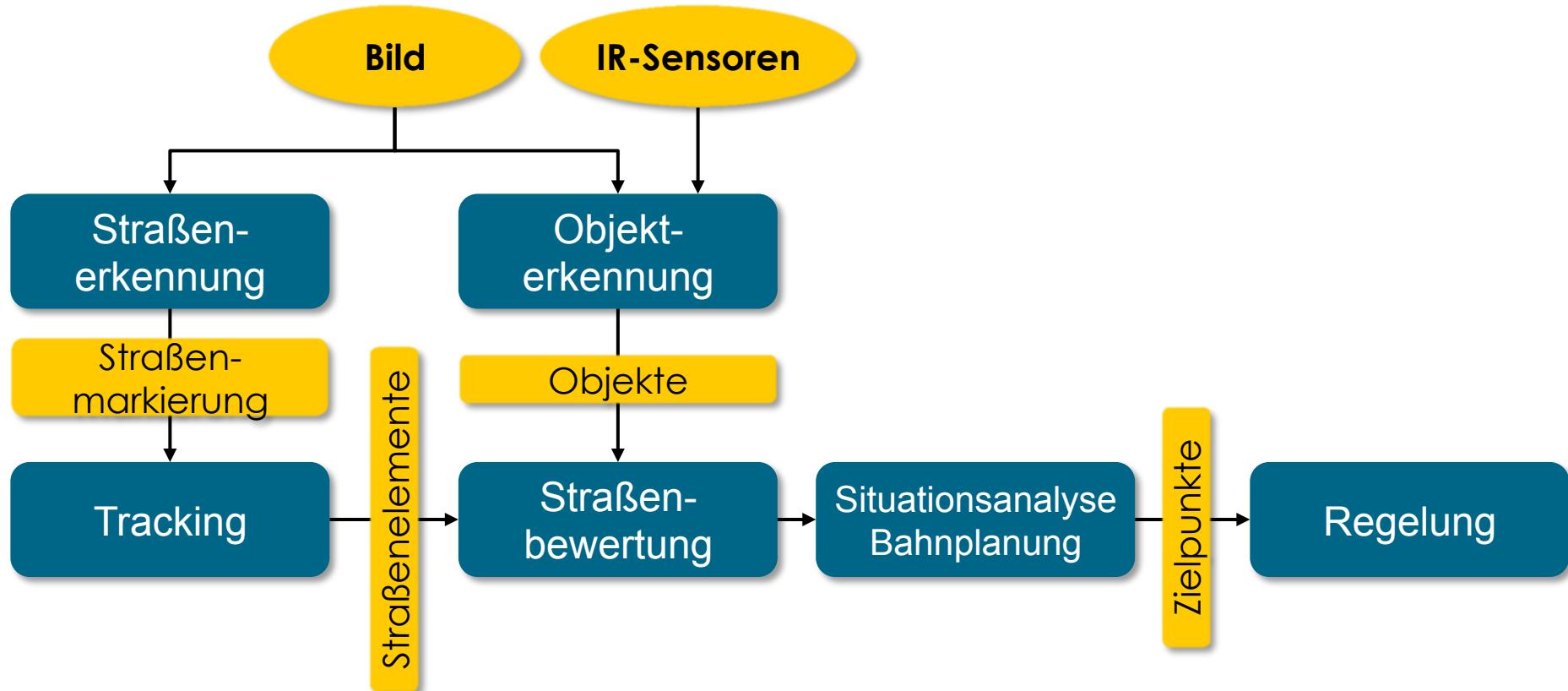
2.255g



32 bit



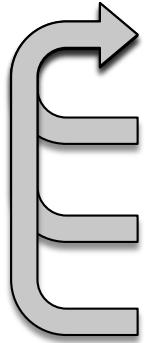
Softwarearchitektur





Qualitätssicherung

Vorgehen:



1. Rapid Prototyping mittels Python
 2. Programmierung in C++ → Codezeilen: ca. 65.000
 3. Unitests
 4. Modultests
 5. Systemtests
- Simulationen in ADTF: **420 km**
Testparcours: **52 km**

Gliederung

Konzeption

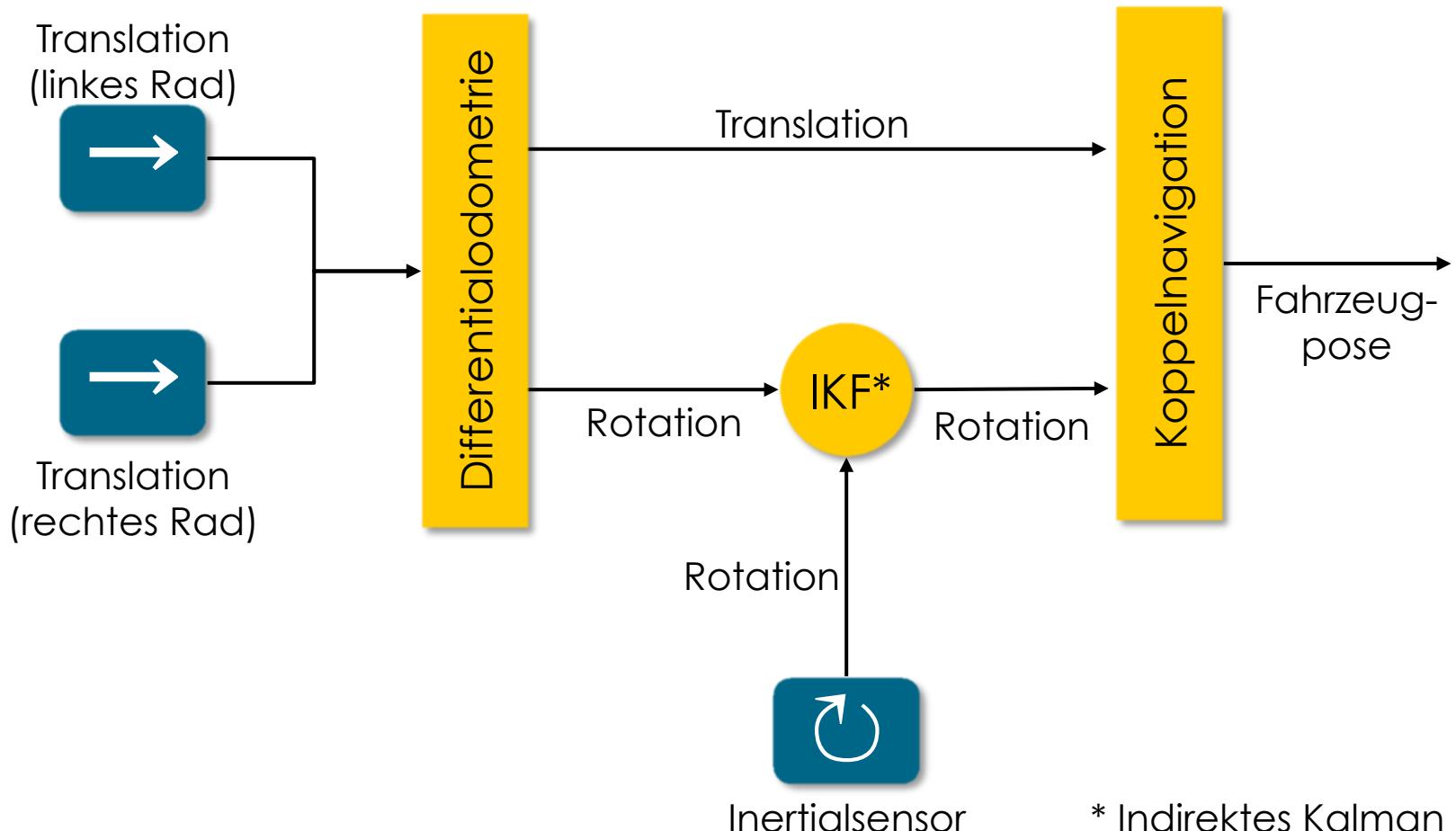
Freie Fahrt



Hindernisfahrt

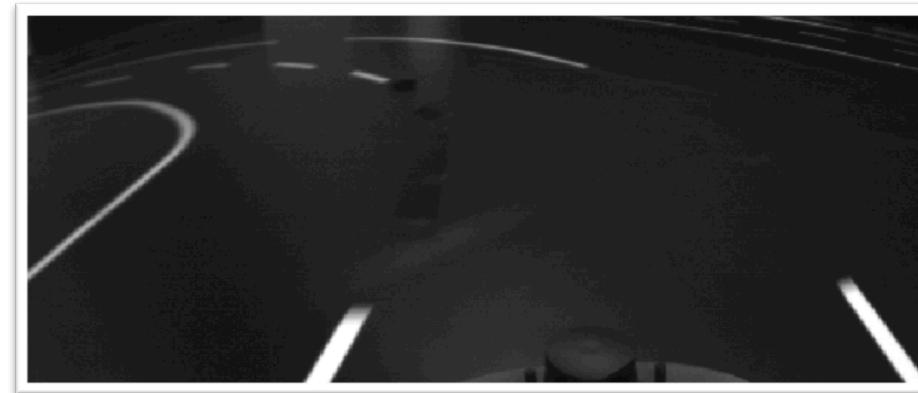
Parken

Lokalisierung

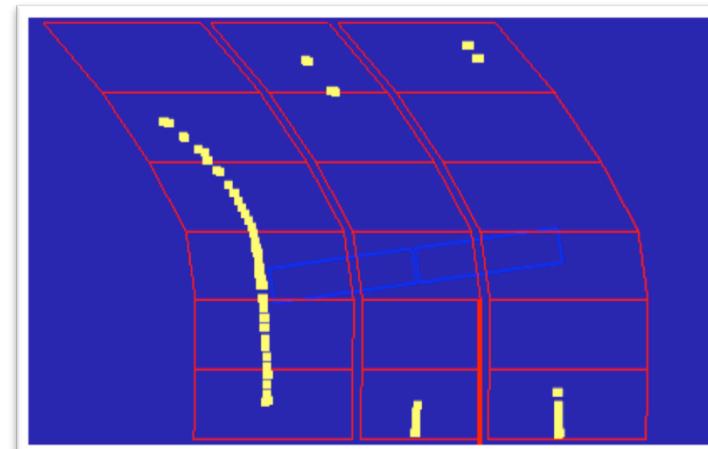


* Indirektes Kalman Filter

Straßenerkennung

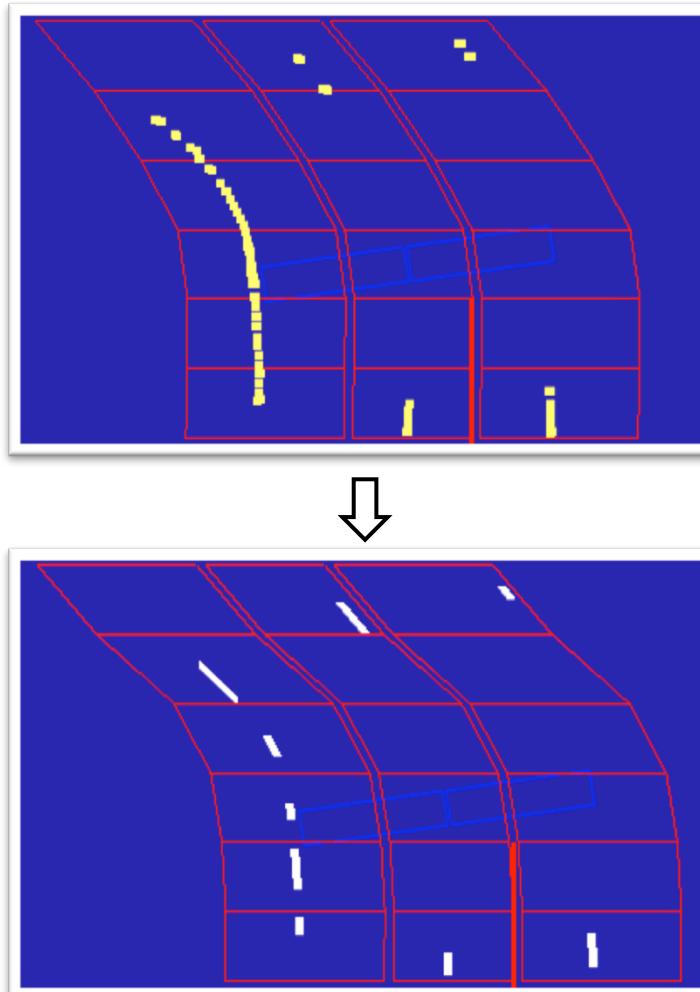
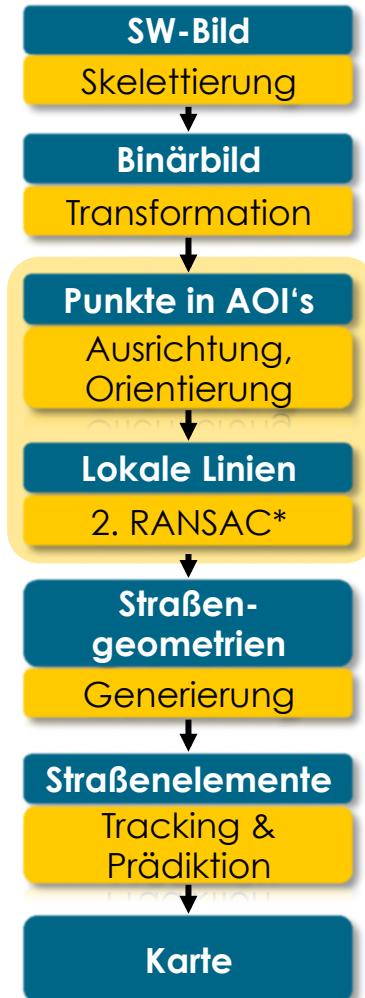


Straßenerkennung



* Area of Interest

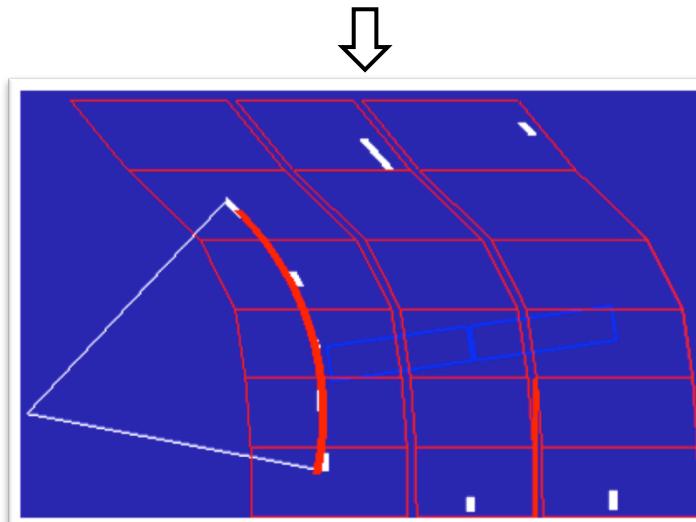
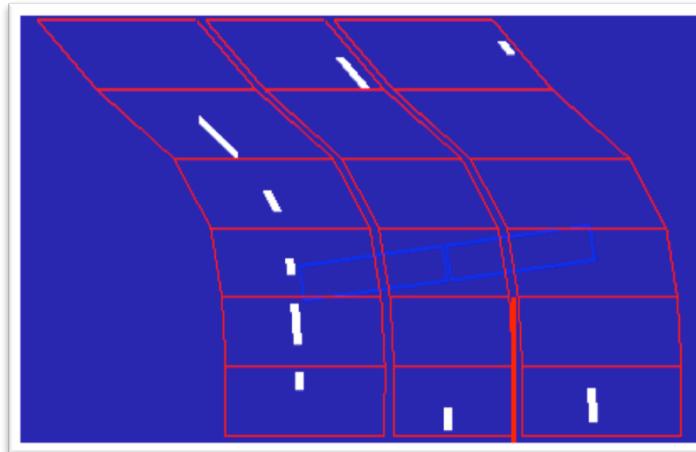
Straßenerkennung



* Random Sample and Consensus

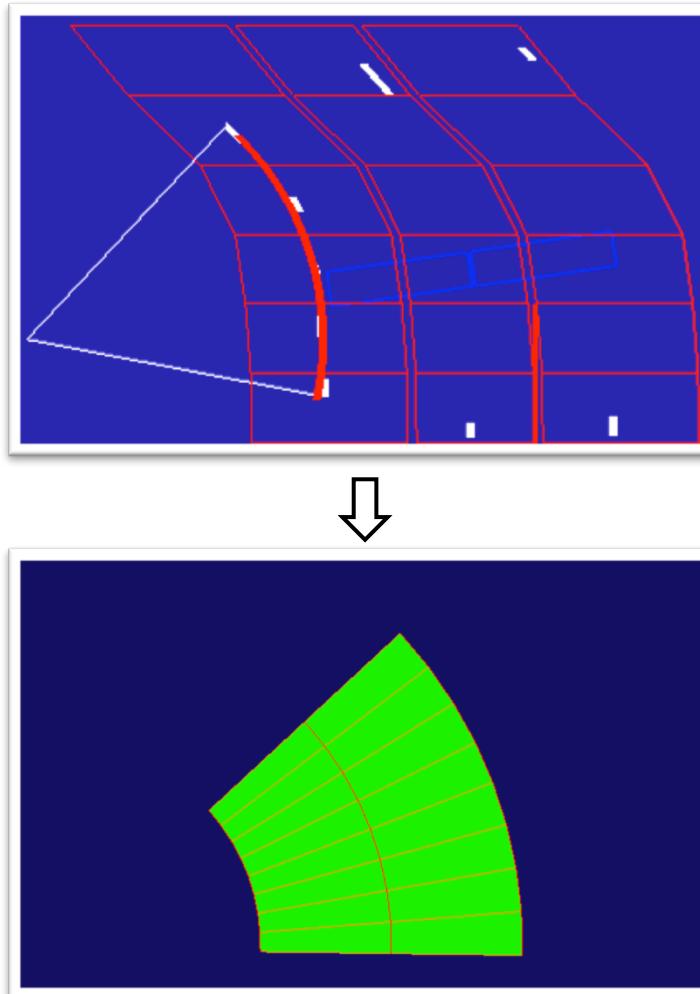
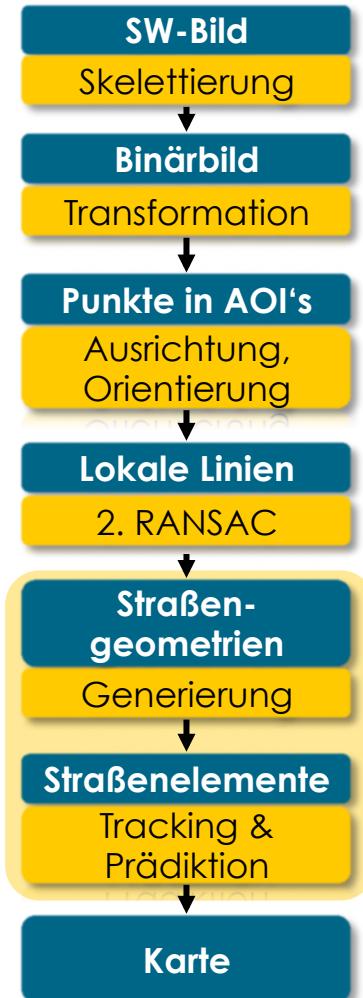


Straßenerkennung

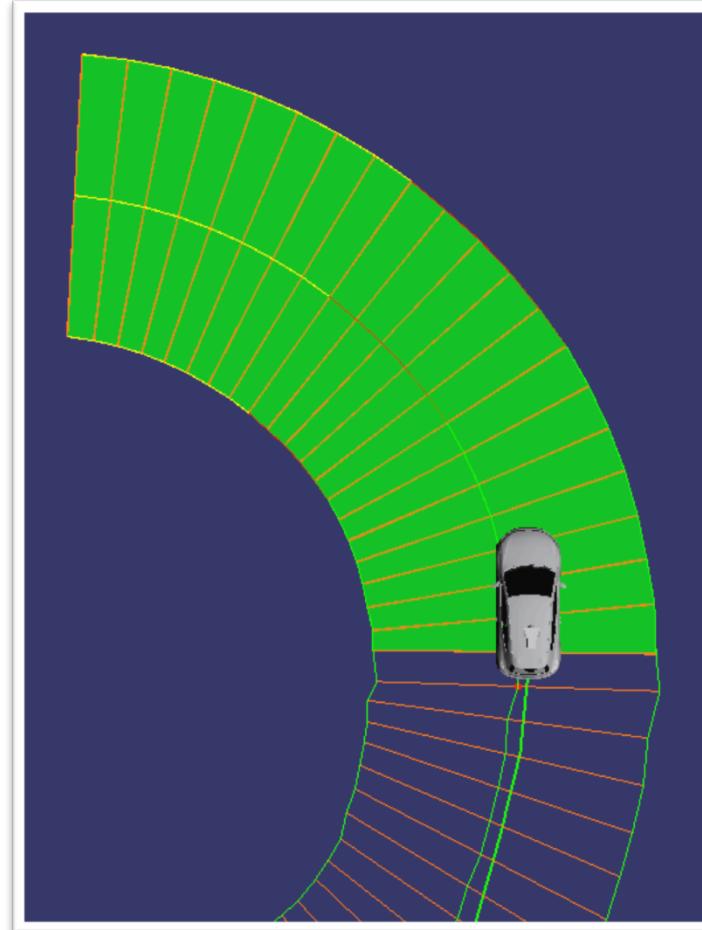


* Random Sample and Consensus

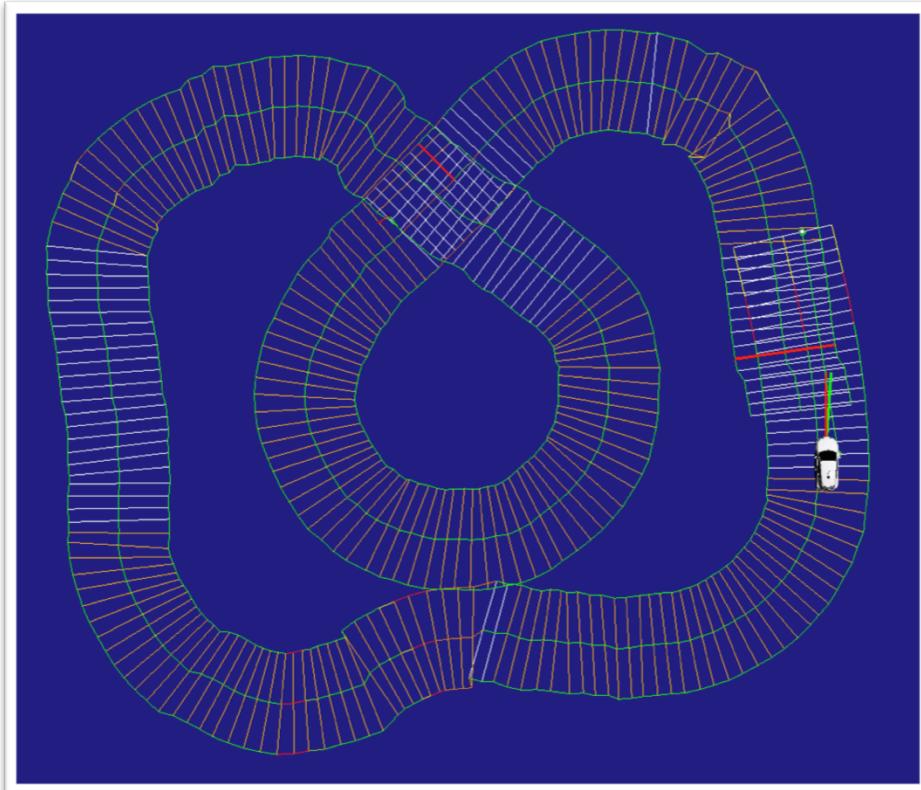
Straßenerkennung



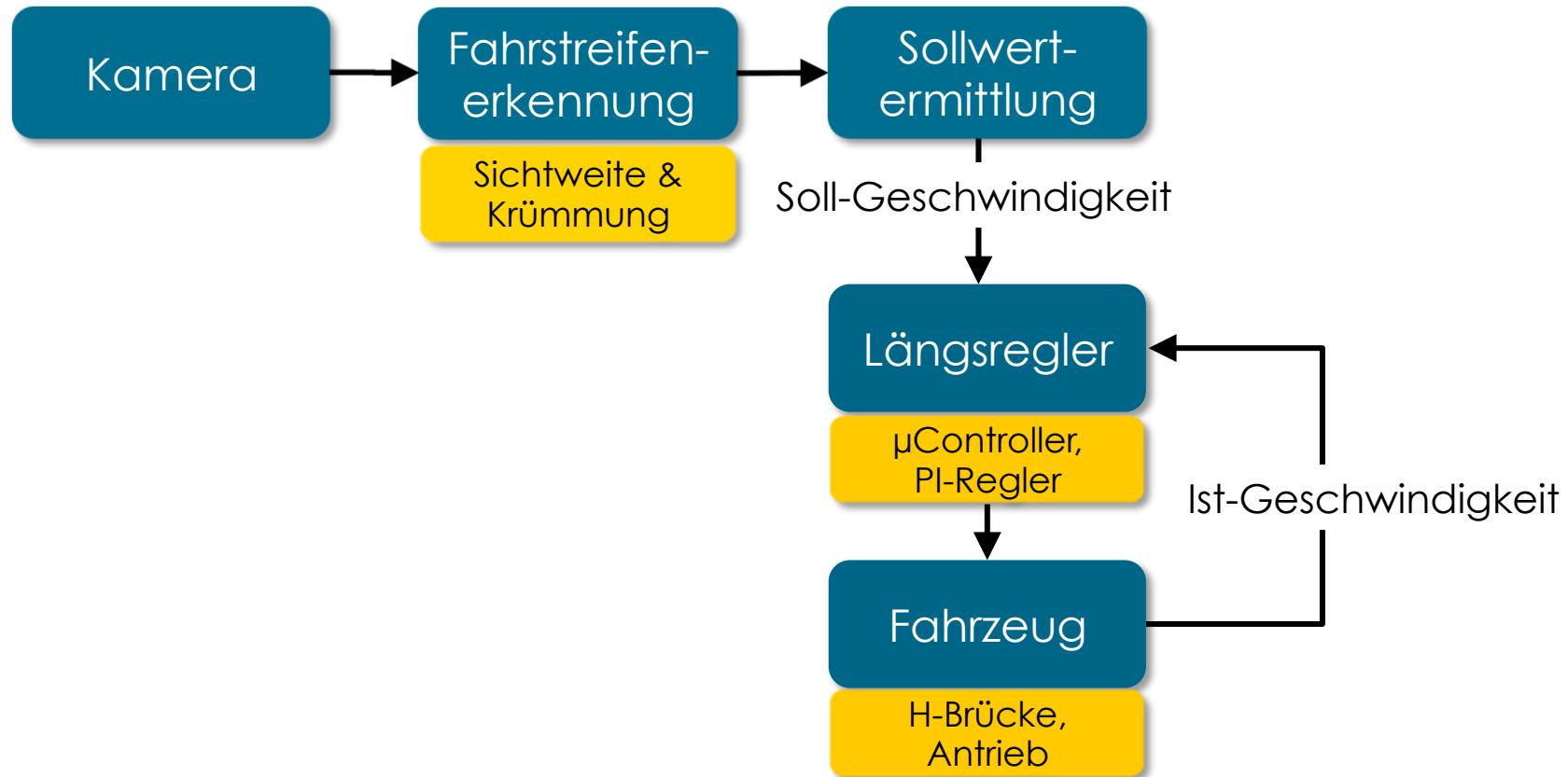
Straßenerkennung



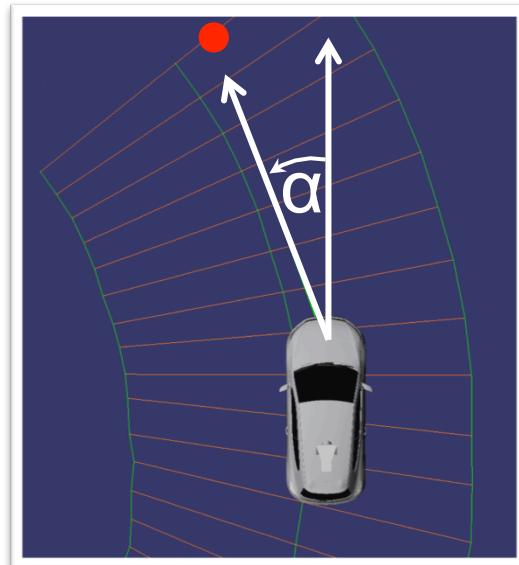
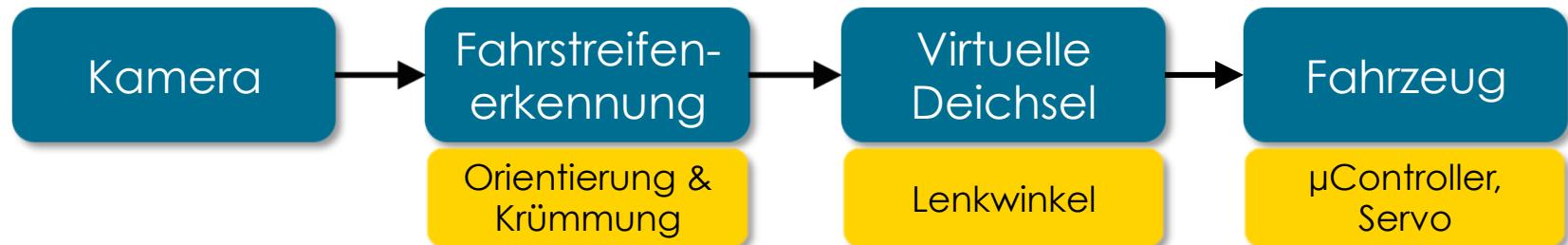
Straßenerkennung



Längsregelung



Querregelung



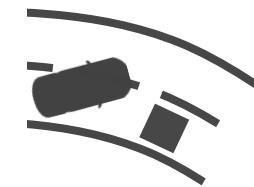
Gliederung

Konzeption

Freie Fahrt

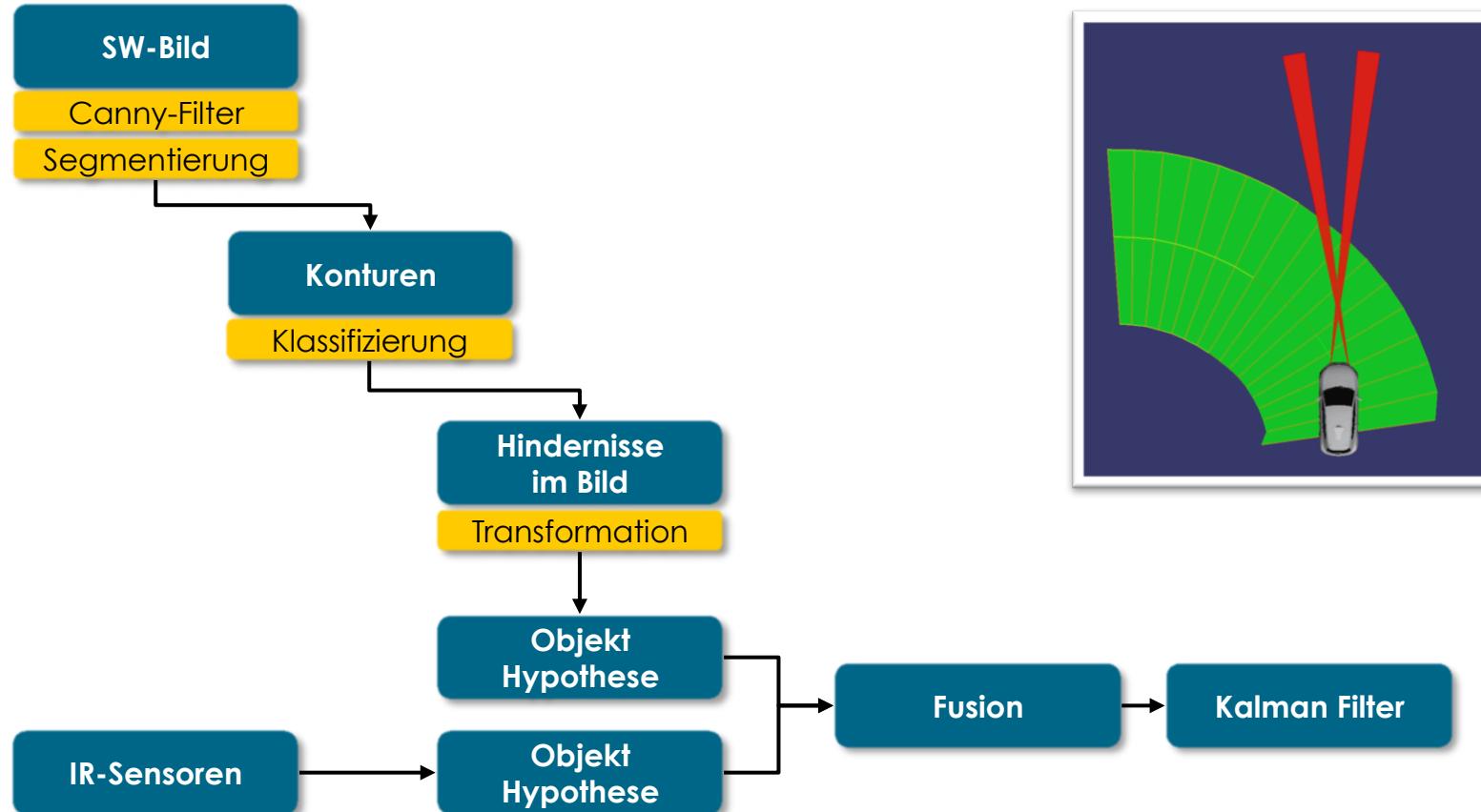
Hindernisfahrt

Parken



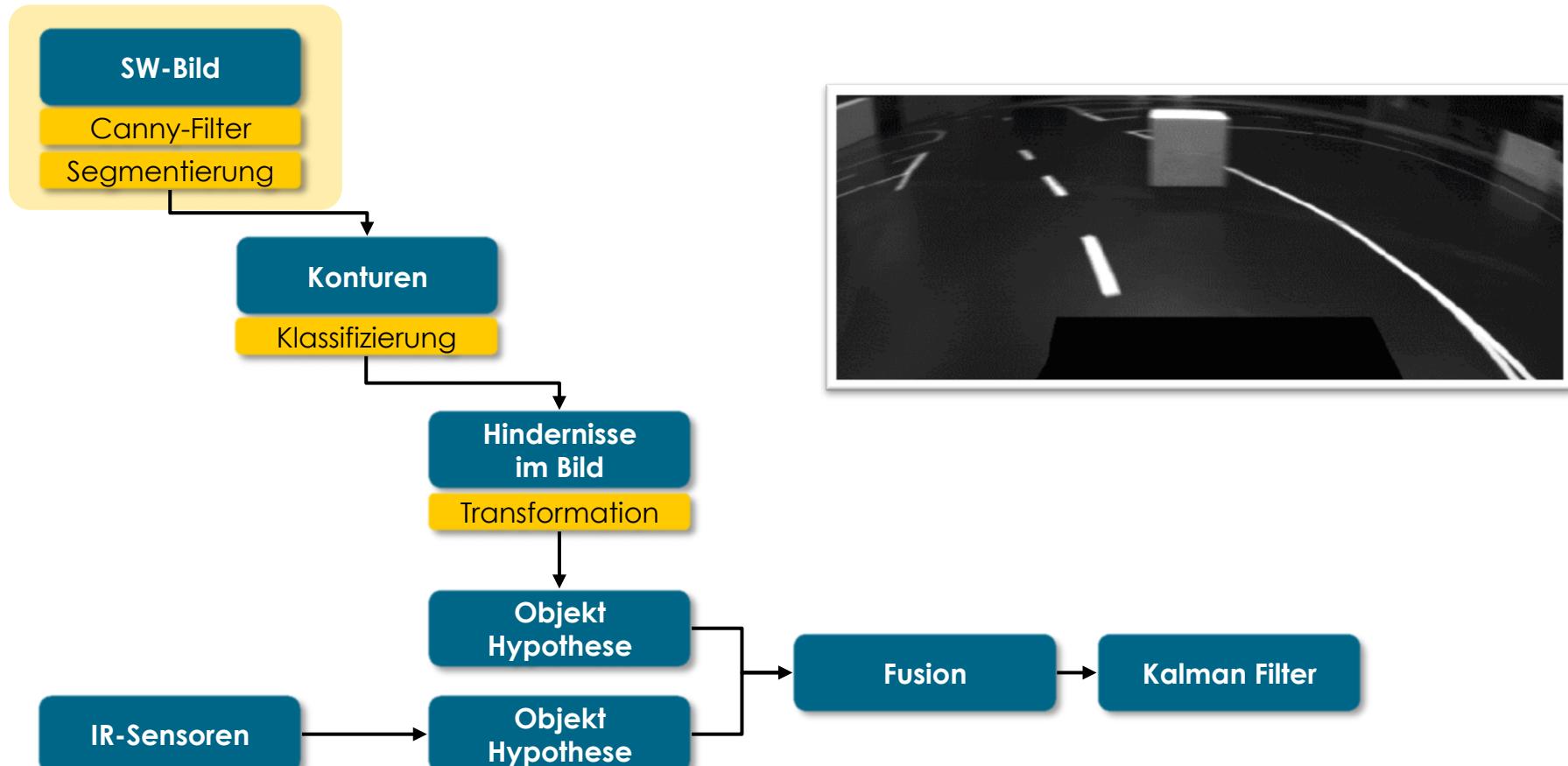


Objekterkennung



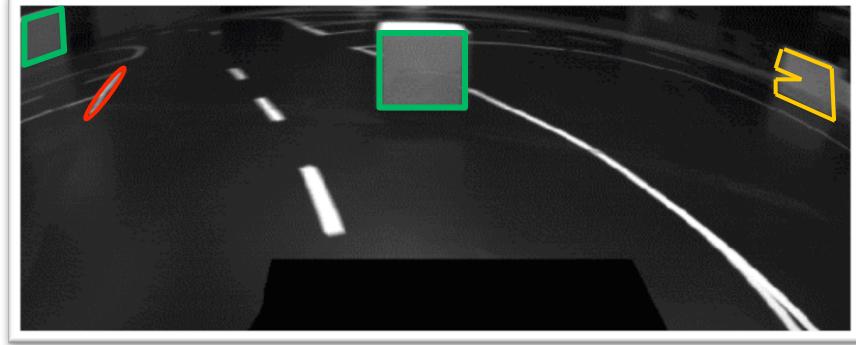
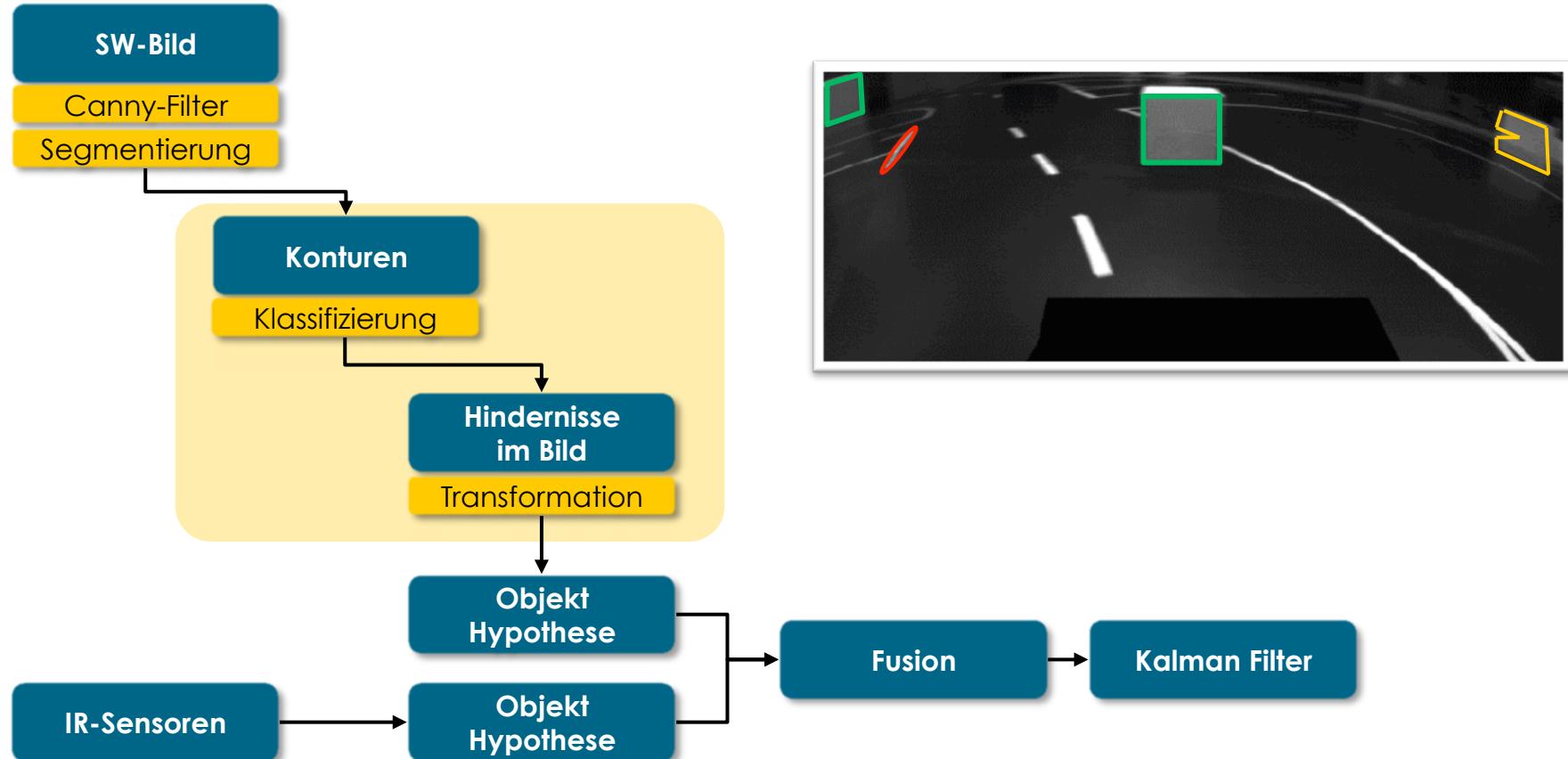


Objekterkennung



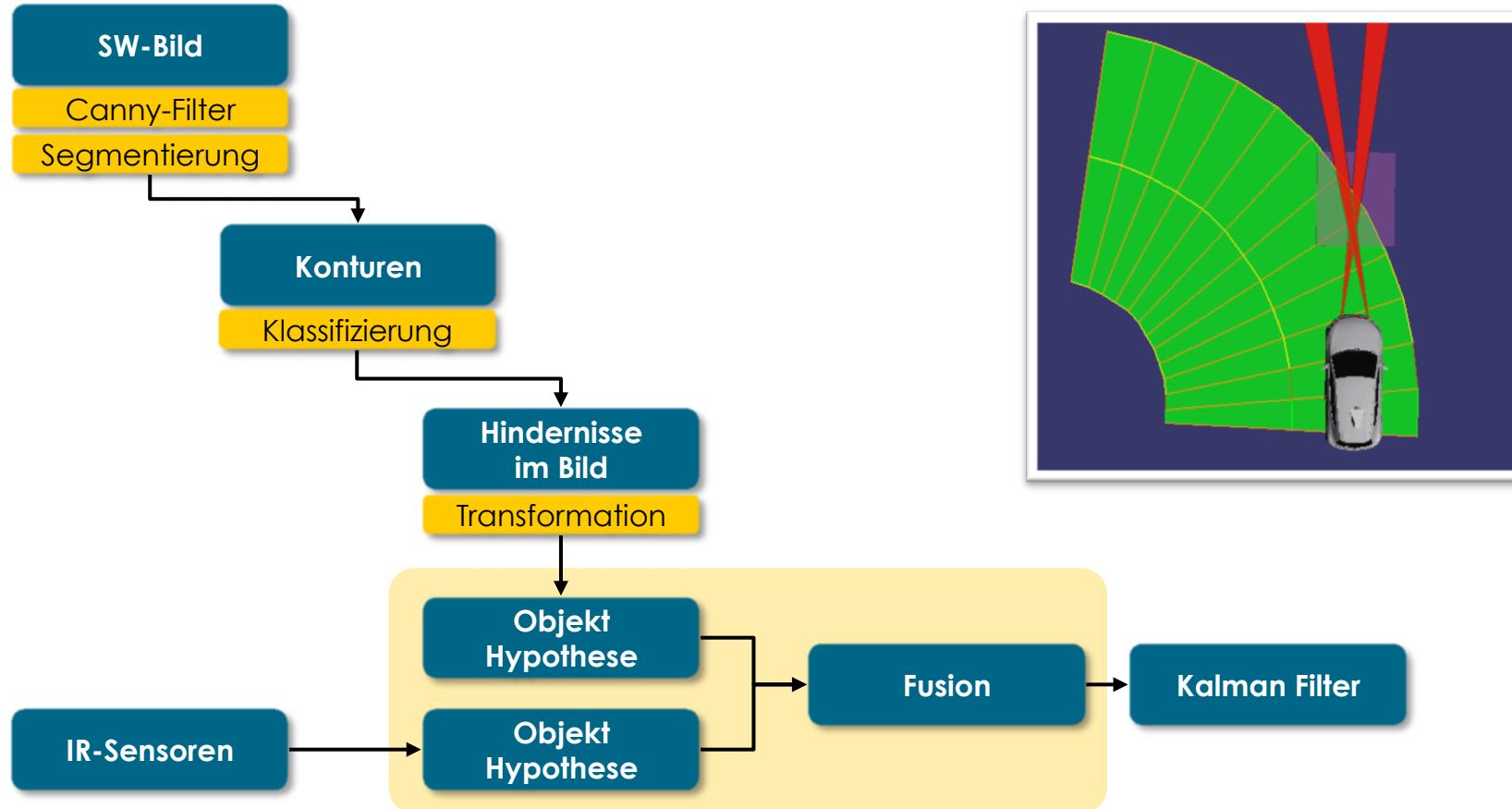


Objekterkennung



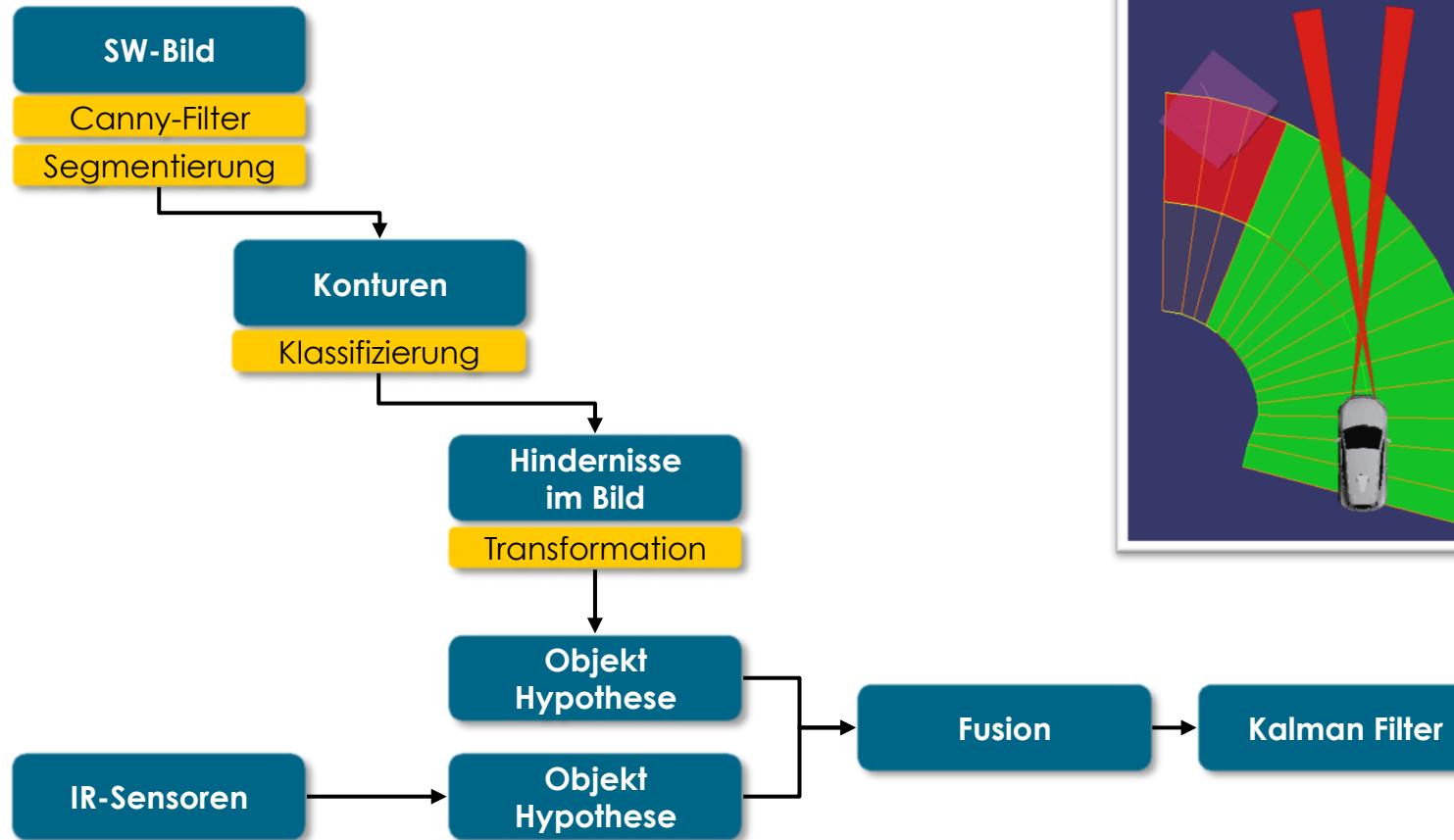


Objekterkennung





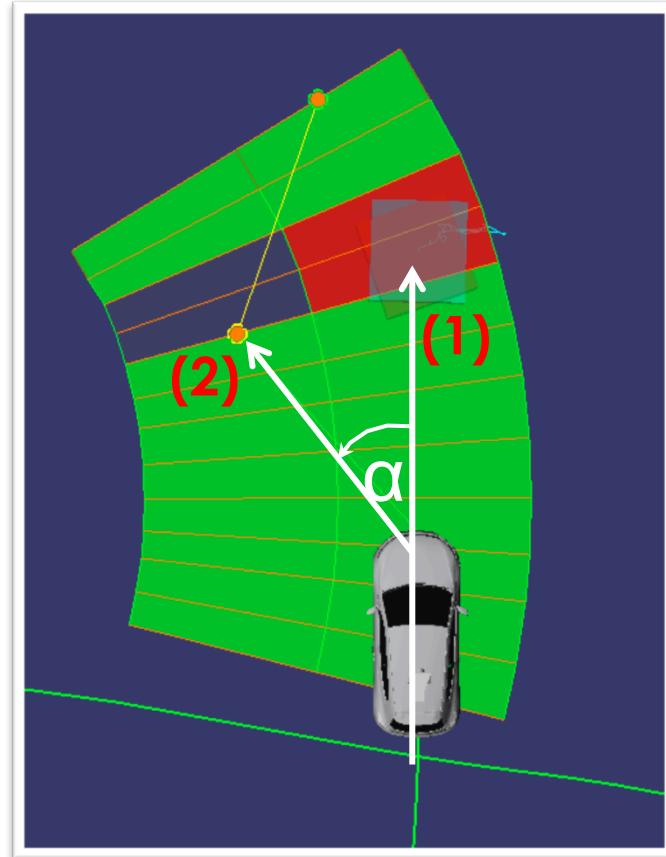
Objekterkennung





Ausweichstrategie

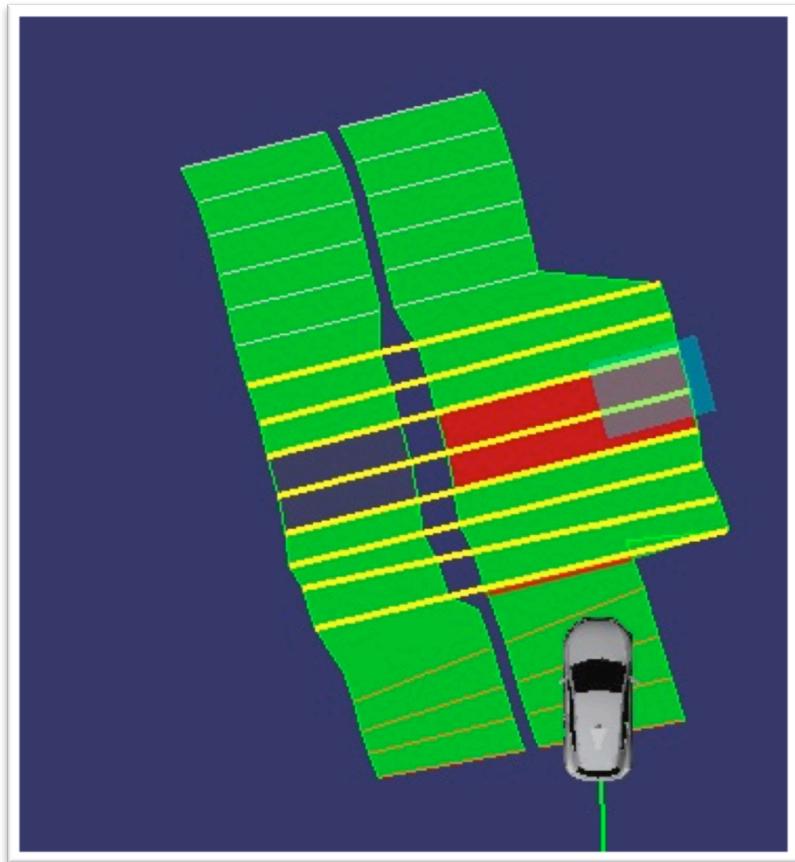
- Abfrage der Attribute der Straßenelemente: frei / belegt?
- Berechnung des Zielpunktes
- Ermittlung des Lenkwinkels α nach dem Prinzip der virtuellen Deichsel





Kreuzungssituation

- Stopplinienerkennung mittels RANSAC*
- Virtuelle Vergrößerung der Fahrstreifenbreite
- Gewährung der Vorfahrt mittels Zustandsautomat



* Random Sample and Consensus

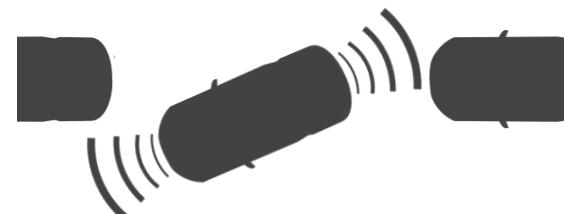
Gliederung

Konzeption

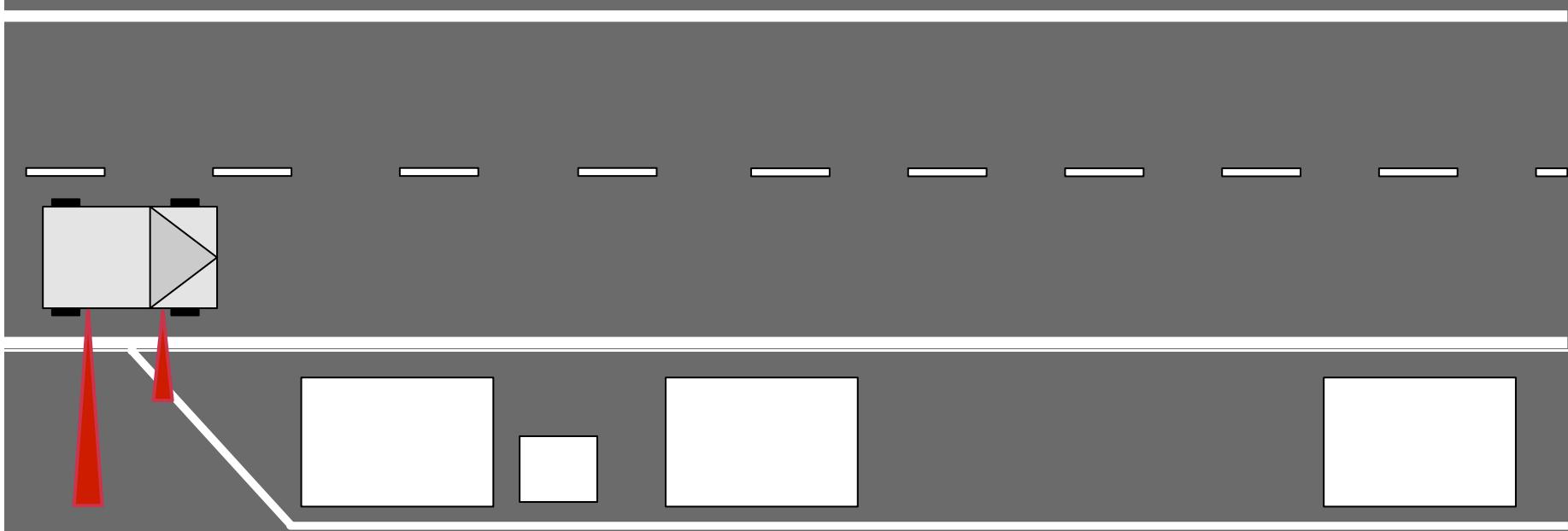
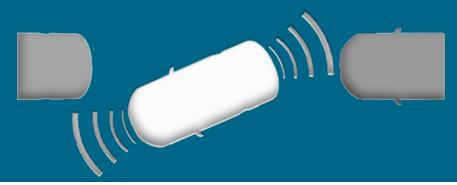
Freie Fahrt

Hindernisfahrt

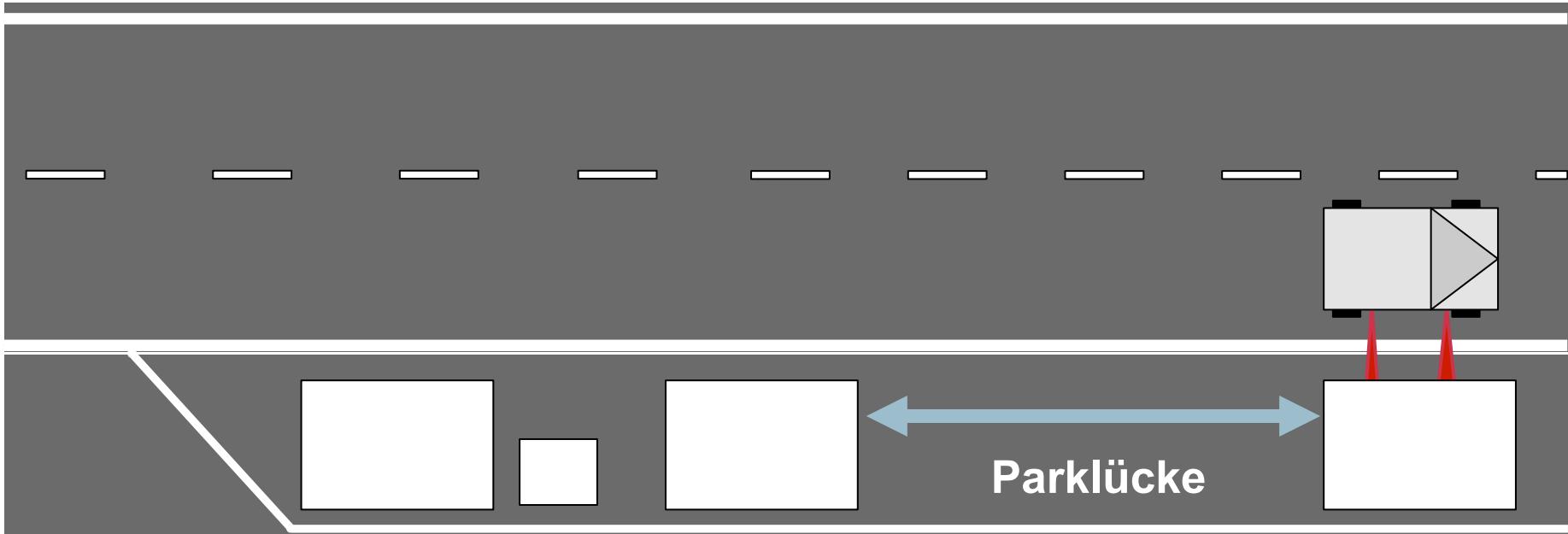
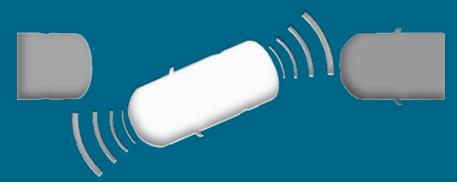
Parken



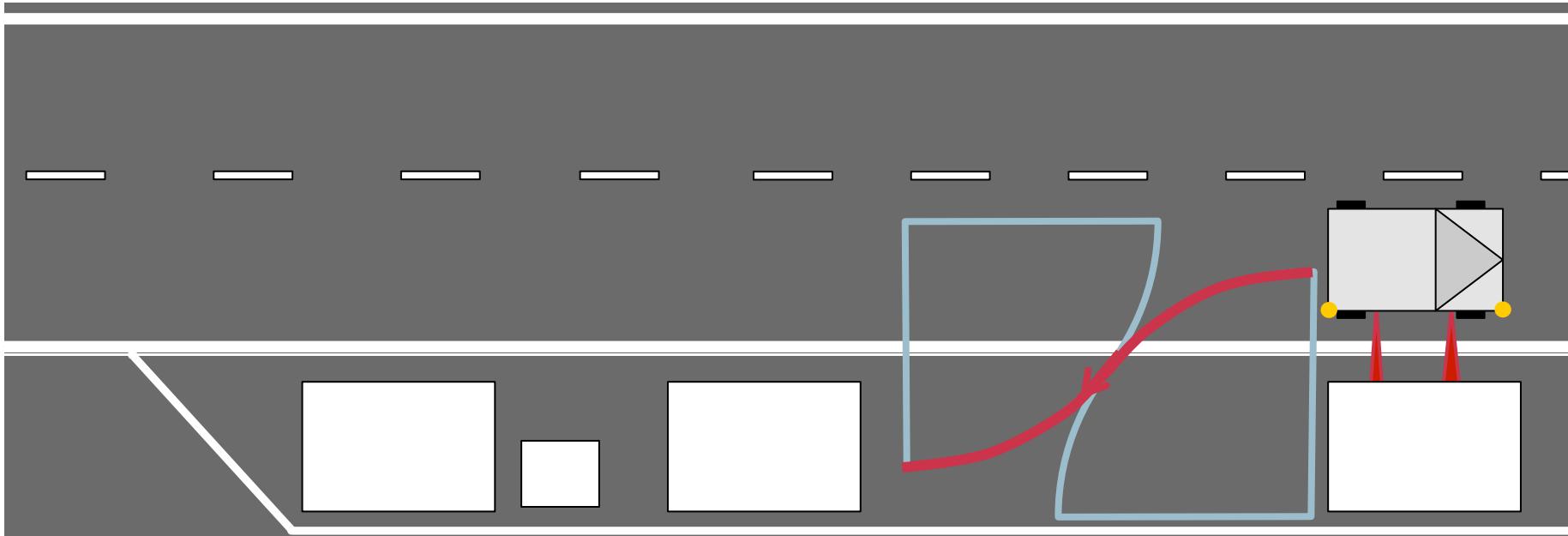
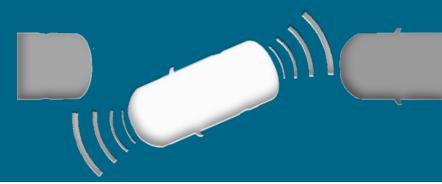
Einparkkonzept



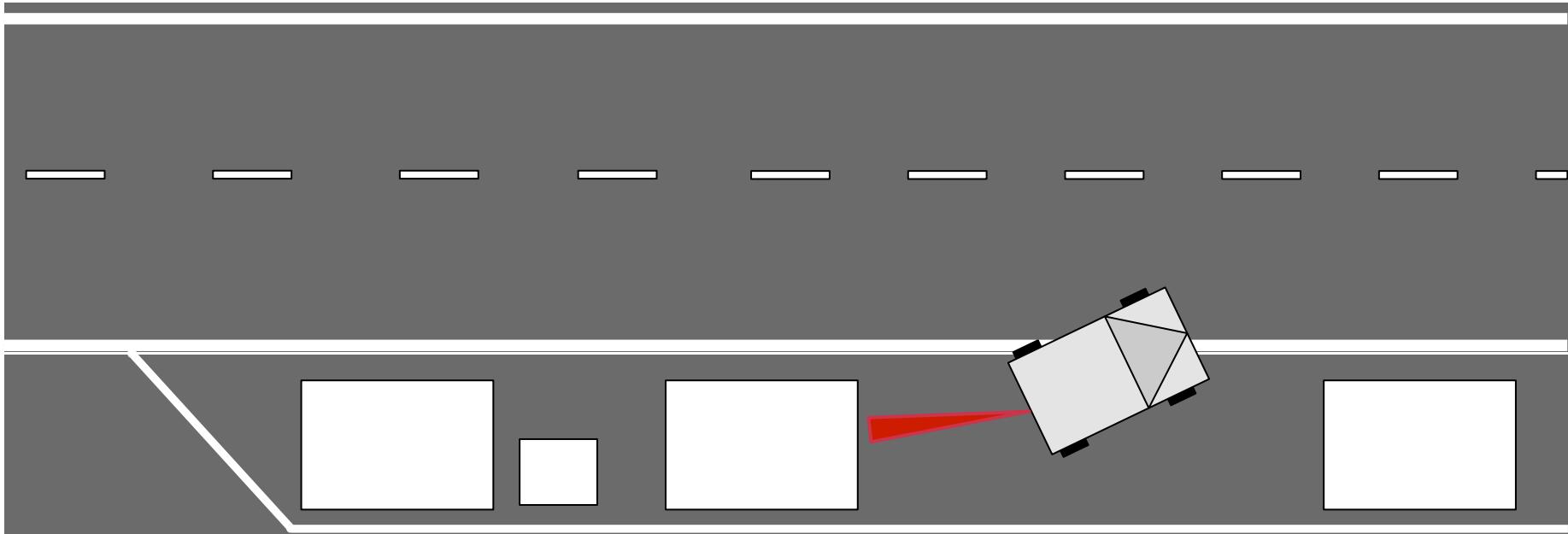
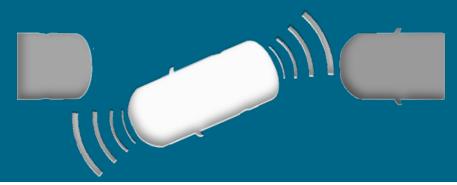
Einparkkonzept



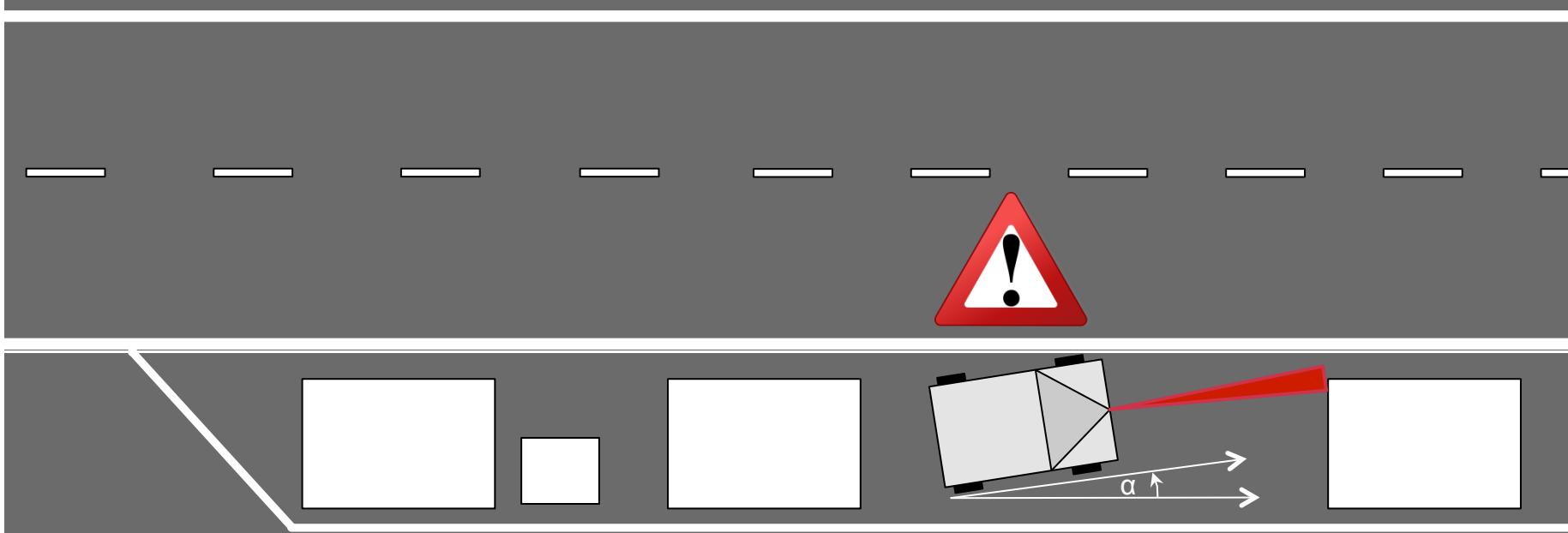
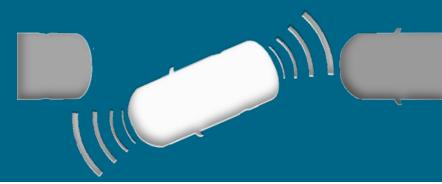
Einparkkonzept



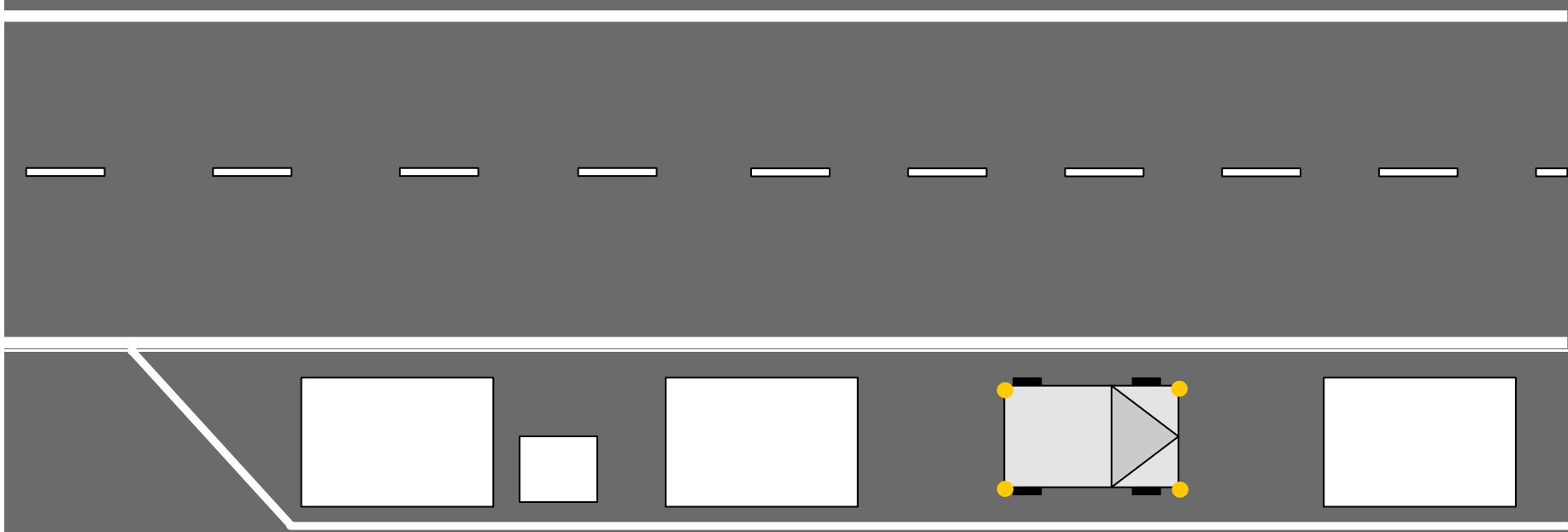
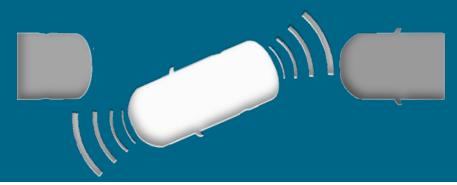
Einparkkonzept



Einparkkonzept



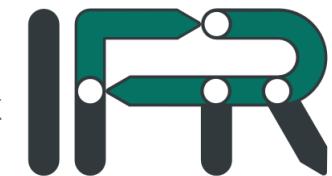
Einparkkonzept



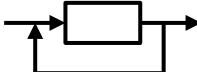
Wir danken für die Unterstützung:



Institut für
Regelungstechnik

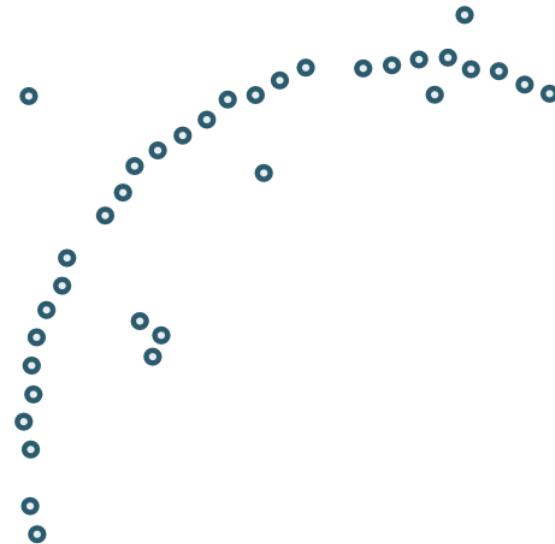


Danke für die Aufmerksamkeit

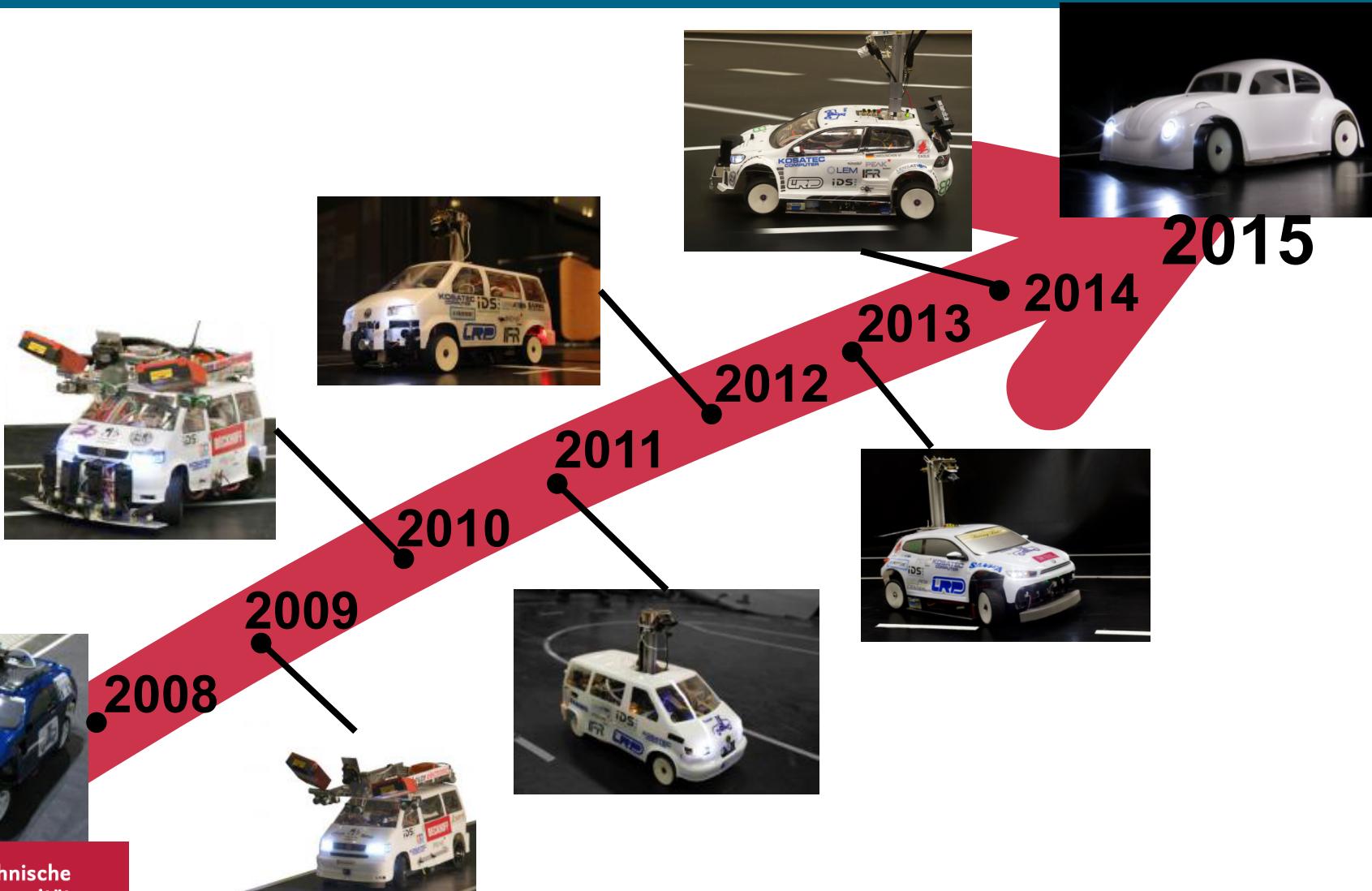
Konzept	Management	Hardware	Kosten	Softwarearchitektur
				
	Bahnerkennung			Lokalisierung
	Sensoren			
				

RANSAC – Random Sample and Consensus

- Robuster iterativer Algorithmus zum Auffinden analytischer Geometrien
- Hypothesengenerierung:
 - Zufällige Wahl minimaler Punktzahl für Modell
- Hypothesenbewertung:
 - Zähle Punkte aus Punktmenge, die zu Hypothese passen („**Inlier**“)
- Übernahme der Hypothese mit größtem **Consensus-Set**

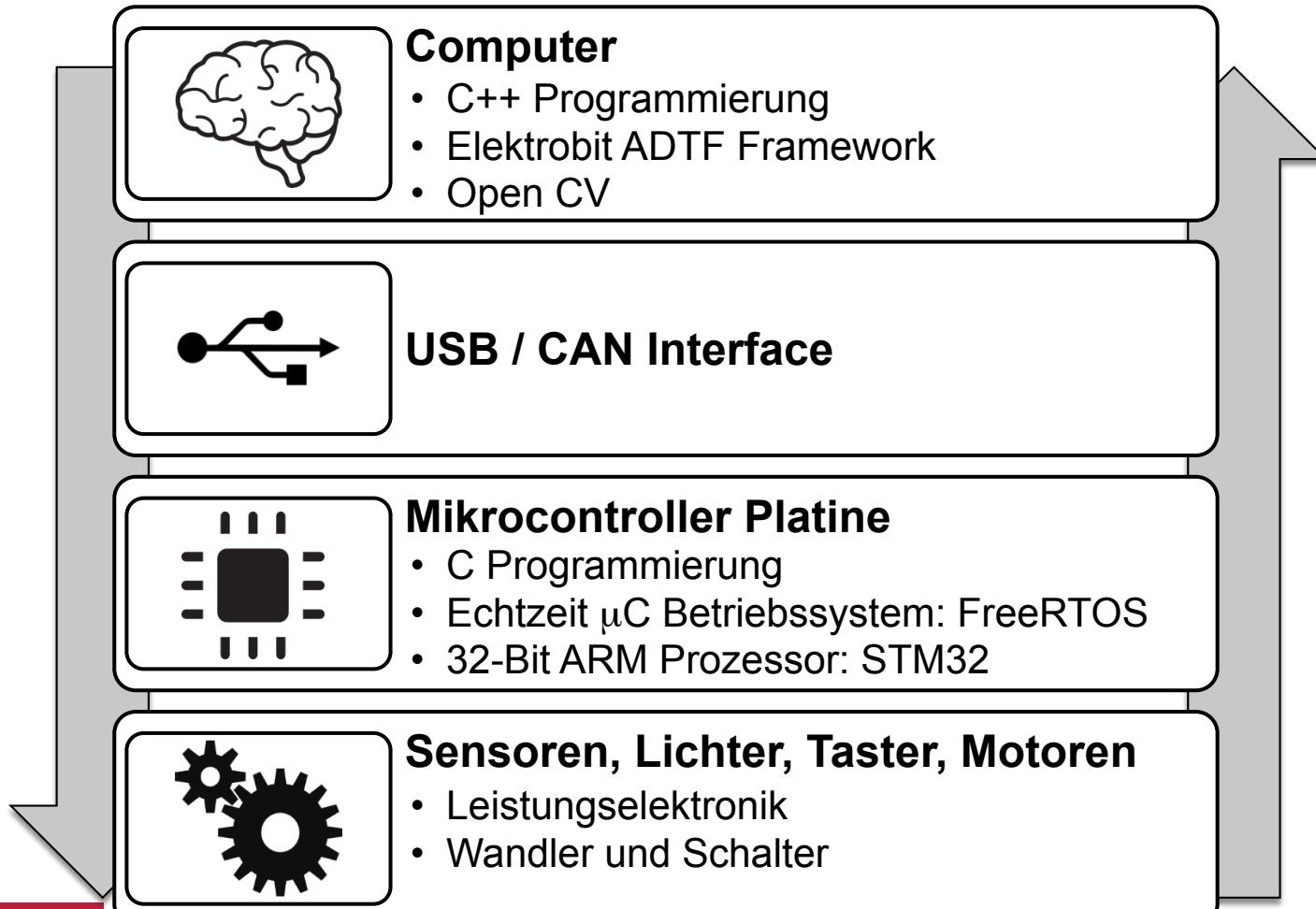


Evolution: Carolinchen I bis VII





Systemarchitektur





Softwarearchitektur

Wahrnehmung

Entscheidung

Aktion

Straßen Erkennung
& Vorhersage

Entscheidungs-
findung

Lokalisierung

Situations-
analyse

Objekterkennung

Bahnplanung

Sensoren

Aktoren



Qualitätssicherung

Codezeilen (C, C++): ca. 65.000

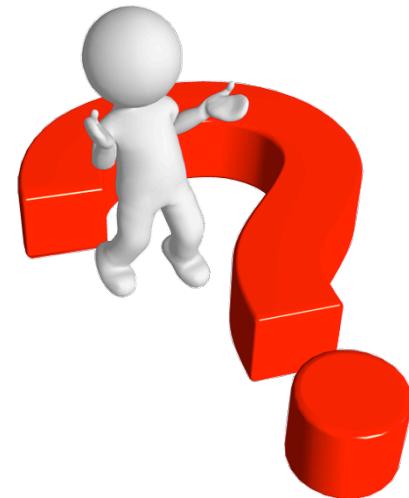
Testkilometer: 10 km

Testkilometer simuliert: 65 km

Vorgehen:

1. Rapid-Prototyping mit Python
2. Unitests] Simulationen in ADTF und
3. Modultests] Praxistest auf Testparcours

Vielen Dank für Ihre Aufmerksamkeit!



Gibt es Fragen?