

Grundlagen der Robotik

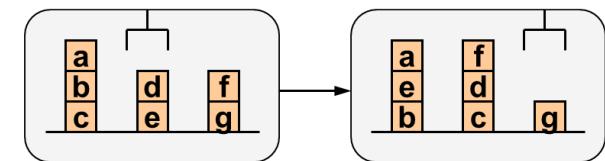
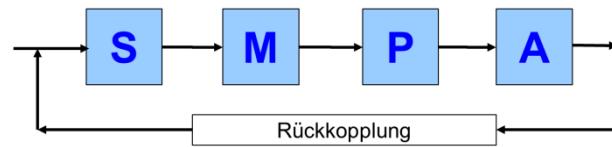
14. Verhaltenssteuerung cont., Pfadplanung

Prof Sven Behnke

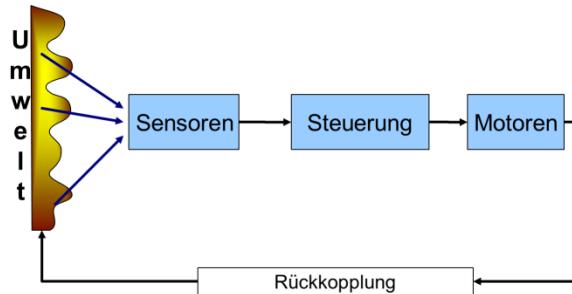


Letzte Vorlesung

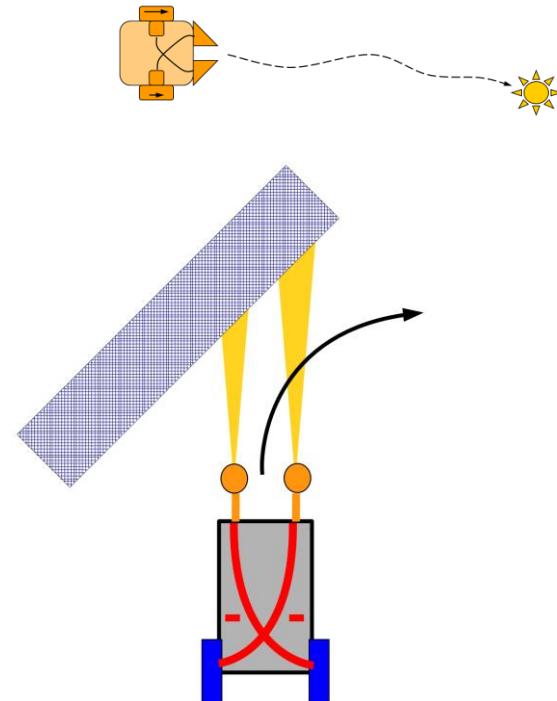
- Architekturen zur Verhaltenssteuerung
 - Deliberativ



- Reaktiv

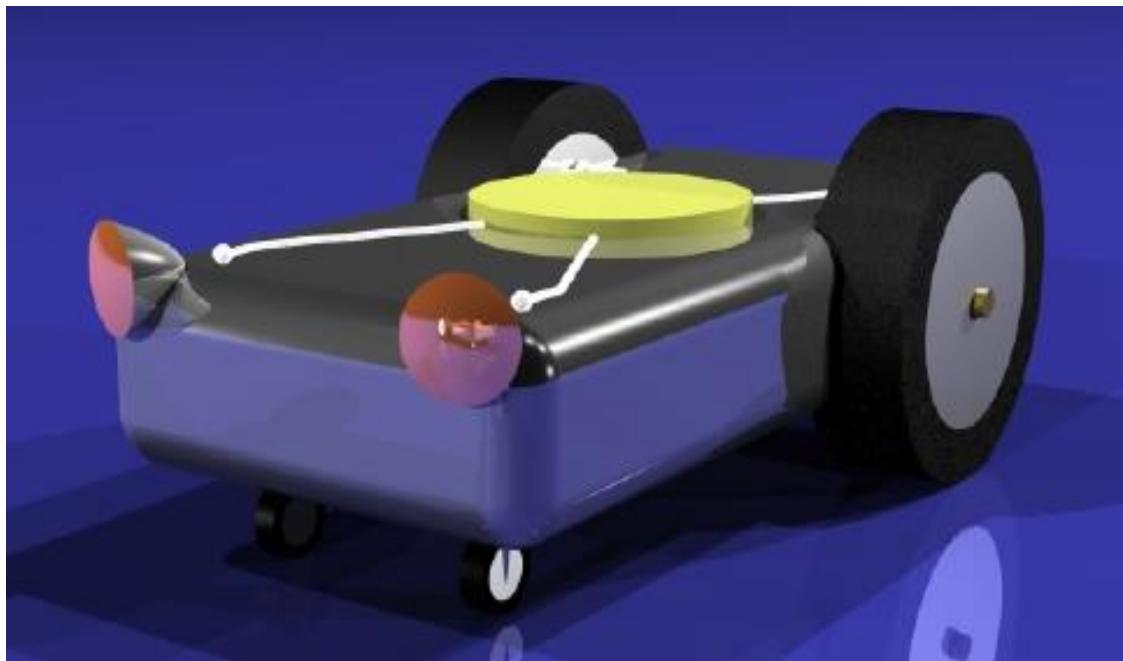


- Braitenberg-Vehikel



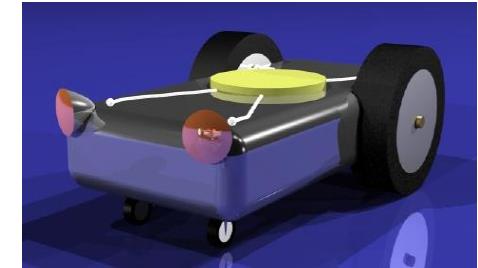
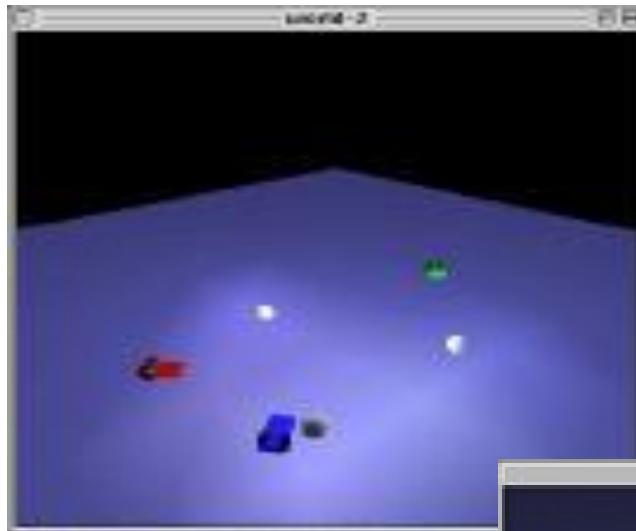
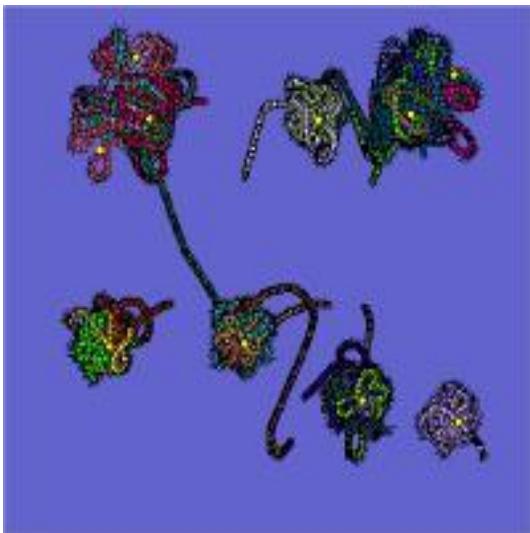
Braitenberg-Vehikel, Implementation

- Seit der Veröffentlichung des Buches durch Valentino Braitenberg 1984 hat es eine Vielzahl von Implementationen zu diesem Ansatz gegeben; in Simulationen und an echten, realen Robotersystemen.



From: <http://people.cs.uchicago.edu/~wiseman/vehicles/>

Braitenberg-Vehikel, Implementation



From: <http://people.cs.uchicago.edu/~wiseman/vehicles/>

Erinnerung: Architekturen zur Verhaltenssteuerung

- Deliberativ

- Sense-plan-act
- Langsam, komplex

- Reaktiv

- Sense-act
- Schnell, primitiv

- Hybrid

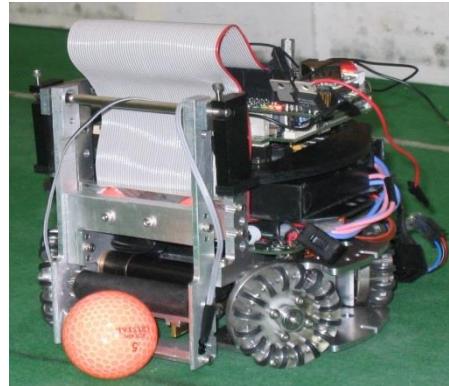
- Deliberative und reaktive Schicht
- Eventuell Zwischenschicht

- Verhaltensbasiert

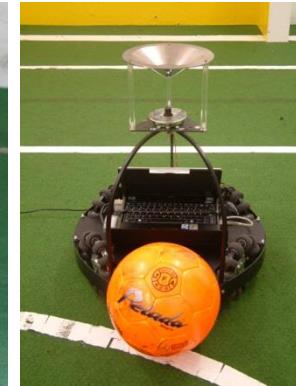
- Mehrere einfache Verhalten
- Aktivierung situationsabhängig

Omnidirektionale Fahrwerke

- Geschwindigkeit (x, y, θ) verteilt auf drei oder mehr Räder
- Gute Manövrierbarkeit
- Vorteile in
 - Beengten Situationen
 - Dynamischen Umgebungen



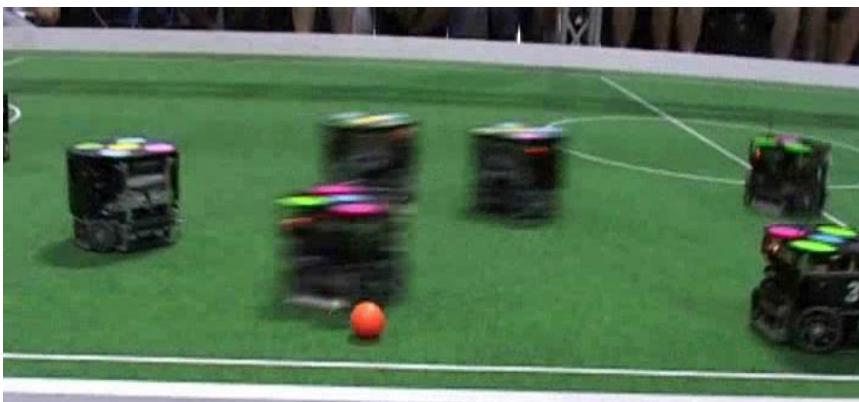
Cornell Big Red



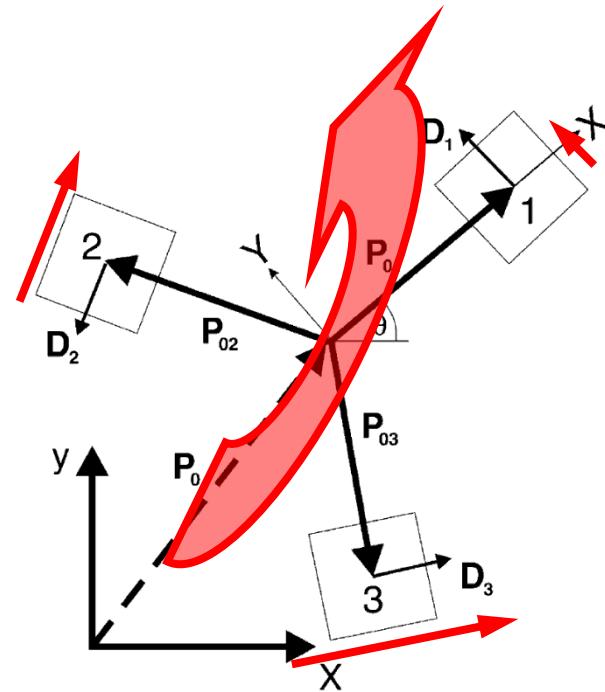
FU-Fighters



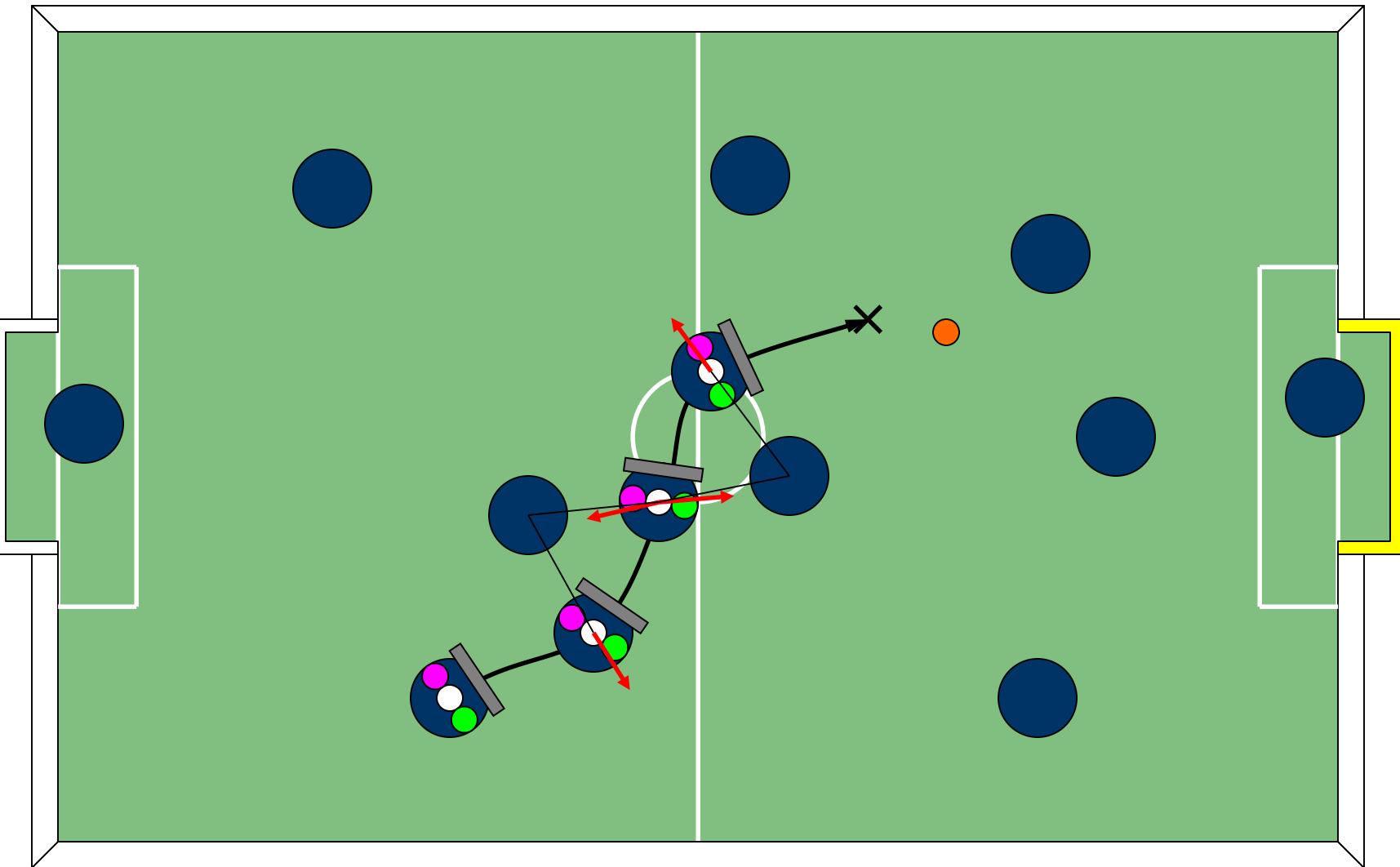
Armar III



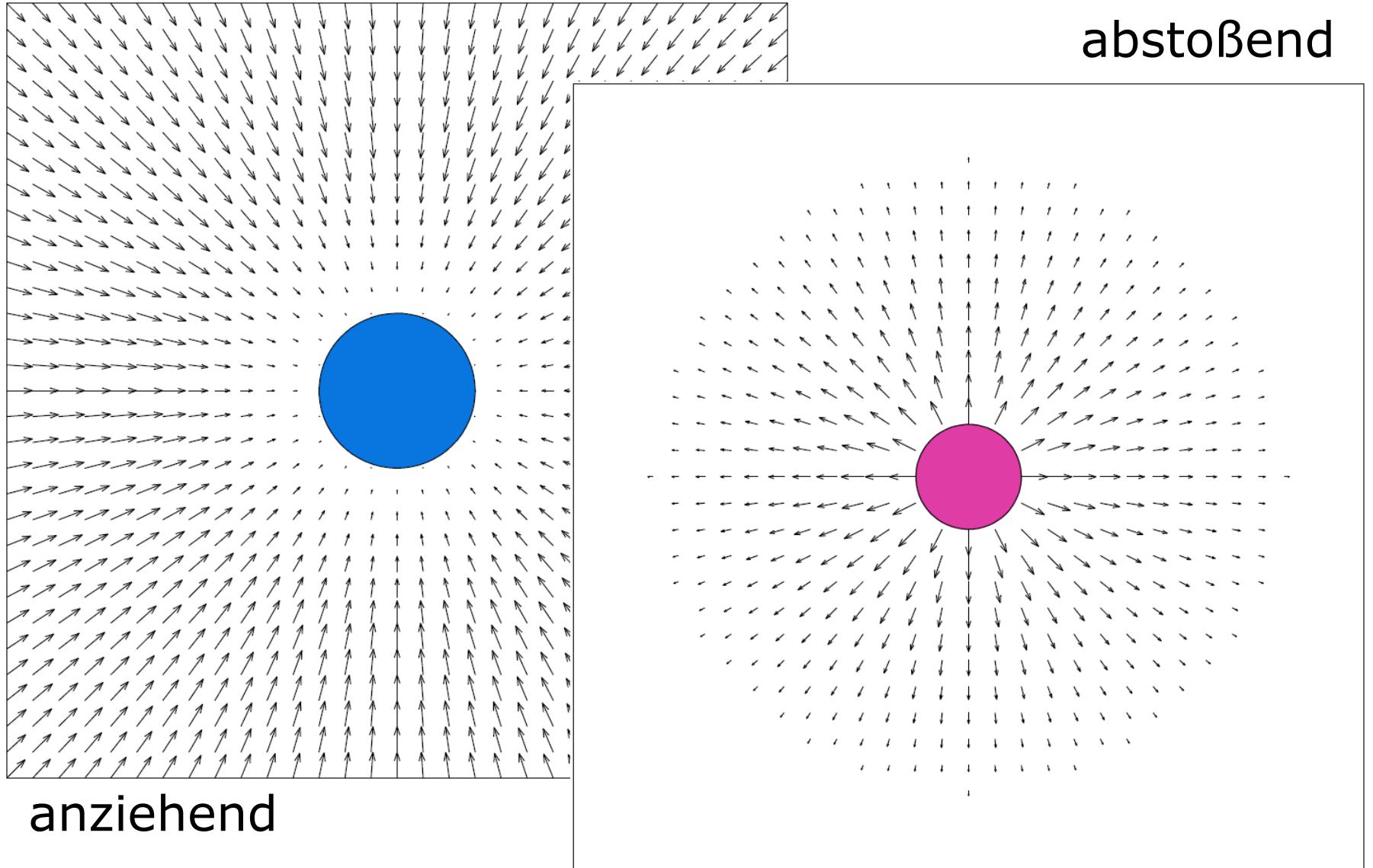
RoboCup
2005



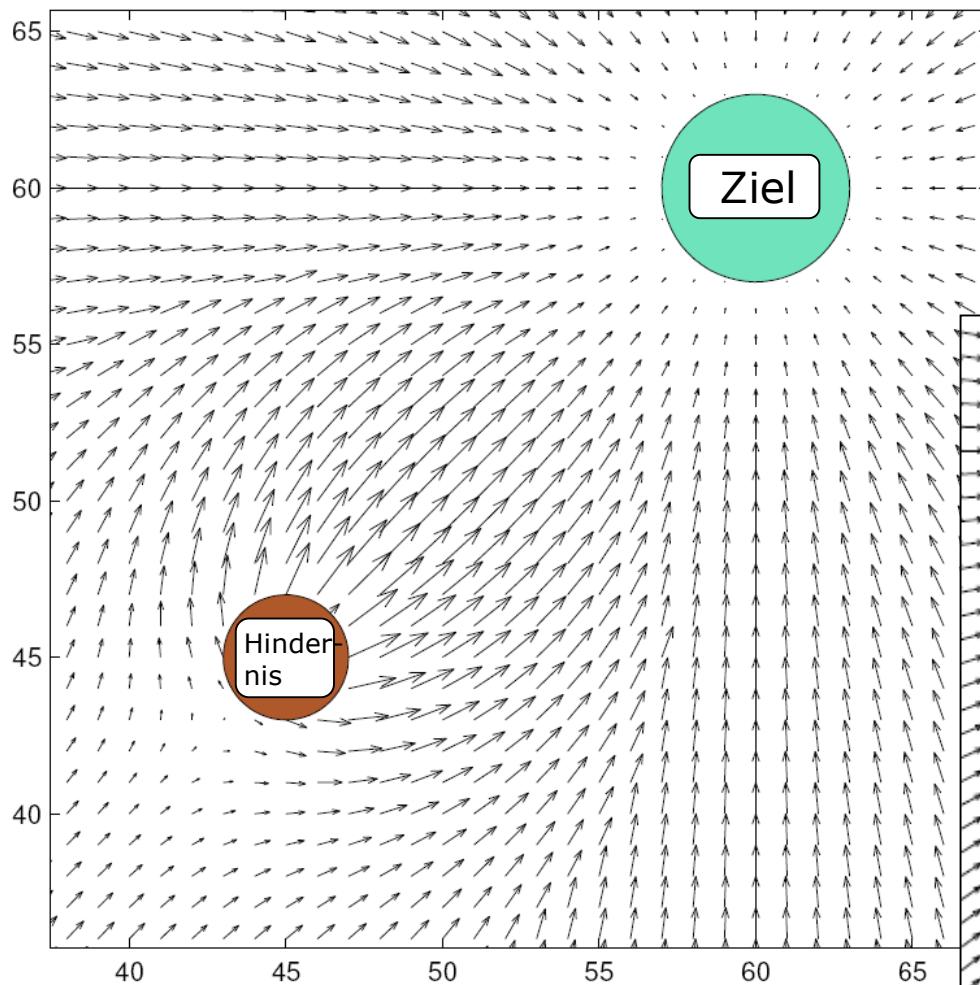
Reaktive Hindernisvermeidung



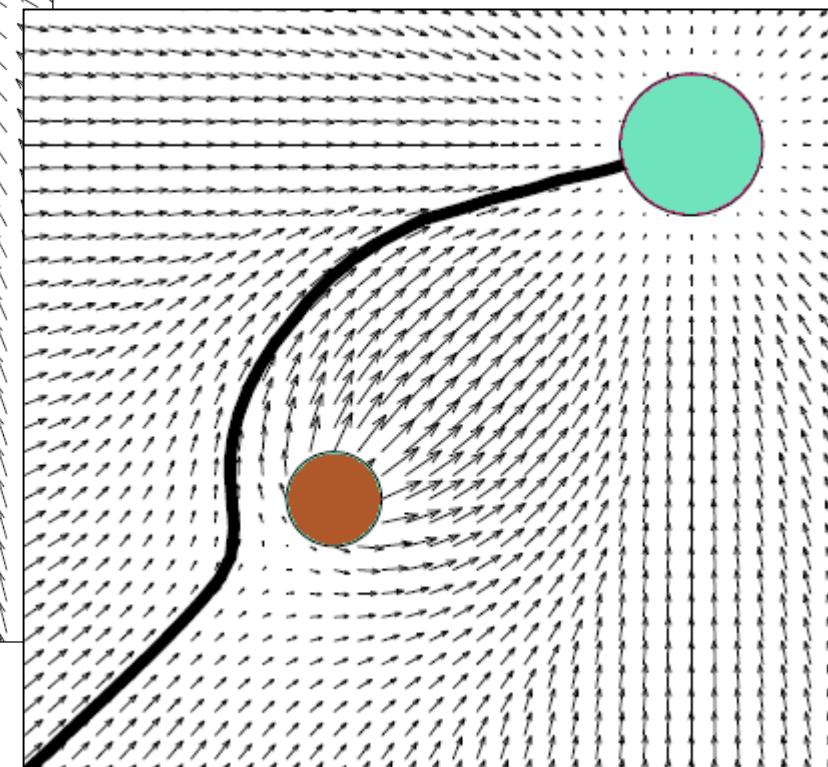
Potentialfelder



Kombinierte Potentialfelder

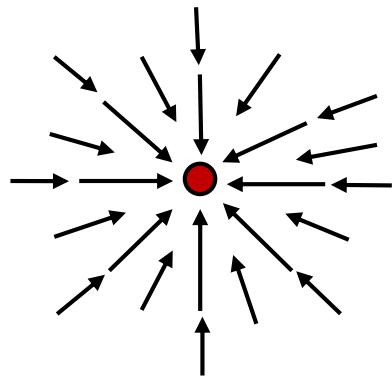


- Additive Kombination
- Bewegung in Pfeilrichtung
- Zum Ziel unter Vermeidung von Hindernissen

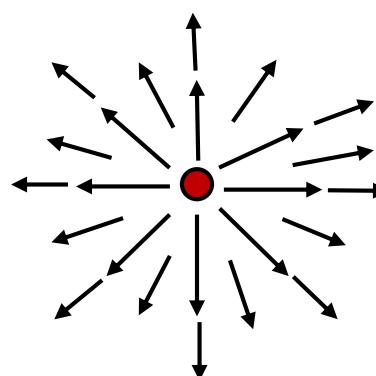


Potentialfeld-Typen

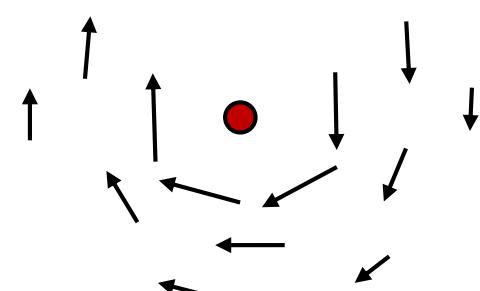
■ Anziehend



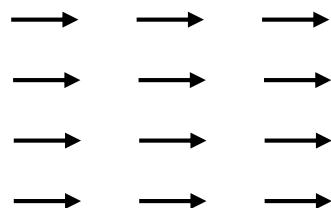
■ Abstoßend



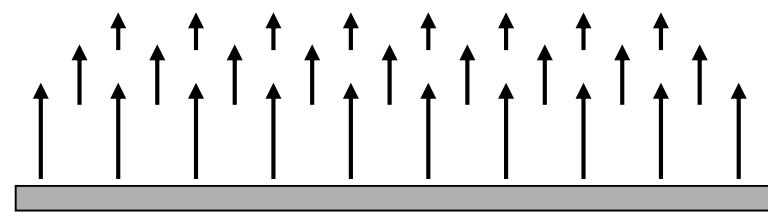
■ Tangential



■ Uniform

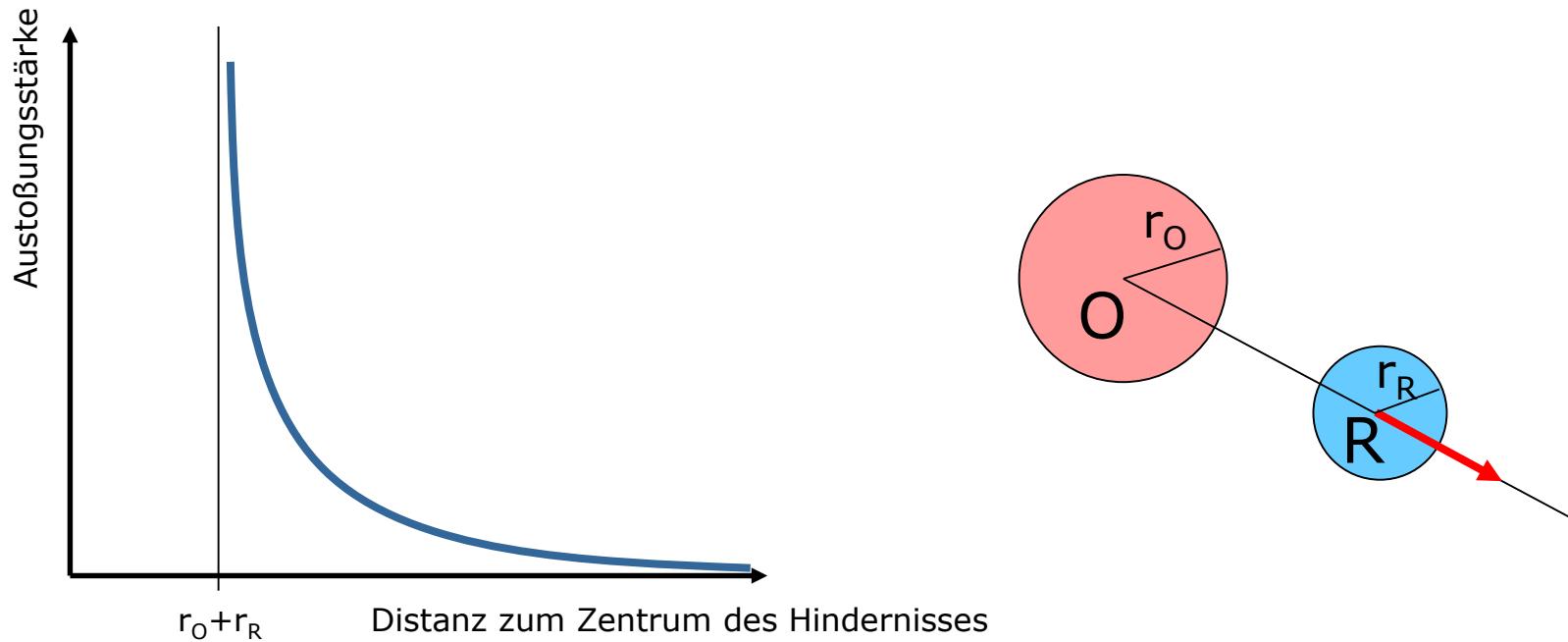


■ Normal zur Wand



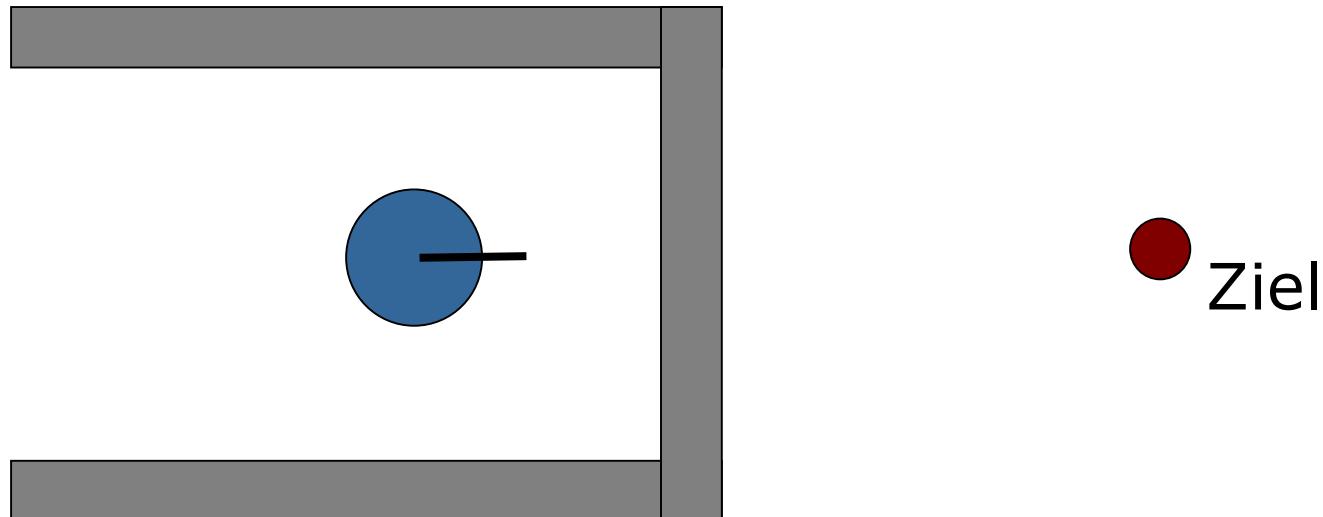
Beispiel: Abstoßendes Potential

- Richtung: Ausgehend vom Zentrum des Hindernisses
- Stärke: Zunehmend mit Nähe,
z.B. $1/(\text{dist}(O, R) - (r_O + r_R))$



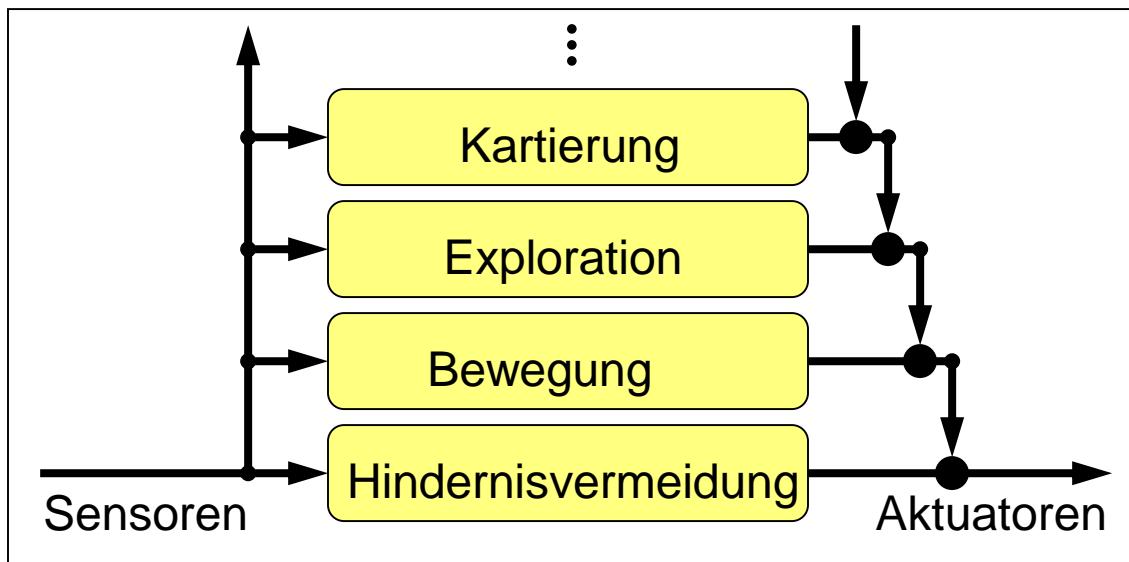
Problem: Lokale Minima

- Anziehende und abstoßende Potentiale löschen sich aus
- Roboter findet nicht heraus
- Typische Situation:



Subsumptionsarchitektur

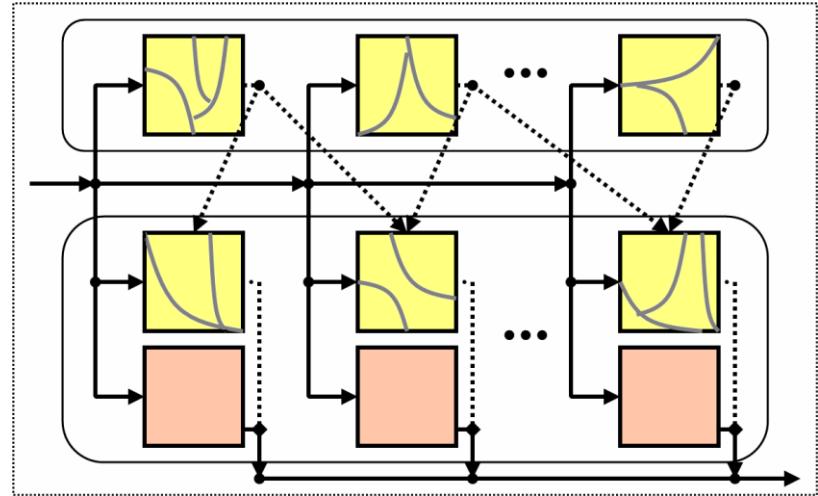
- Verhaltensbasiert, endliche Automaten, Nachrichten
- Kompetenzschichten
- Zunehmende Komplexität
- Subsumption: Überstimmen tieferer Schichten
- Inkrementelle Vorgehensweise



Rodney Brooks 1985

Dual Dynamics

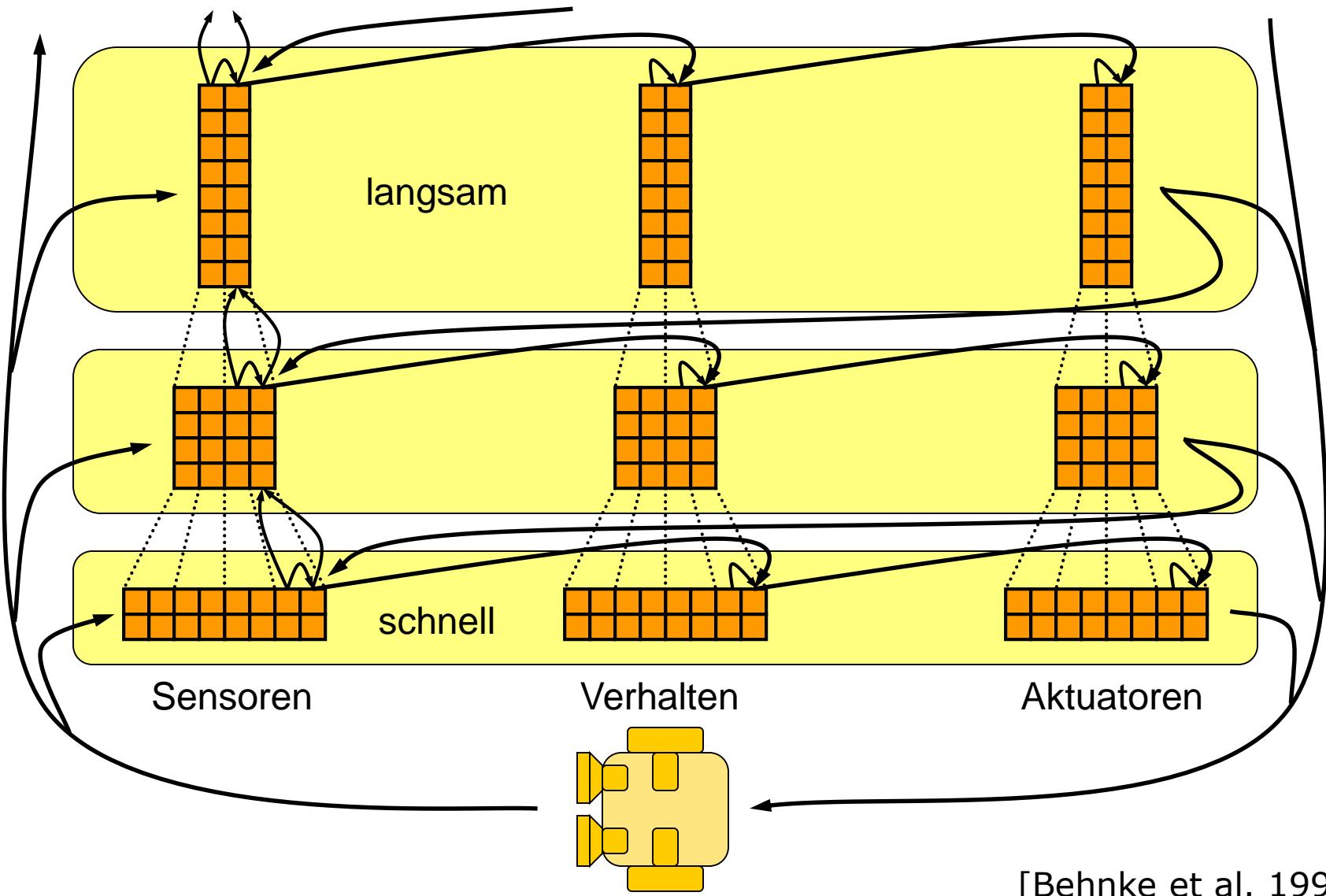
- Verhaltensbasiert
- Dynamische Systeme
- Zwei Verhaltensarten
 - elementar
 - komplex



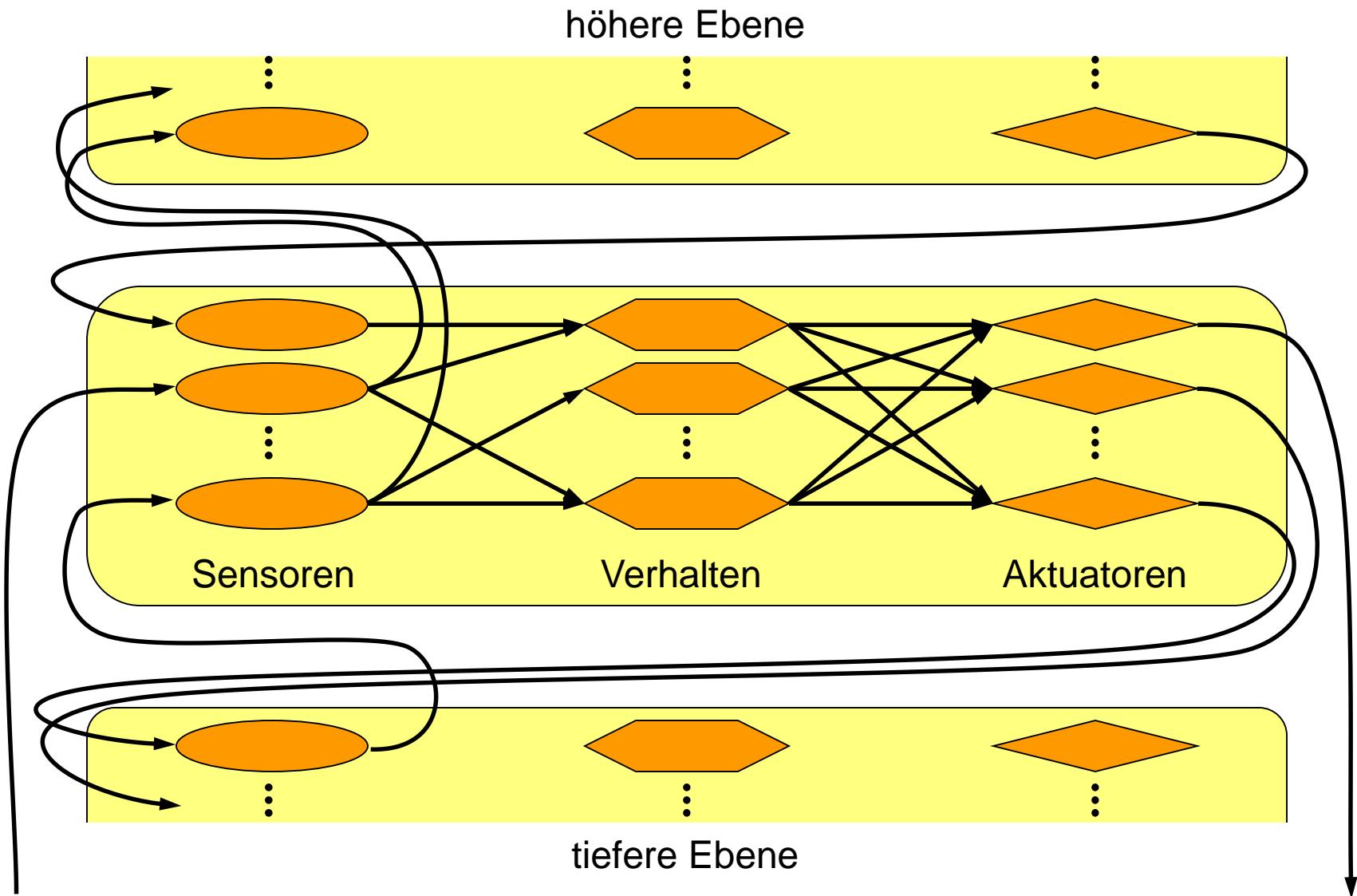
Herbert Jaeger 1995

- Elementare Verhalten haben zwei Module:
 - Aktivierungsdynamik: Soll ich jetzt etwas tun?
 - Zieldynamik: Was soll ich tun?
- Keine Zieldynamik bei komplexen Verhalten
- Bifurkationen, Modi, Multifunktionalität

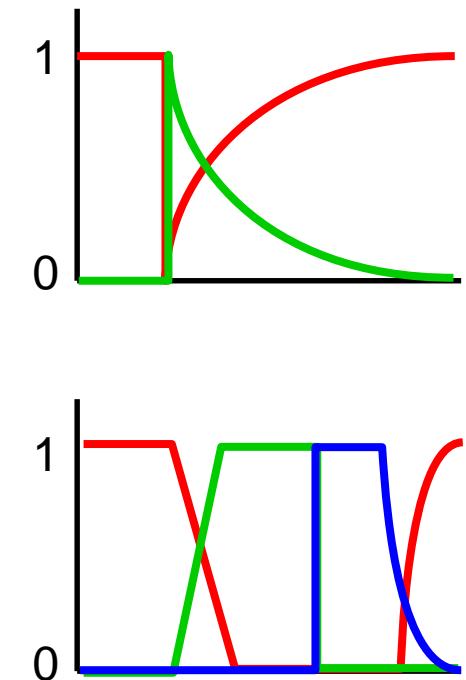
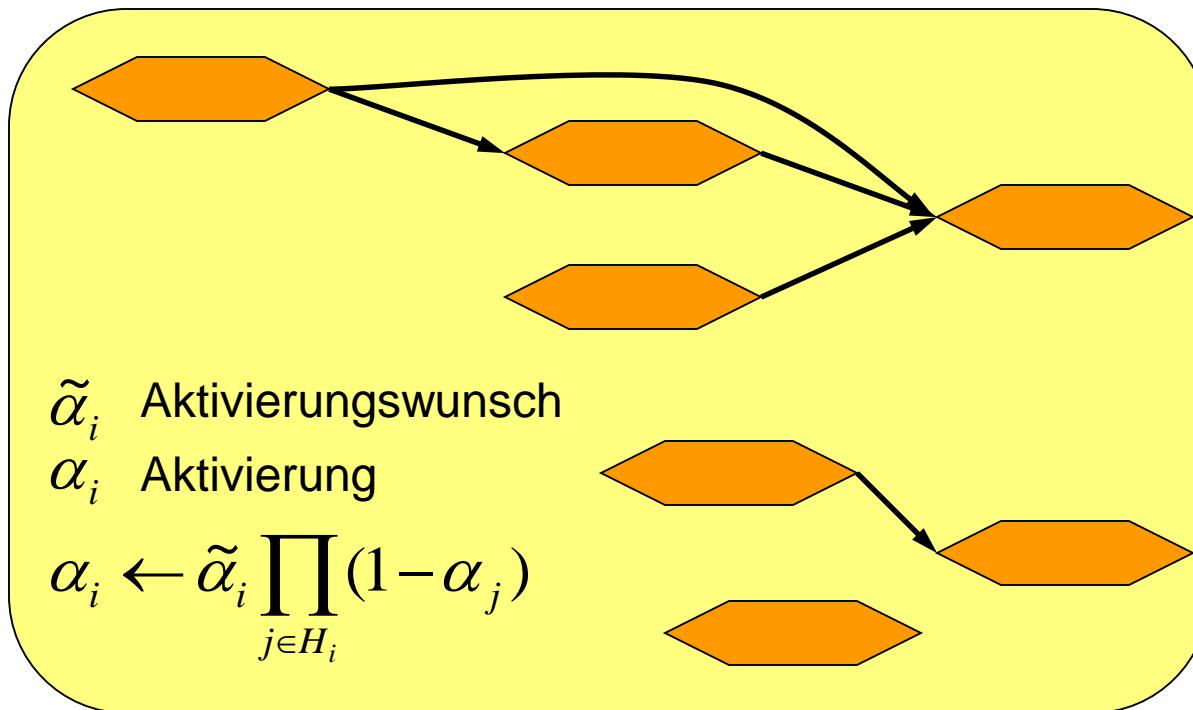
Hierarchie reaktiver Verhalten



Aufbau einer Ebene

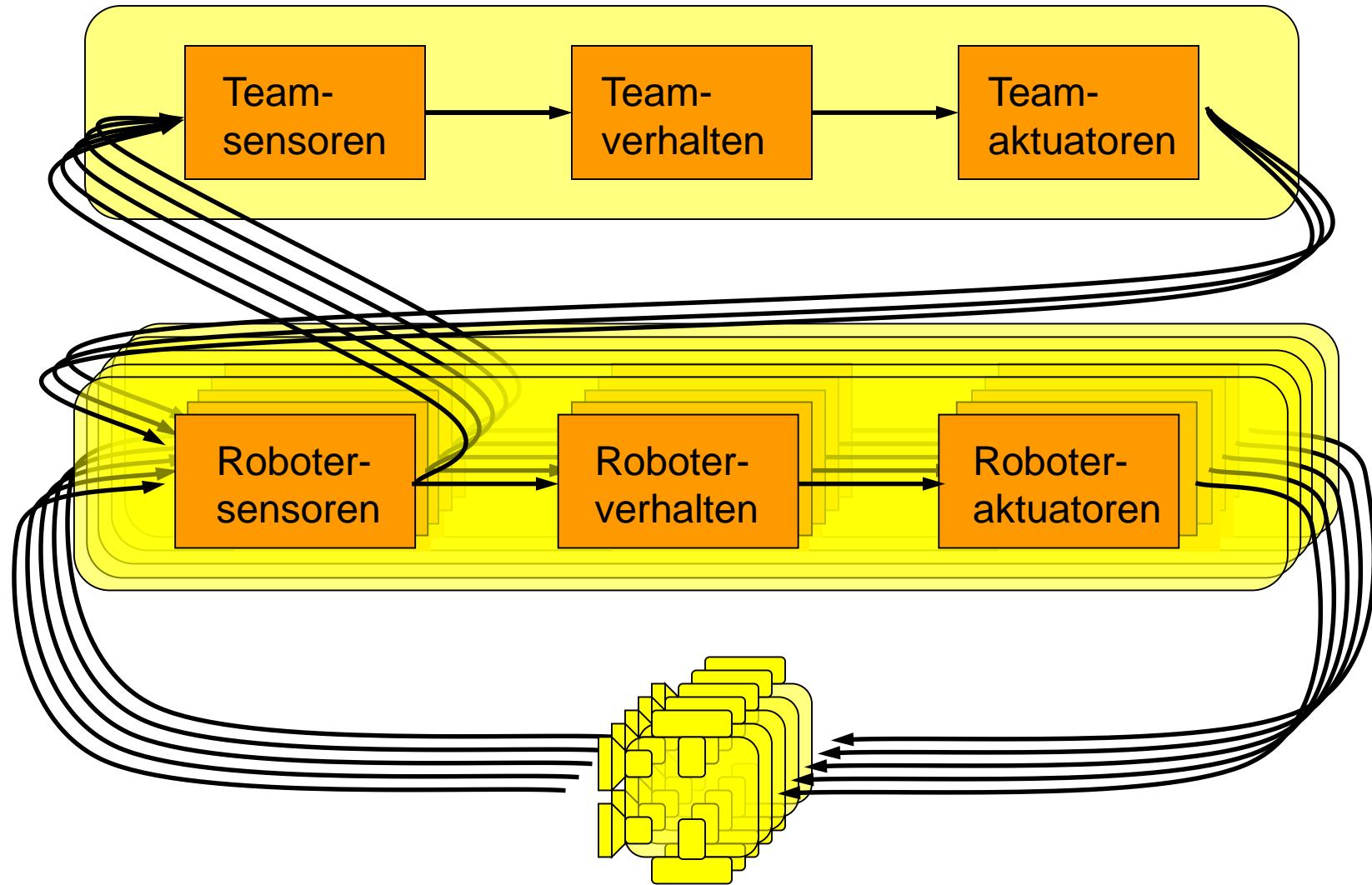


Hemmung von Verhalten

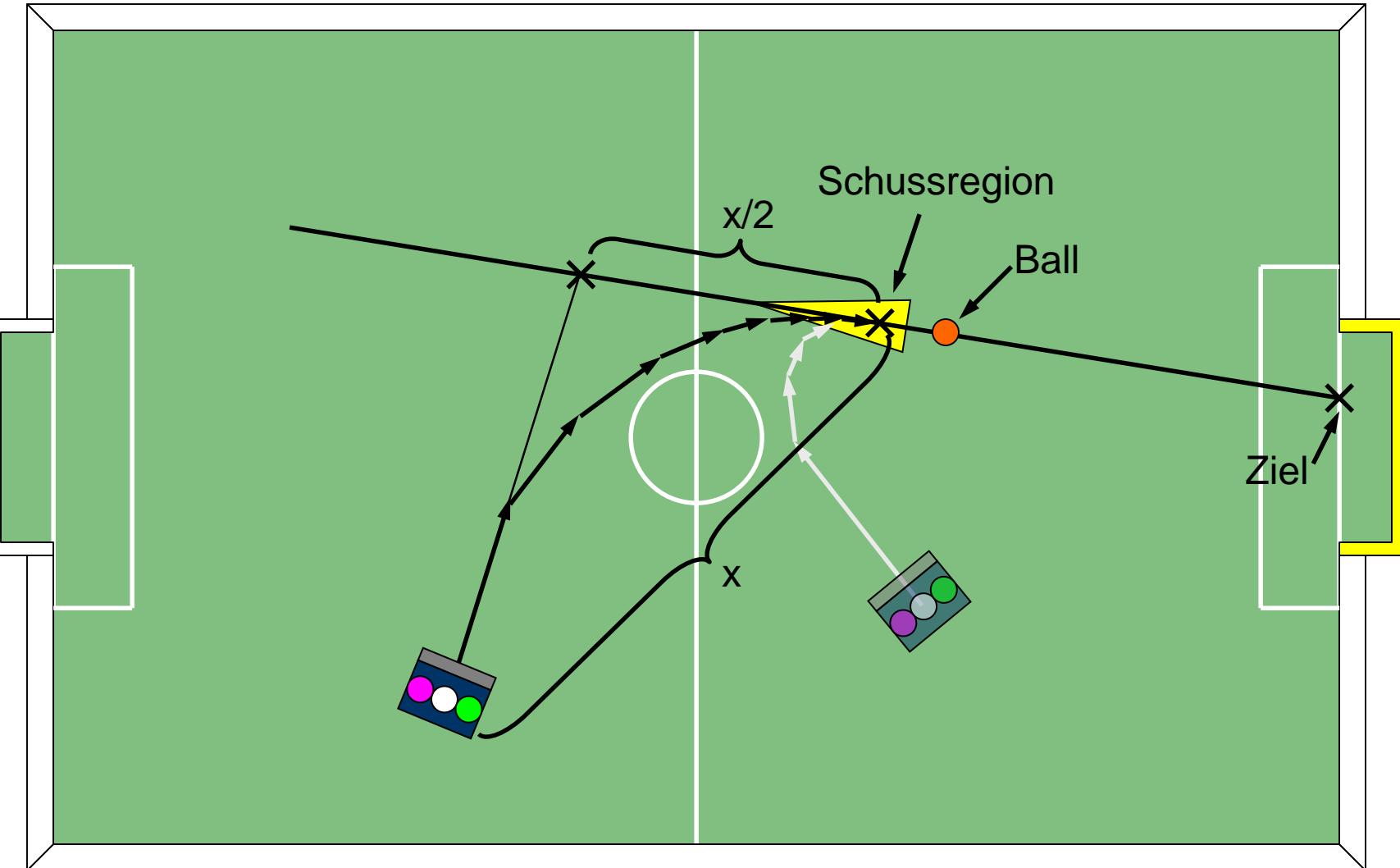


- Mehrere Verhalten können gleichzeitig aktiv sein
- Gerichteter azyklischer Hemmgraph
- Multiplikative Hemmung
- Aktuatorodynamik: Gewichtetes Mittel der Änderungswünsche

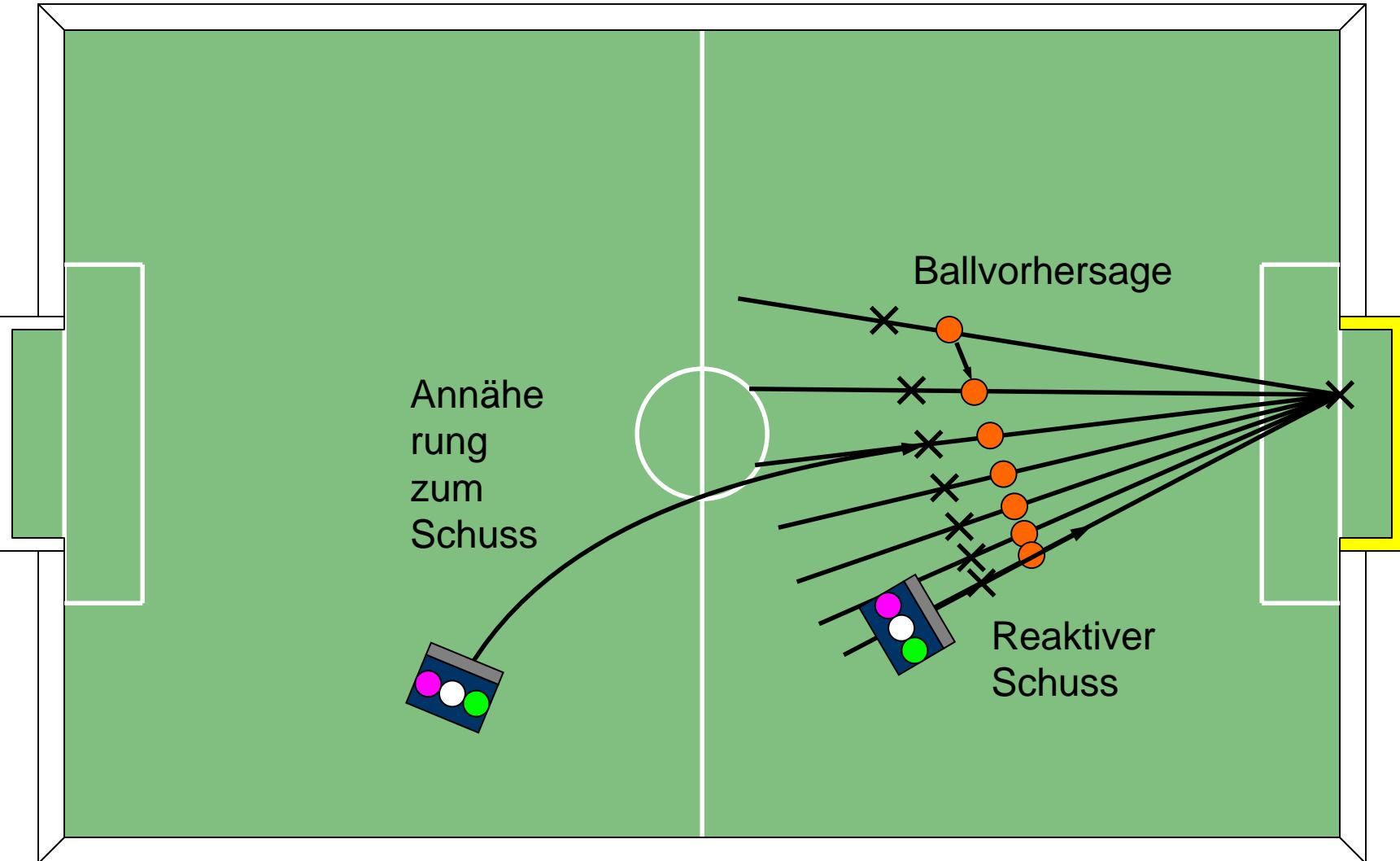
Team-Ebenen



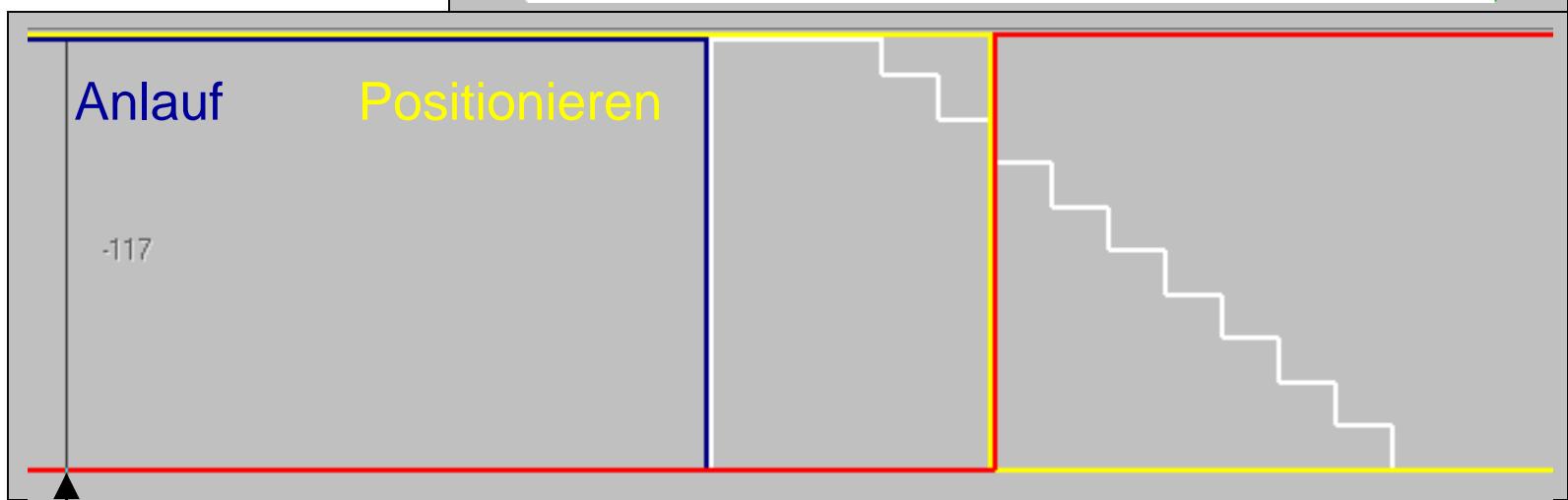
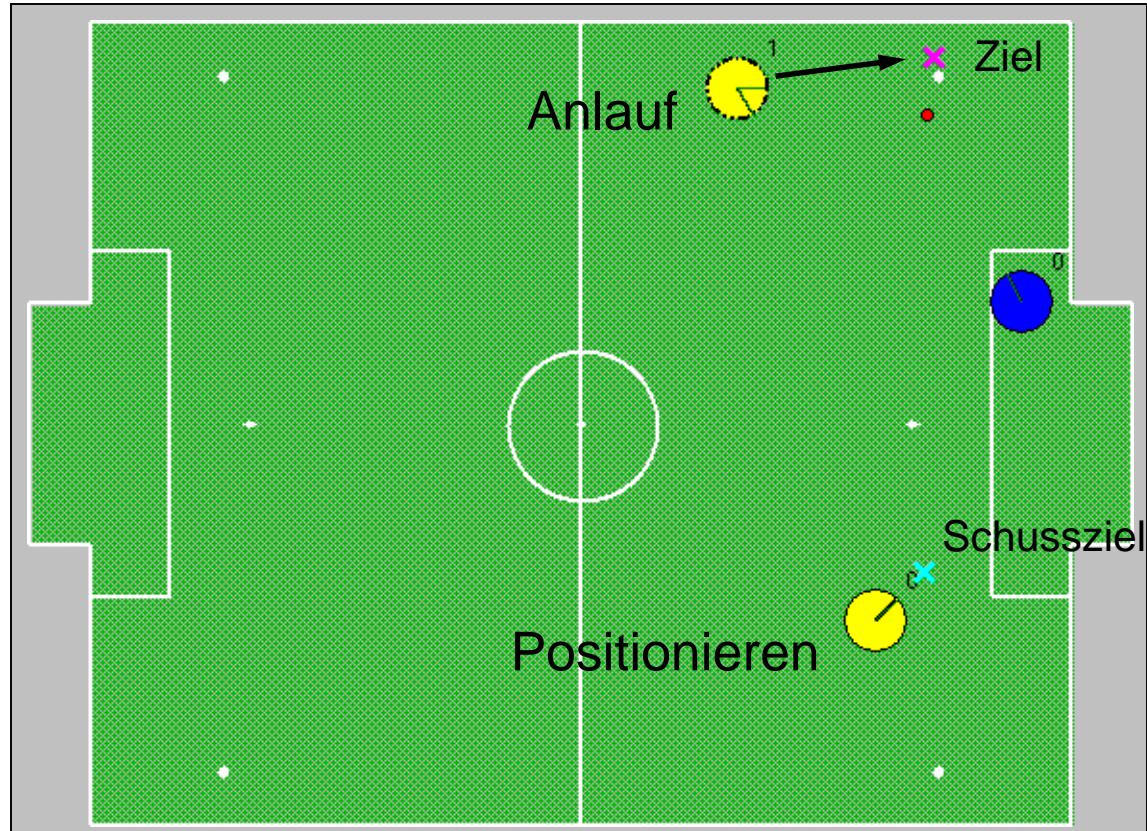
Annäherung zum Schuss



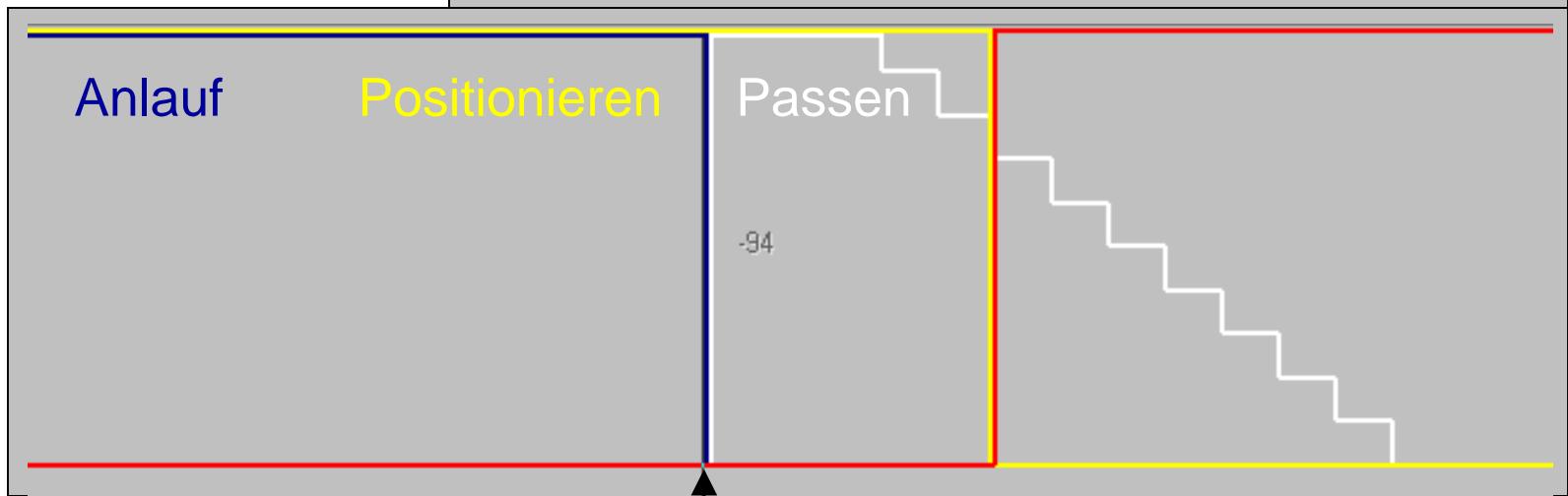
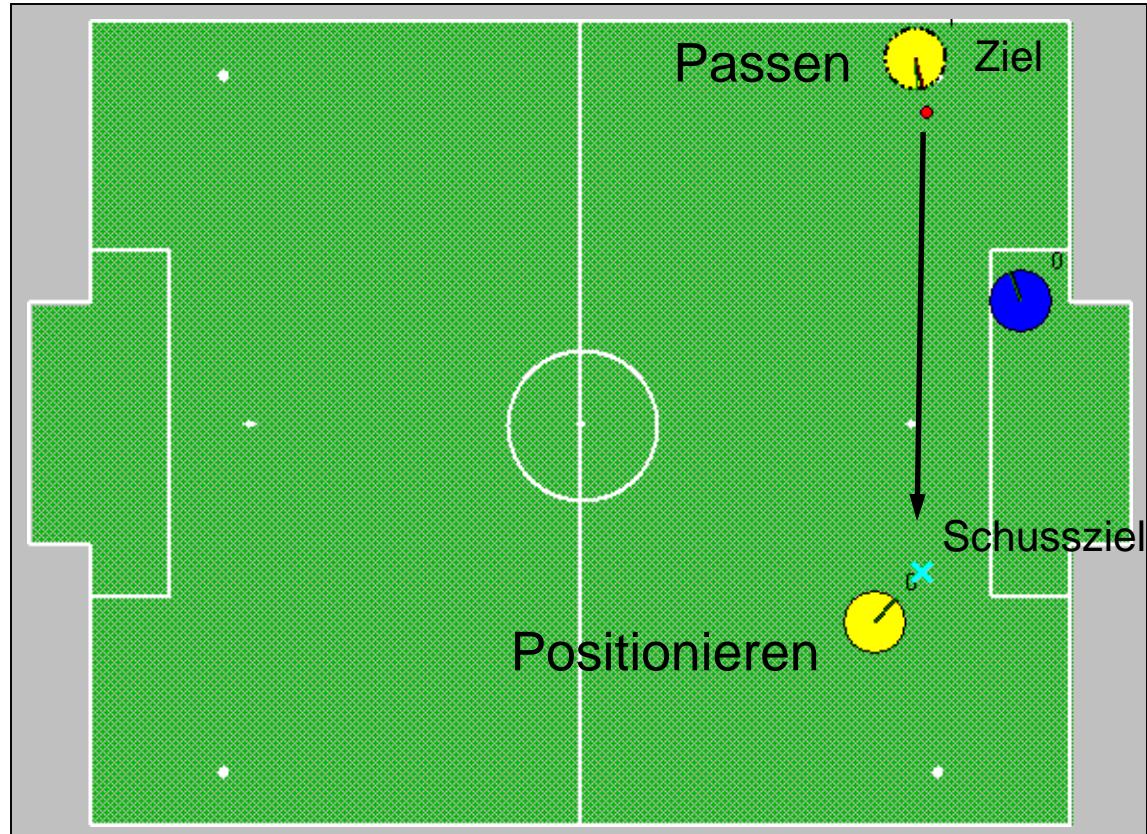
Schuss eines rollenden Balls



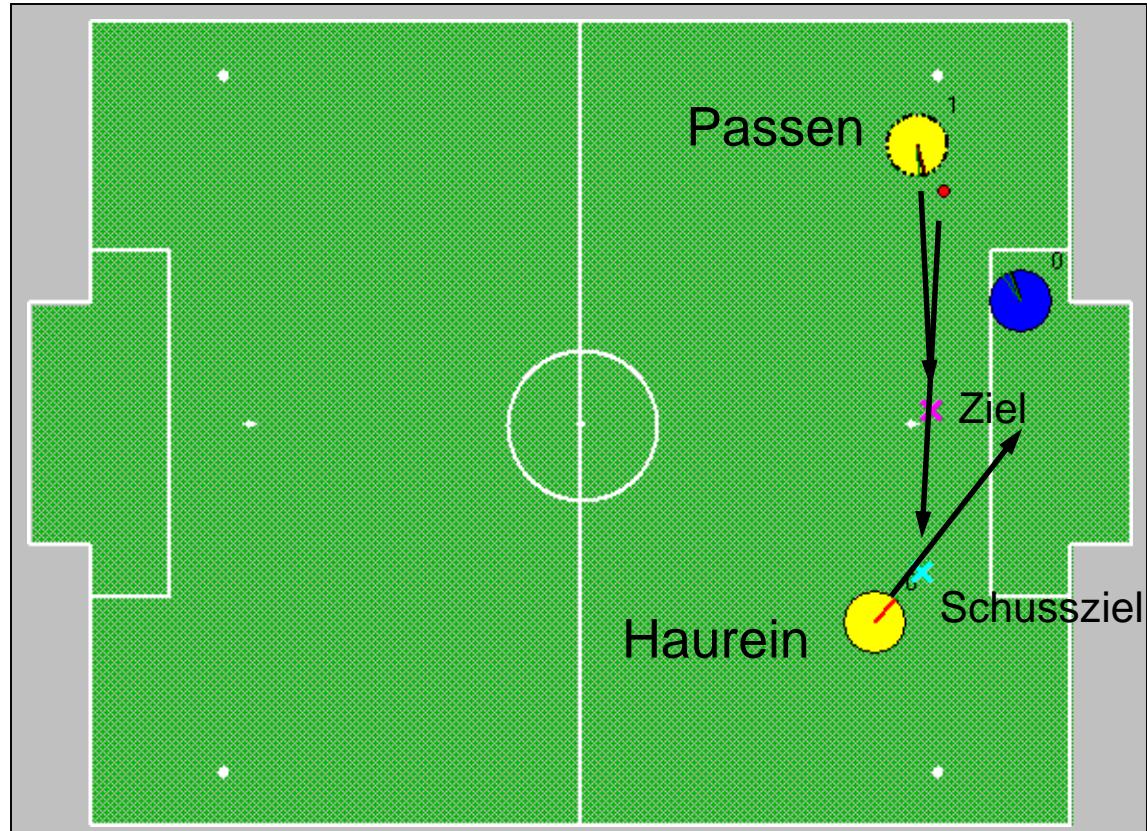
Passen



Passen



Passen



Anlauf

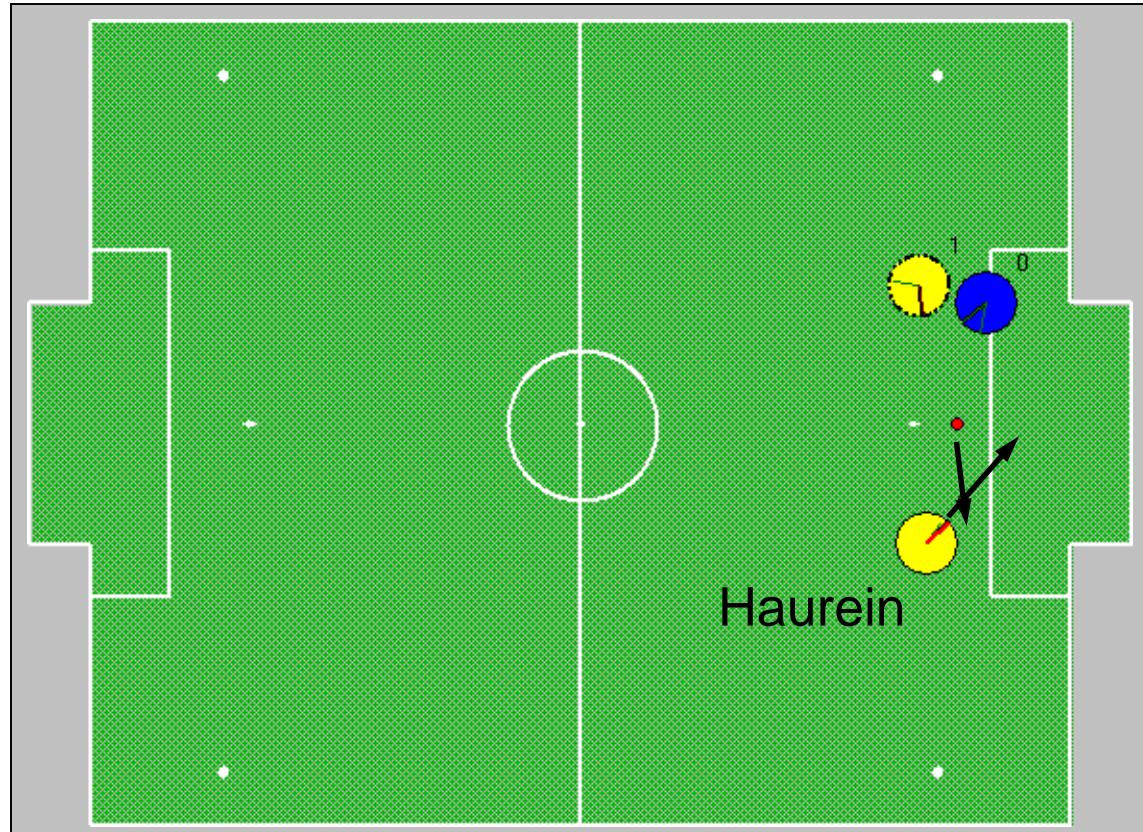
Positionieren

Passen

Haurein

-84

Passen



Anlauf

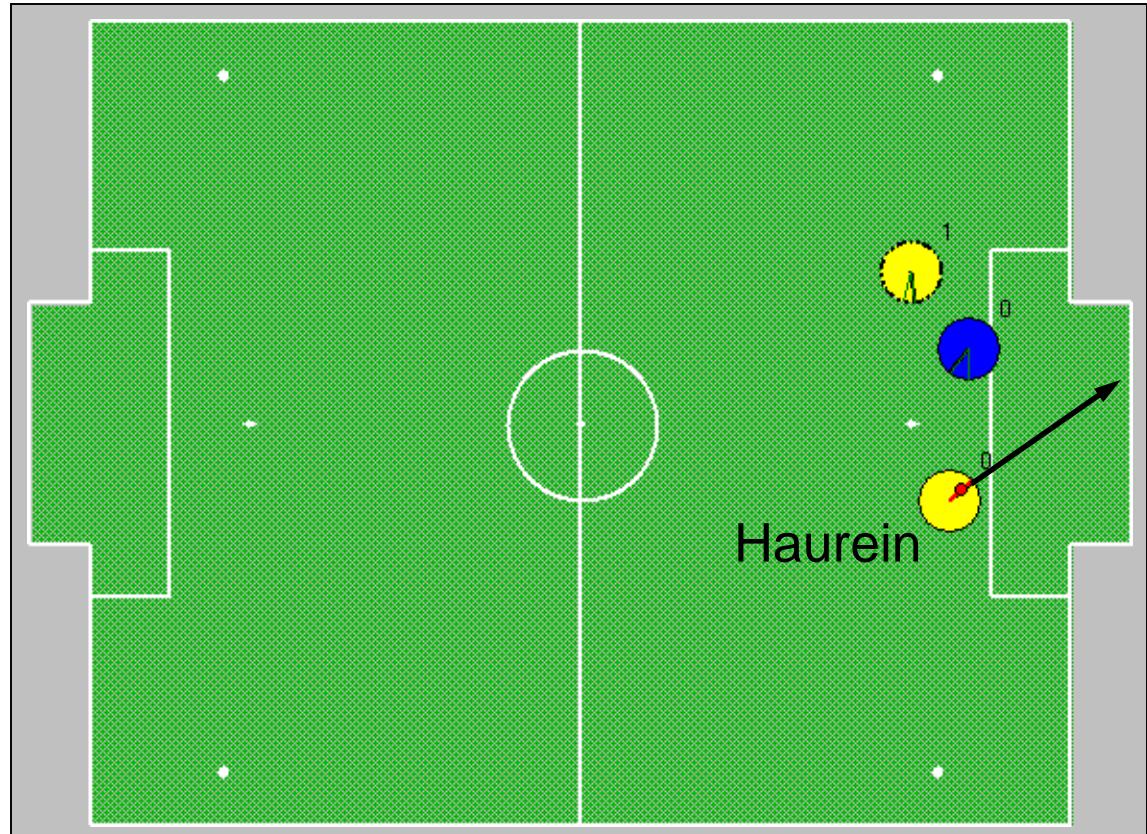
Positionieren

Passen

Haurein

.74

Passen



Anlauf

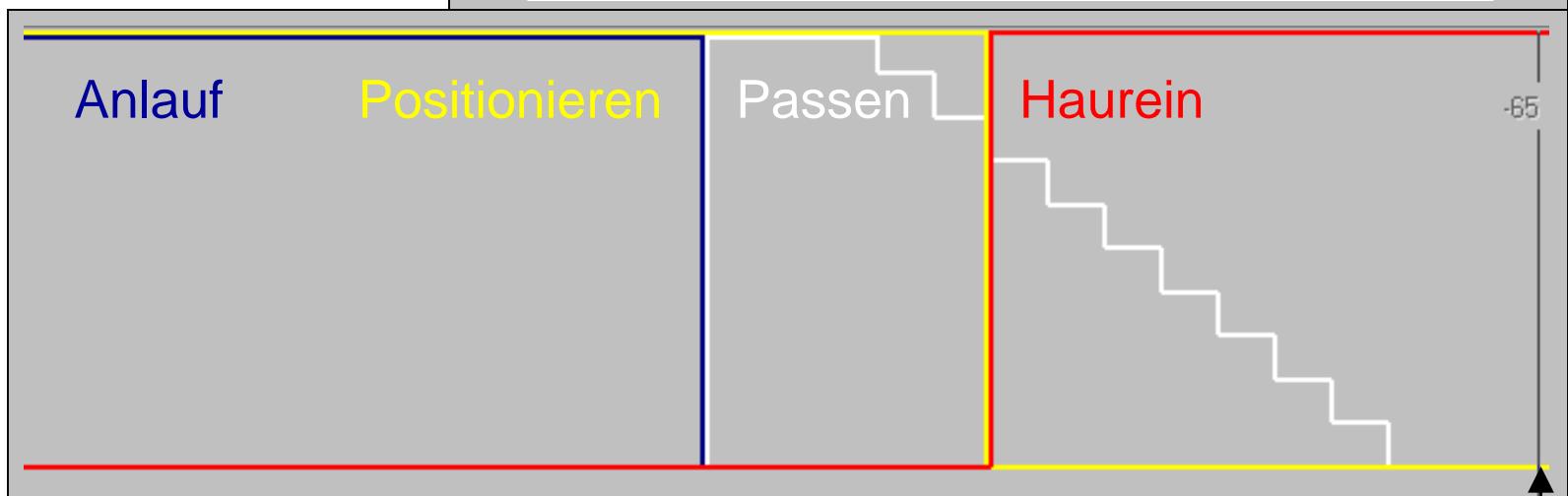
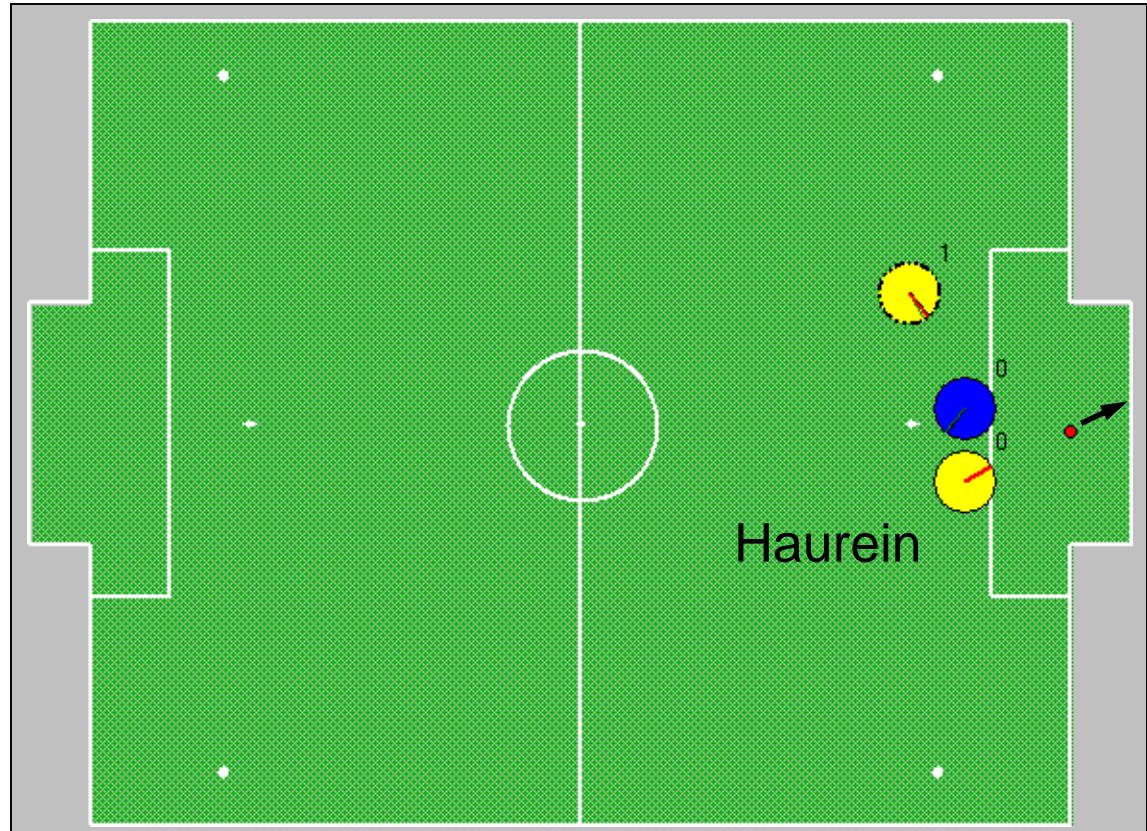
Positionieren

Passen

Haurein

-70

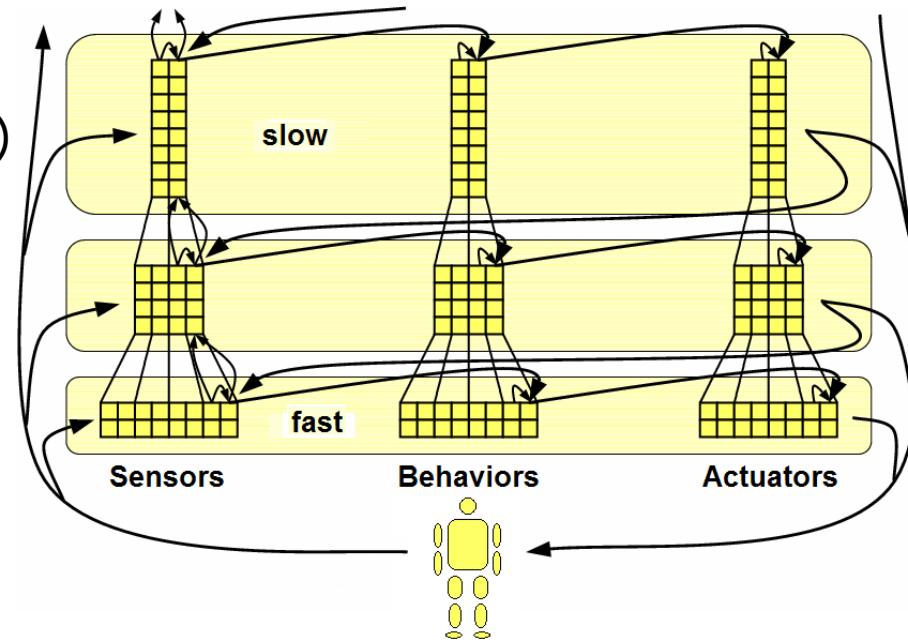
Passen



Hierarchisch-Reaktive Verhaltenssteuerung für humanoide Roboter

■ Hierarchisch, reaktiv

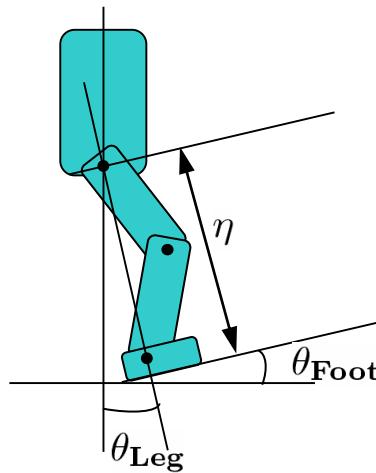
- Zeithierarchie
(kHz, 83 Hz, 41.5 Hz, 20.75 Hz)
- Agentenhierarchie
(Einzelgelenk, Körperteil,
Roboter, Team)
- Komplexitätsreduktion
durch Interaktions-
beschrankungen



■ Kinematische Steuerung von Körperteilen:

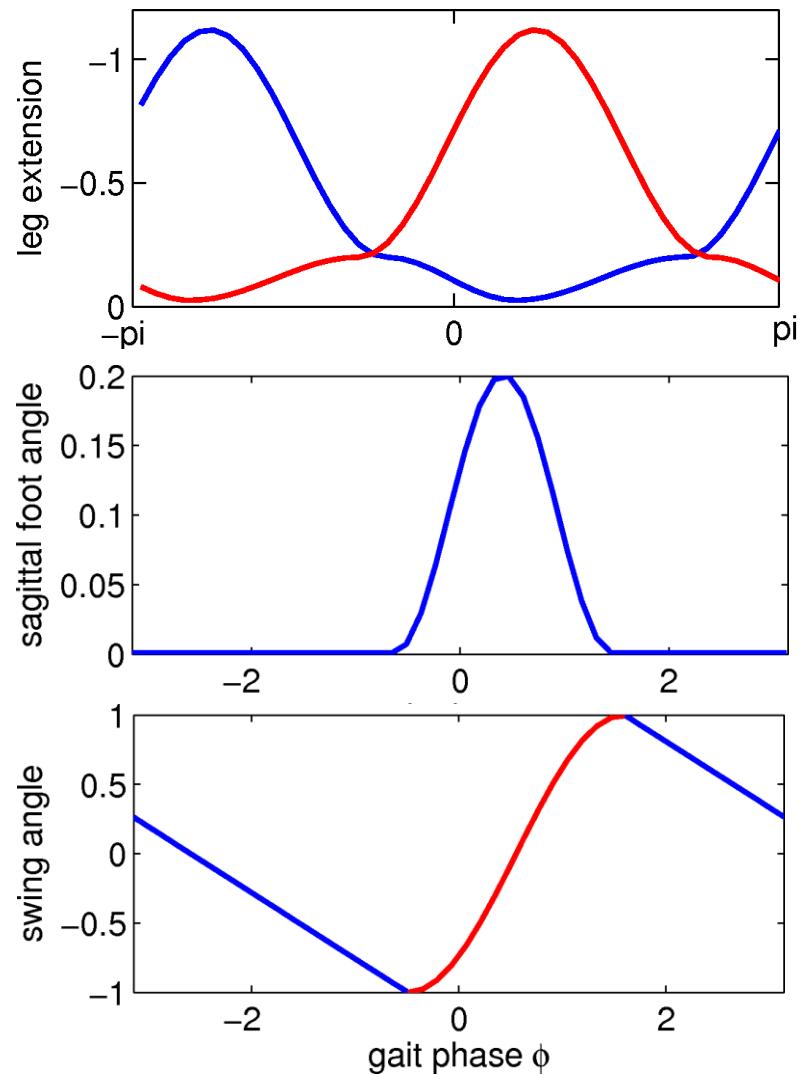
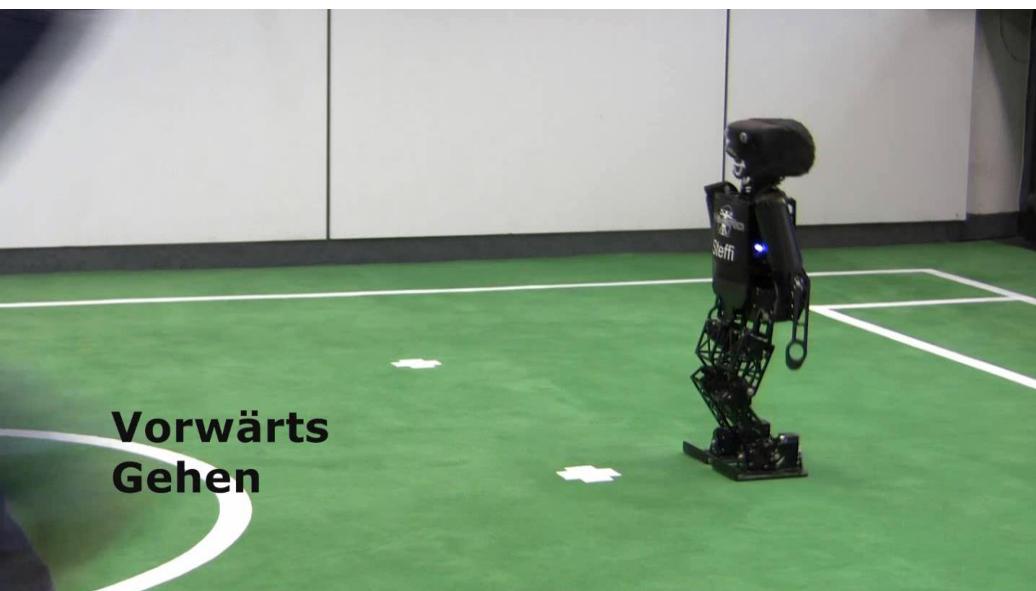
- Beinwinkel
- Fußwinkel
- Beinlänge

[Behnke, Stückler: IJHR'08]



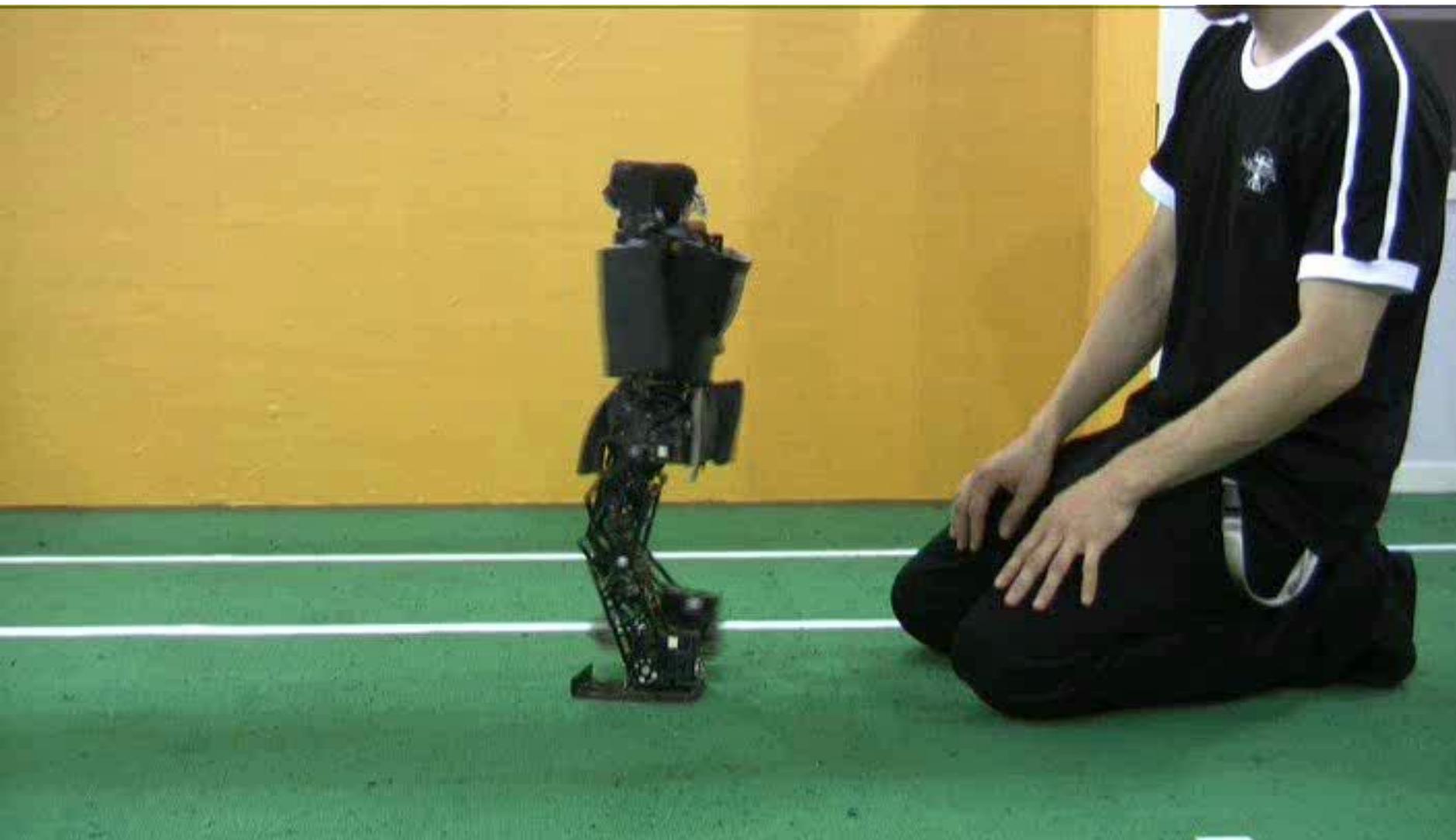
Omnidirektionales Gehen

- Kombination von:
 - Vorwärts/rückwärts Gehen
 - Seitlichem Gehen
 - Drehen auf der Stelle
- Hauptbestandteile:
 - Gewichtsverlagerung
 - Beinverkürzung
 - Bewegung in Laufrichtung



[Behnke: ICRA'06]

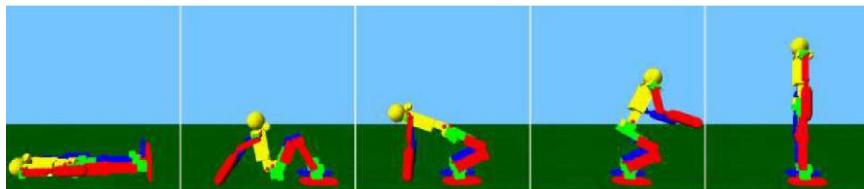
Stabilisierende Reflexe



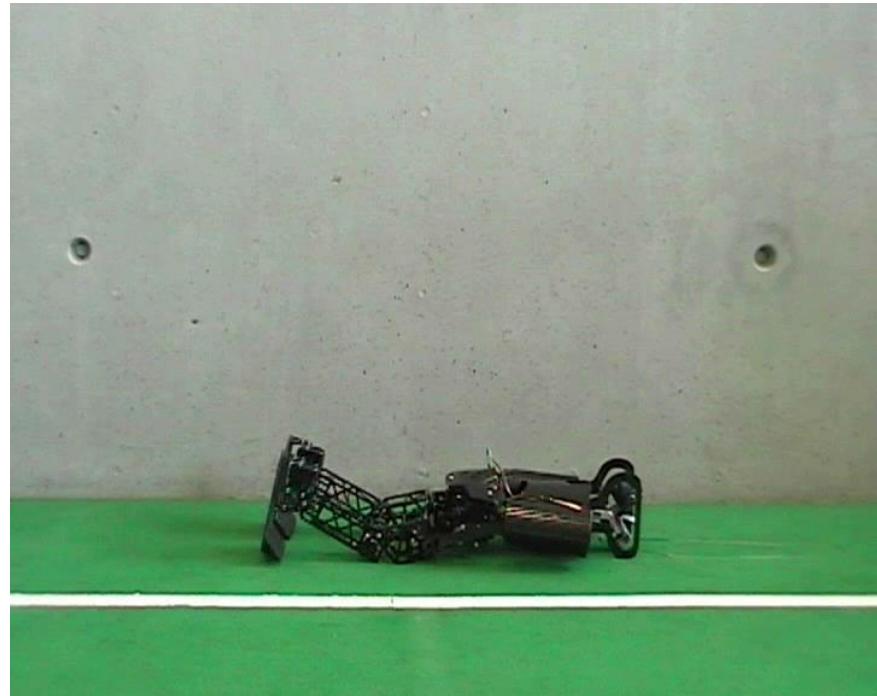
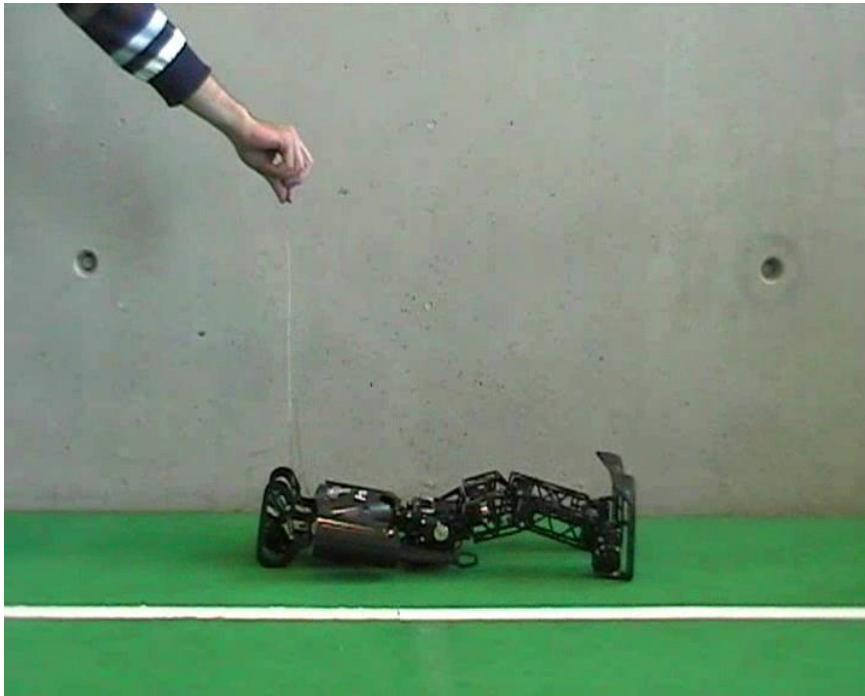
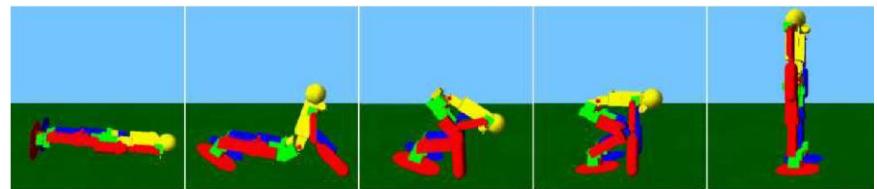
[Behnke, Stückler, Schreiber, TDP'09]

Aufstehen

Rückenlage



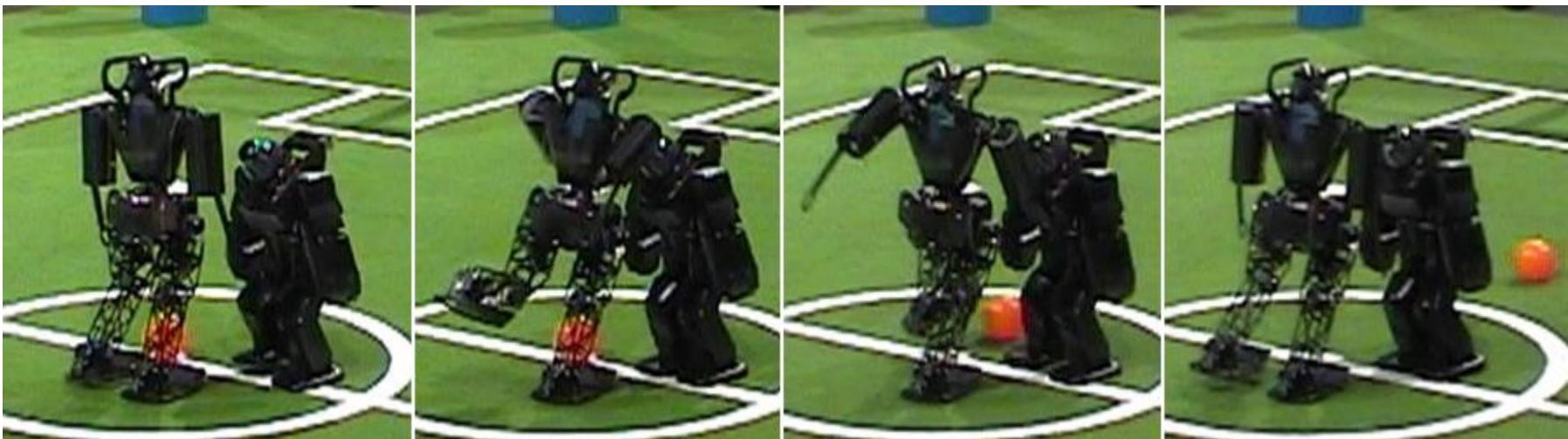
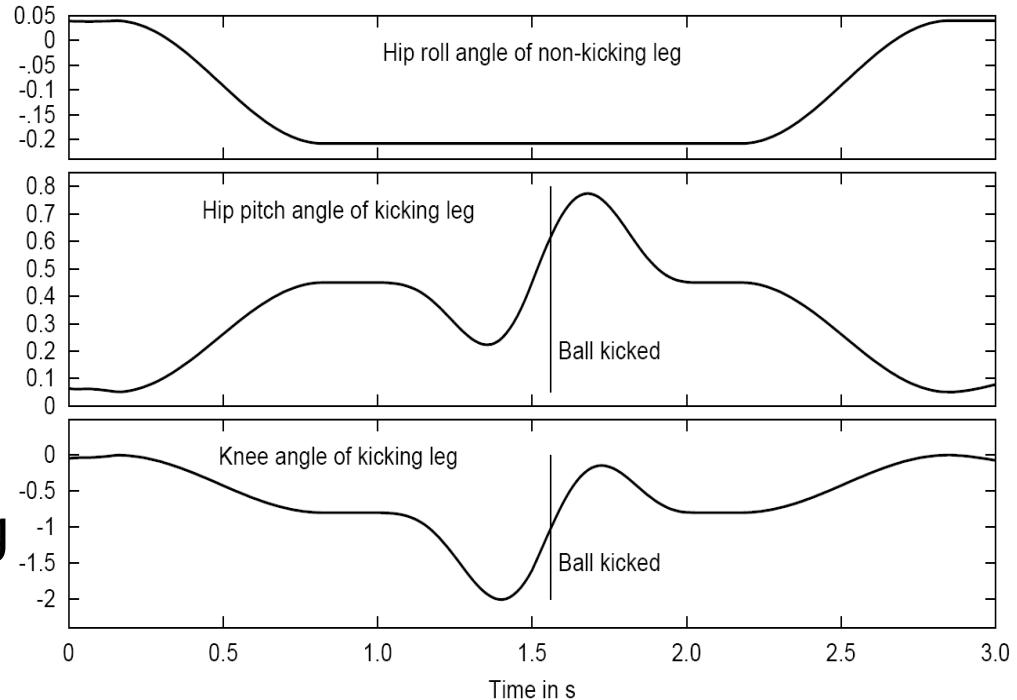
Bauchlage



[Stückler, Schwenk, Behnke: IAS-9, 2006]

Parametrisierbarer Schuss

- Gewichtsverlagerung
- Ausholen
- Schneller Schwung nach vorn
- Abbremsen
- Zurück in Stand
- Schussstärke, -richtung und Ballposition parametrisierbar

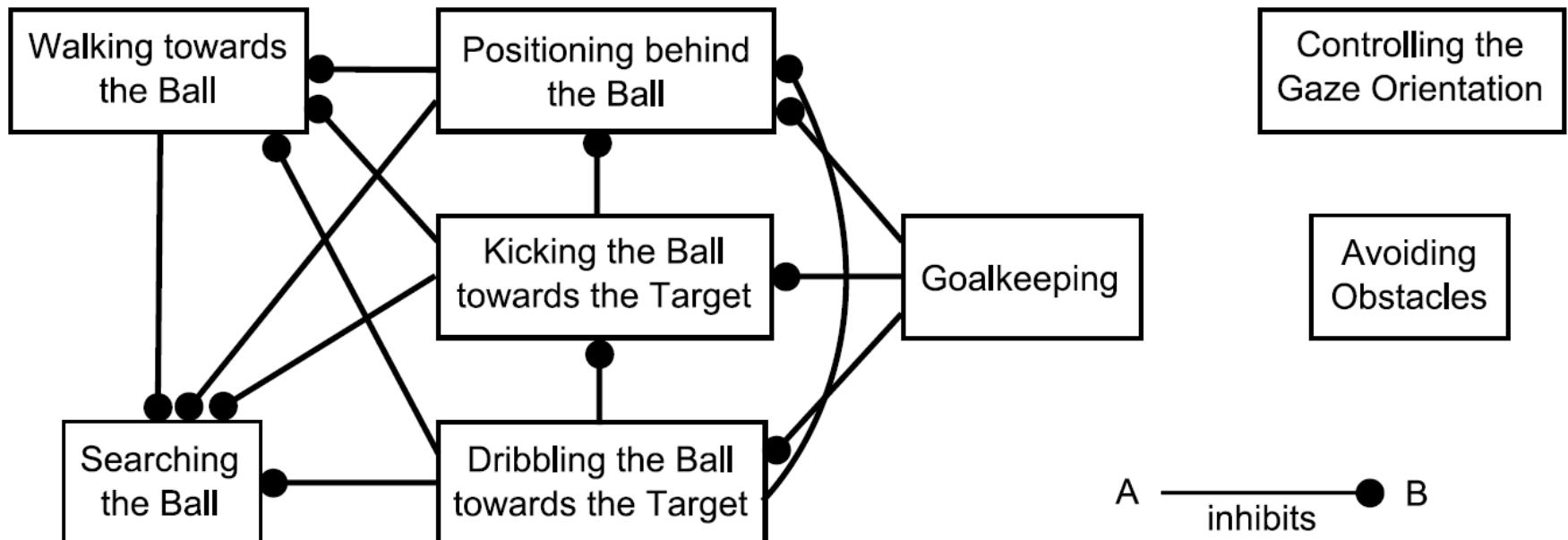


Torwartbewegung zur Seite

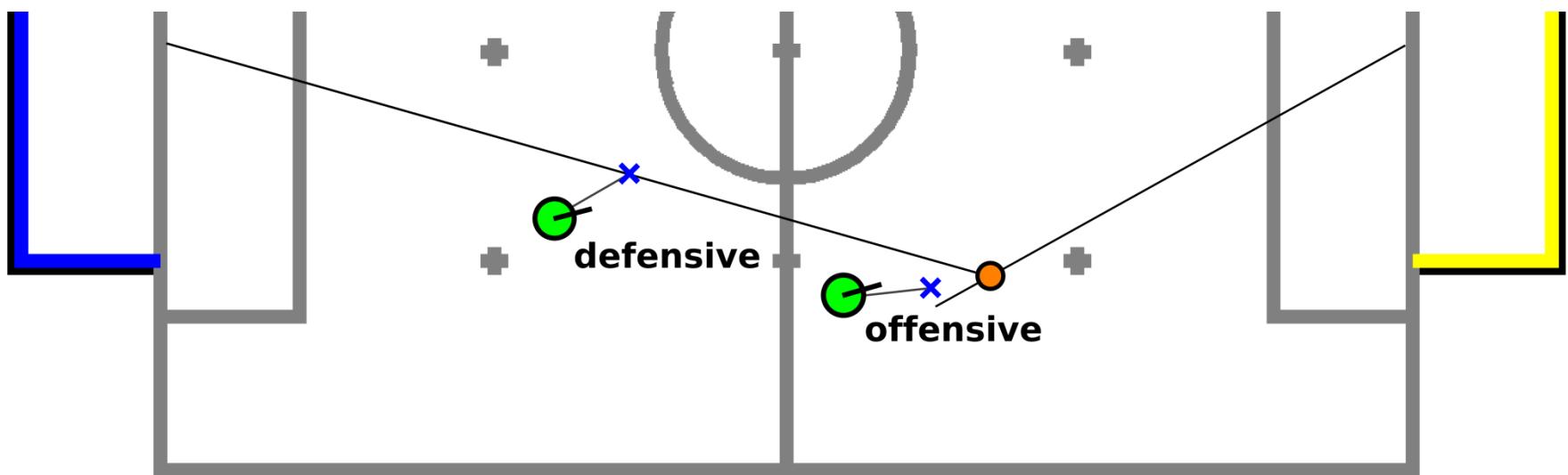


[Missura, Wilken, Behnke: RoboCup'10]

Fußballverhalten



Positionierung der Spieler



Verhaltenshierarchie

■ Taktik und Teamverhalten

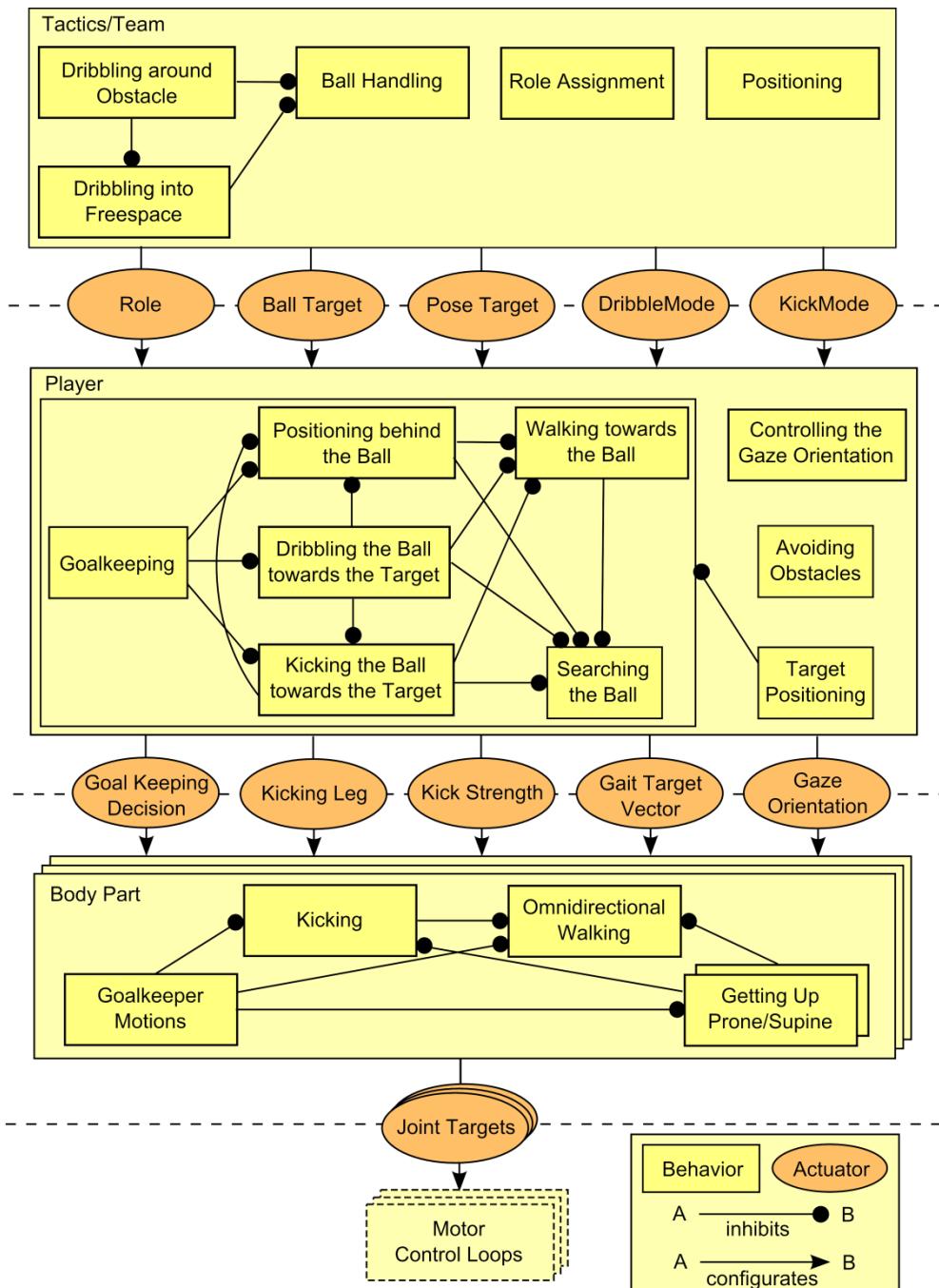
- Rollenzuweisung
- Positionierung auf dem Feld
- Ball bewegen
- Um Hindernisse Dribbeln
- In den Freiraum Dribbeln

■ Fußballverhalten

- Ballsuche
- Zum Ball laufen
- Hinter dem Ball positionieren
- Schuss in Richtung Ballziel
- Dribbling in Richtung Ballziel
- Hindernisvermeidung
- Aktive Kameraausrichtung
- Torwartverhalten

■ Grundlegende Fähigkeiten

- Omnidirektionales Gehen
- Schuss
- Aufstehen
- Torwartbewegungen



RoboCup 2008 KidSize-Finale



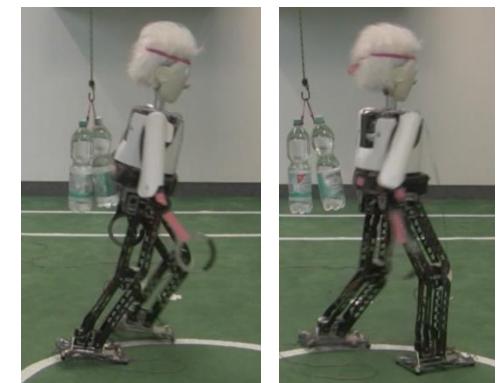
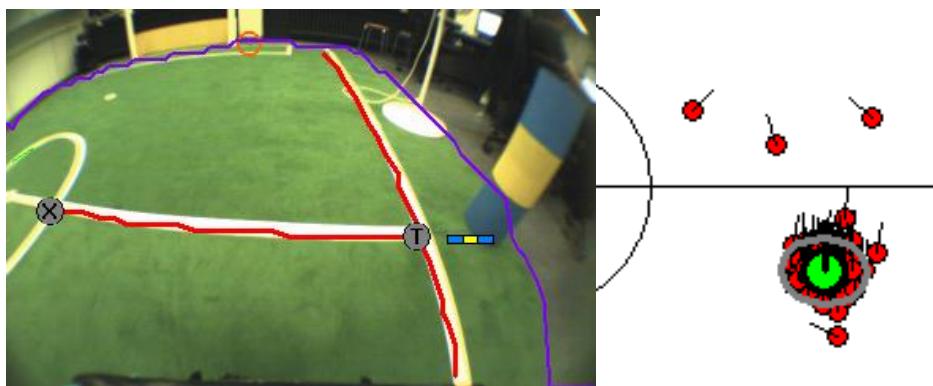
Humanoide Fußballroboter

- Größe: 90-114 cm; Gewicht: 6-8 kg; DoF: 13-20



Unser Team NimbRo hat die RoboCup Humanoid TeenSize Wettbewerbe 5x hintereinander gewonnen (2009-2013).

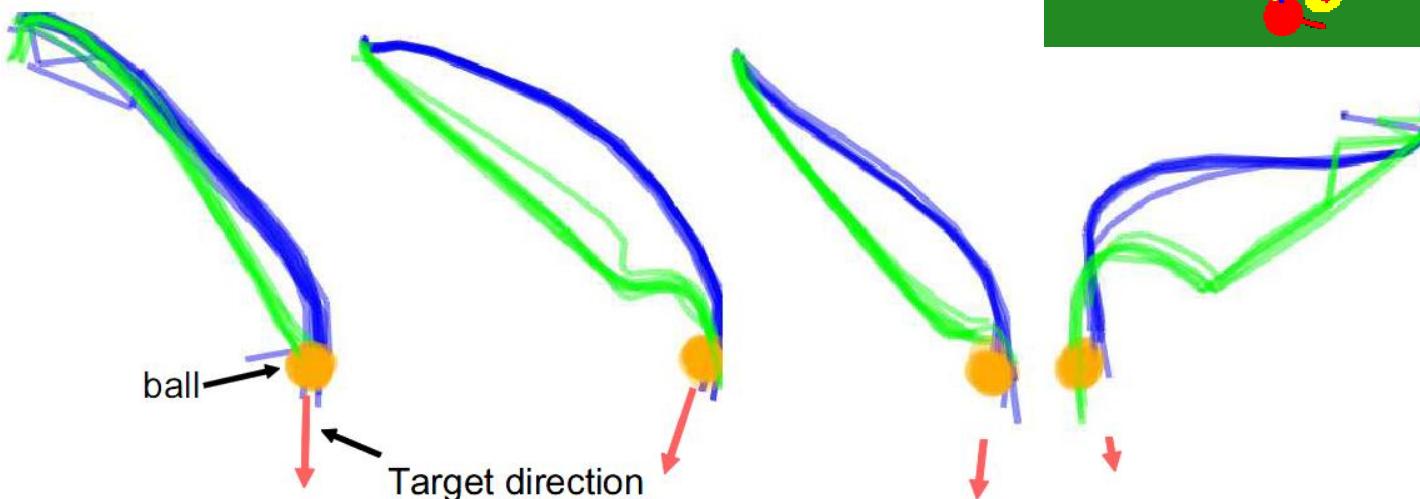
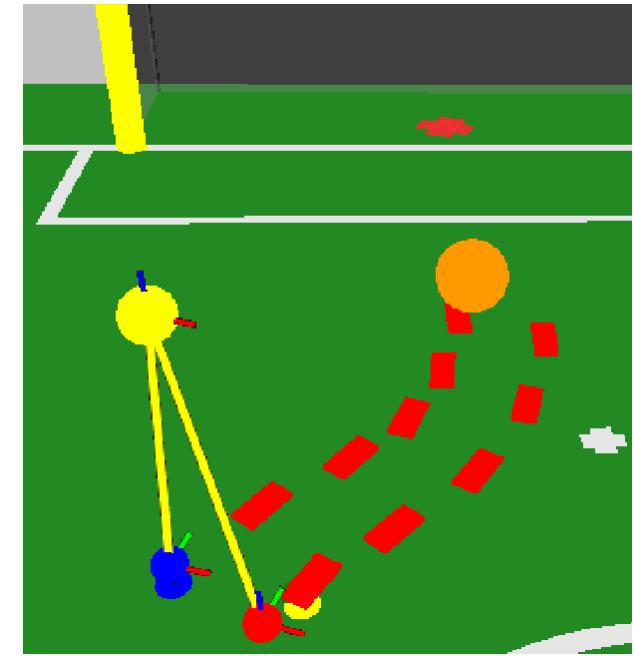
- Probabilistische Lokalisierung, Gangstabilisierung



[Missura and Behnke, IROS 2025]

Planung einzelner Schritte für Fußballroboter

- Ziel: Treffen des Balls ohne Verlangsamung
- Gelerntes Bewegungsmodell
- Offline-Planung
- Funktionsapproximator für ersten Schritt

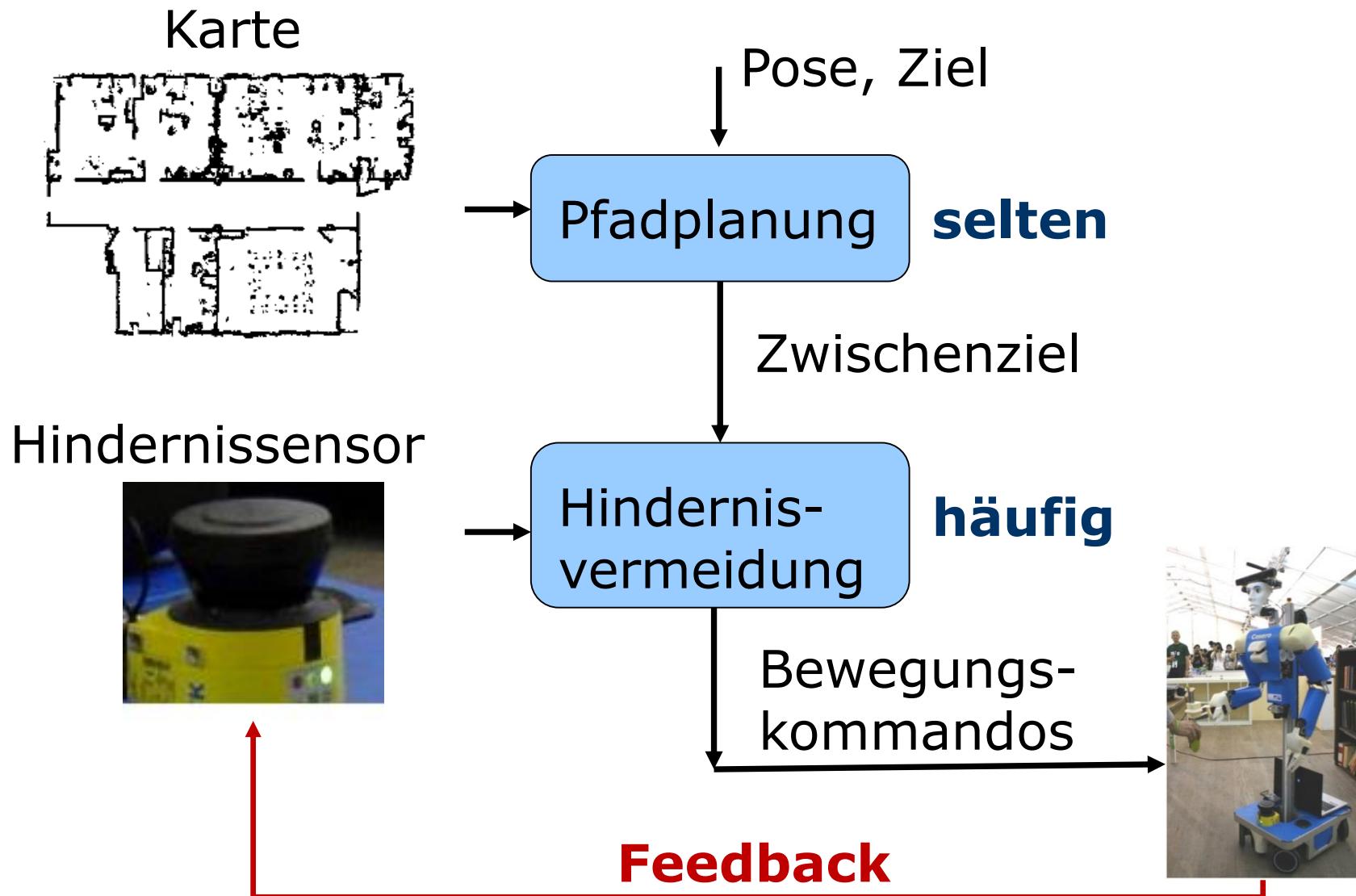


[Schmitz, Missura, Behnke, RoboCup 2011]

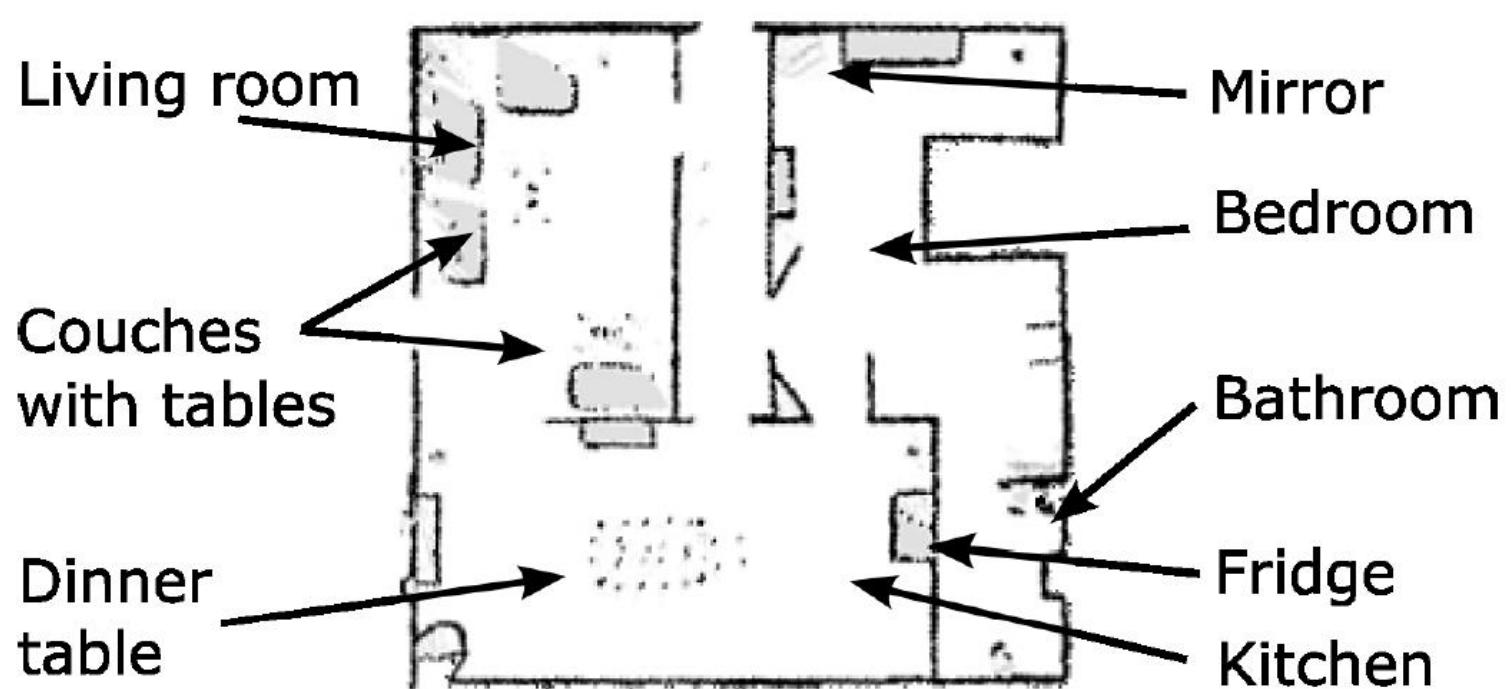
Hybride Verhaltenssteuerung

- Versuch, das Beste aus beiden Welten (deliberativ und reaktiv) zu vereinen
- Beispiel: Drei-Schichten-Architektur (Erann Gat 1998: Alfred)
 - **Deliberative Schicht** (Planer)
 - Repräsentation der Zukunft
 - Langsam, Ziel-getrieben
 - **Zwischenschicht** (Sequenzer)
 - Repräsentation der Vergangenheit
 - Abstraktion
 - Fehlererkennung, Alternativerhalten
 - **Reaktive Schicht** (Controller)
 - Kein Zustand, schnell, Sensor-getrieben

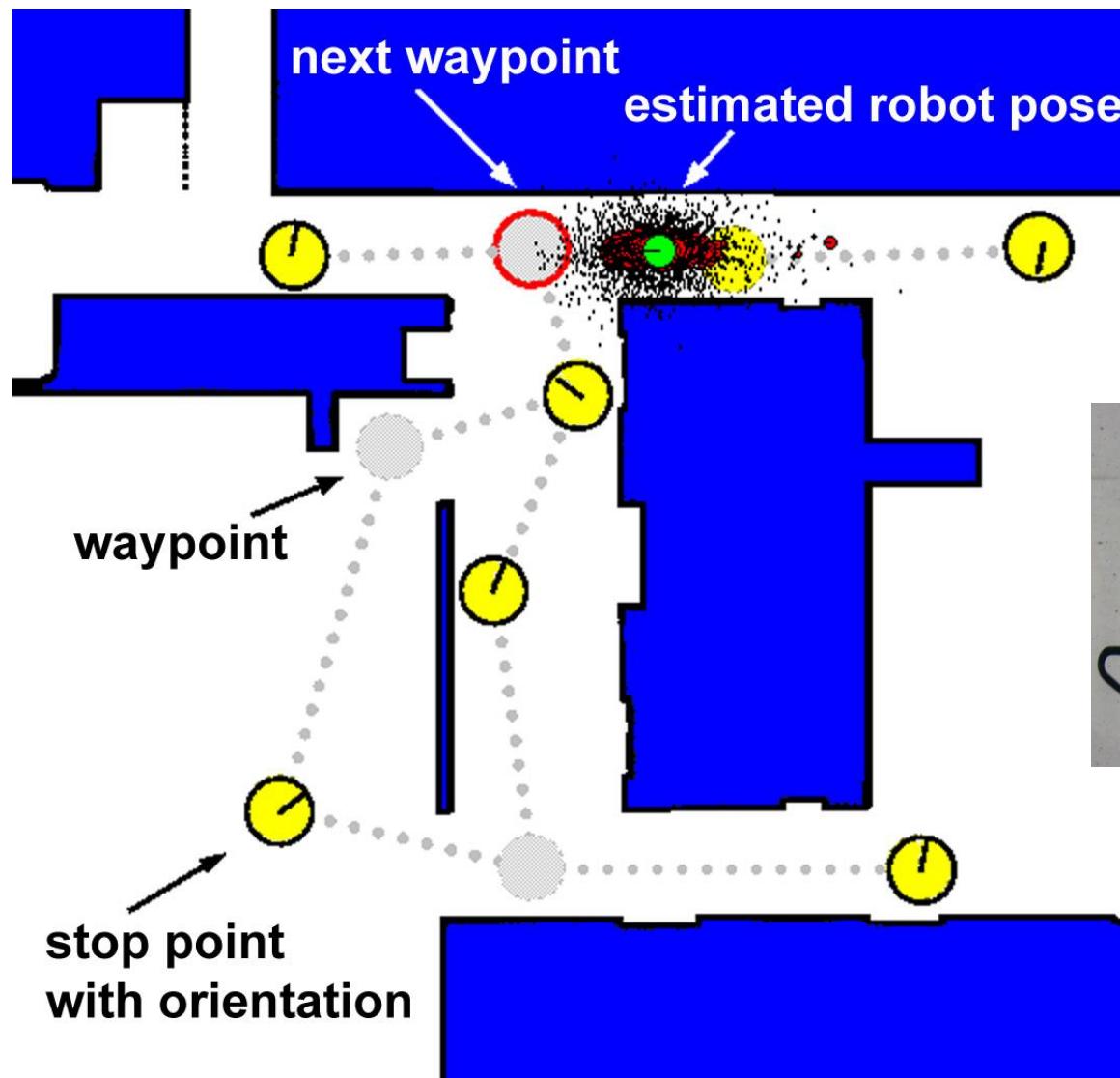
Zweischicht-Architektur zur Navigation



Kartierung der Umgebung



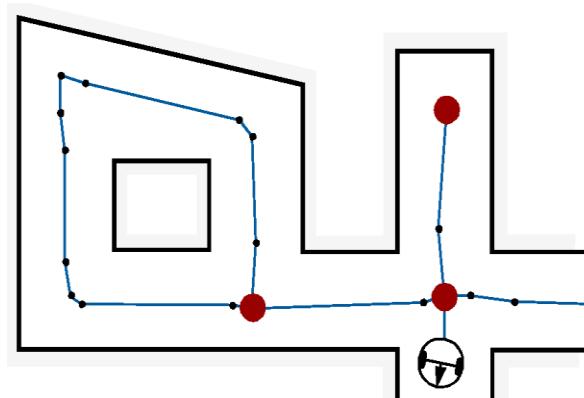
Navigation für Kommunikationsroboter



Überblick Pfadplanung

Routengraphen

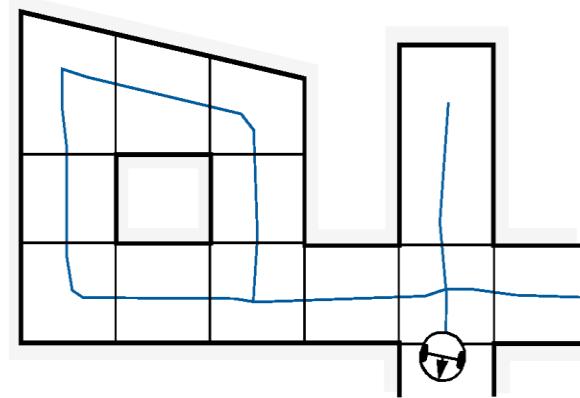
- Knoten und Kanten



- Platzierung der Knoten an ausgezeichneten Stellen
 - Abbiegungen
 - Änderung der Sichtbarkeit

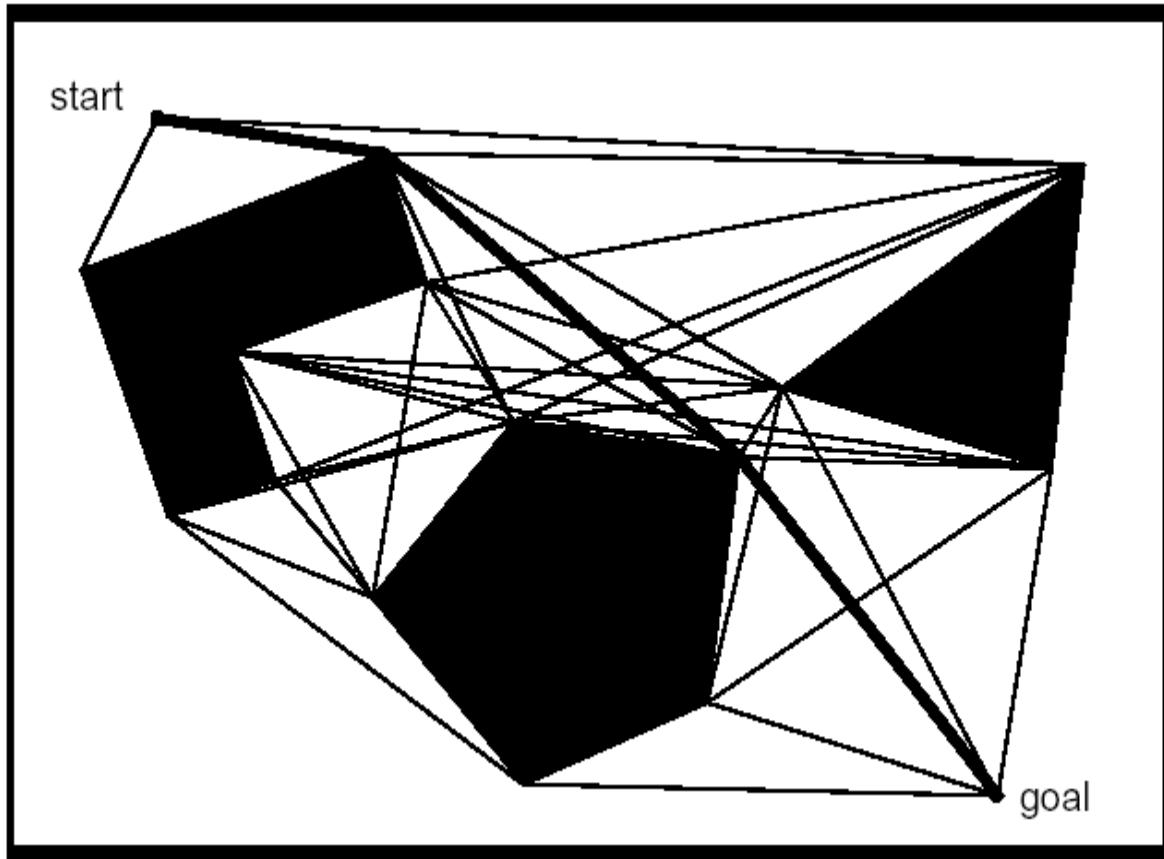
Zelldekomposition

- Unterscheidung von frei und belegt



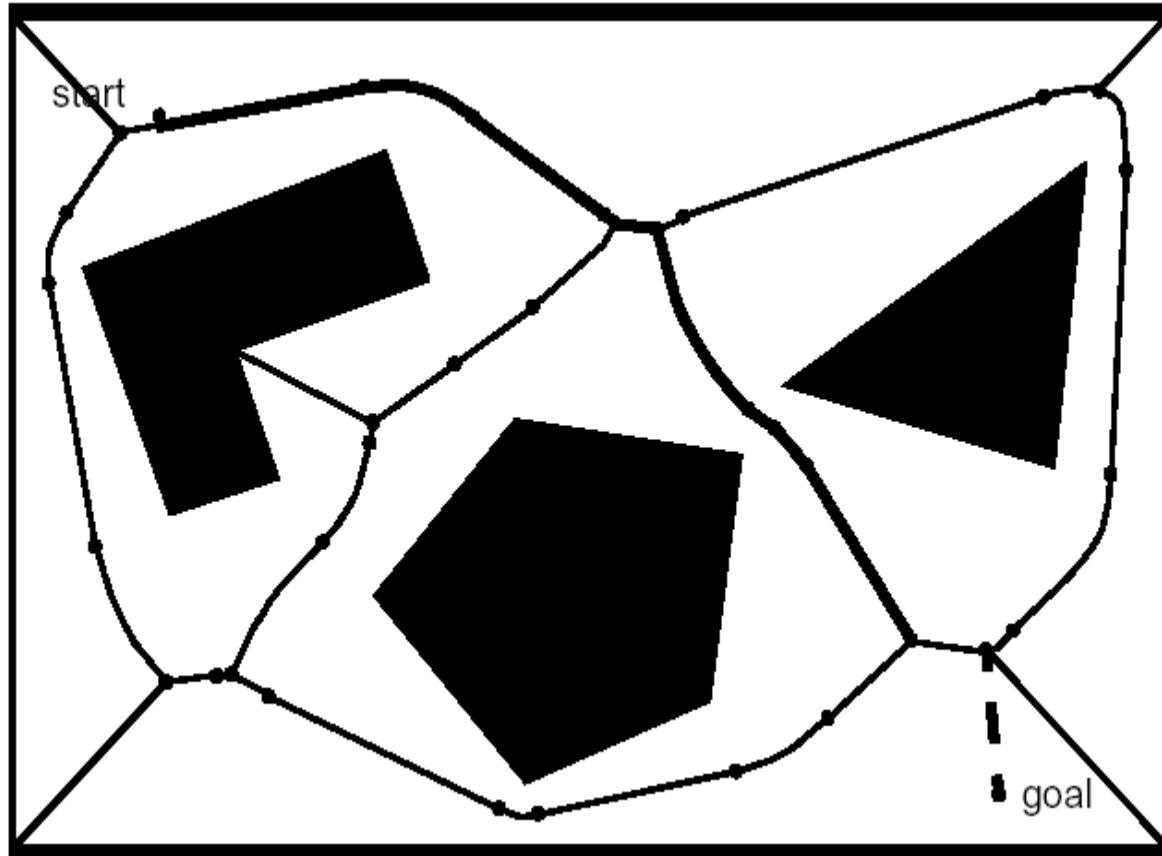
- Zellengrenzen willkürlich oder von Karte abgeleitet

Sichtbarkeitsgraphen



- Knoten: Ecken von Hindernis-Polygonen, Start und Ziel
- Kanten: Gerade Verbindungen wenn nicht blockiert
- Durch Graphsuche gefundener Pfad hat kürzeste Pfadlänge
- Erweitere Hindernisse um Roboterradius!

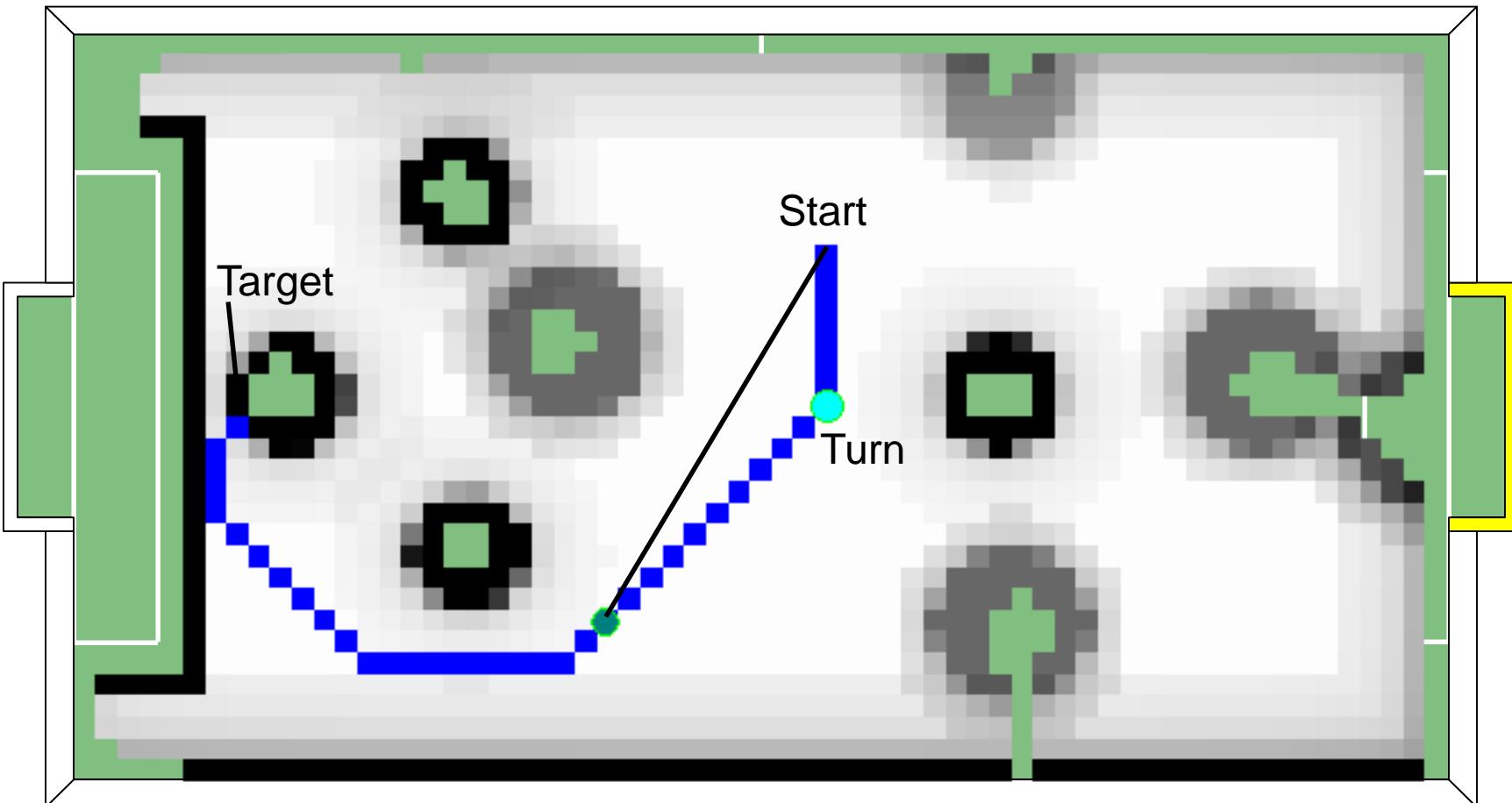
Voronoi-Routengraphen



- Knoten und Kanten sind zu den nächsten Hindernissen gleich weit entfernt
- Pfad maximiert den Abstand zu Hindernissen

Zelldekomposition mit Kosten

- Grid, 8-Nachbarschaft, Beispiel Roboterfußball



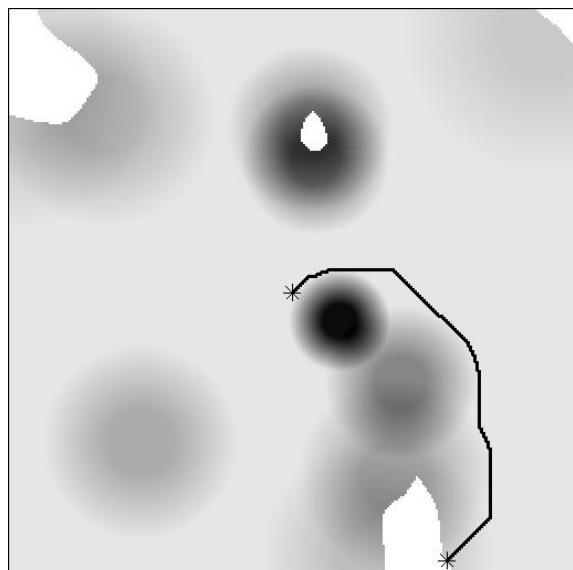
- Hindernisse sind Roboter, Feldgrenze, Torraum
- Pfad durch heuristische Suche (A^*) bestimmt

A* – Suche

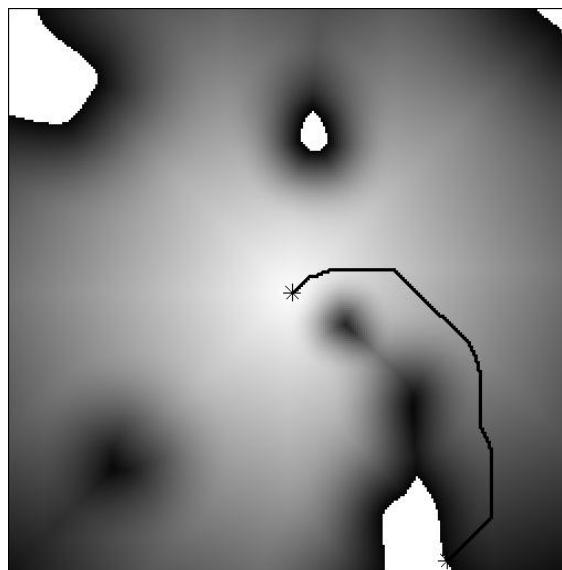
```
PlanPath(target, obstacles, N){ // Gegeben Positionen von Ziel und
    grid = eval = ClearGrid();      Hindernissen, Nachbarschaftsgraph
    q = InitPriorityQueue(); // Startknoten einfügen
    while (!q.empty()){
        p = q.pop(); // nächster zu expandierender Knoten
        if (target == p) // Sind wir fertig?
            return previous; // Ausgabe des Pfades
        for(n ∈ N(p)){ // für alle Nachbarn
            if(eval(n) == 0) // Kosten bisher
                best = ∞;
            else
                best = eval(n); // neue Kosten
            new = p.cost + grid(p)·n.l0 + getGrid(n, obstacles, grid)·n.l1;
            if(new < best){ // Vergleich mit alten Kosten
                eval(n) = new; // Hinzufügen zur Prioritätsschlange
                q.push([n, new, getHeuristic(n, target)]);
                previous(n) = p; // Vorgänger merken
            }
        }
    }
}
```

Effekt der Heuristik

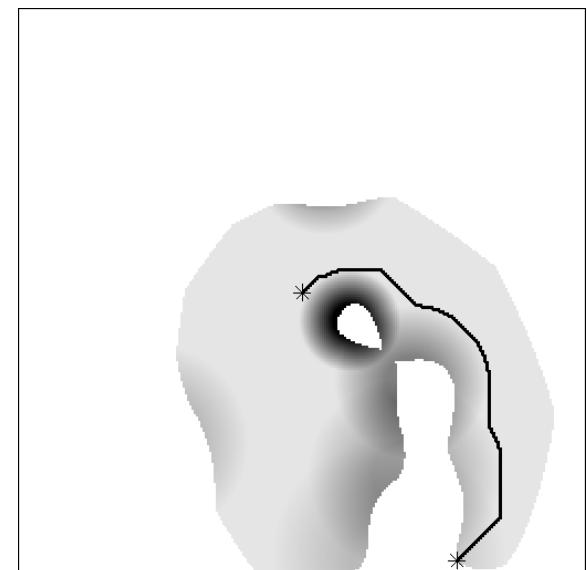
- Heuristik schätzt Kosten, die mindestens noch bis zum Ziel anfallen (Euklidscher Abstand zum Ziel multipliziert mit hindernisfreien Kosten)
- Reduziert Expansion des Suchraums



Ohne Heuristik



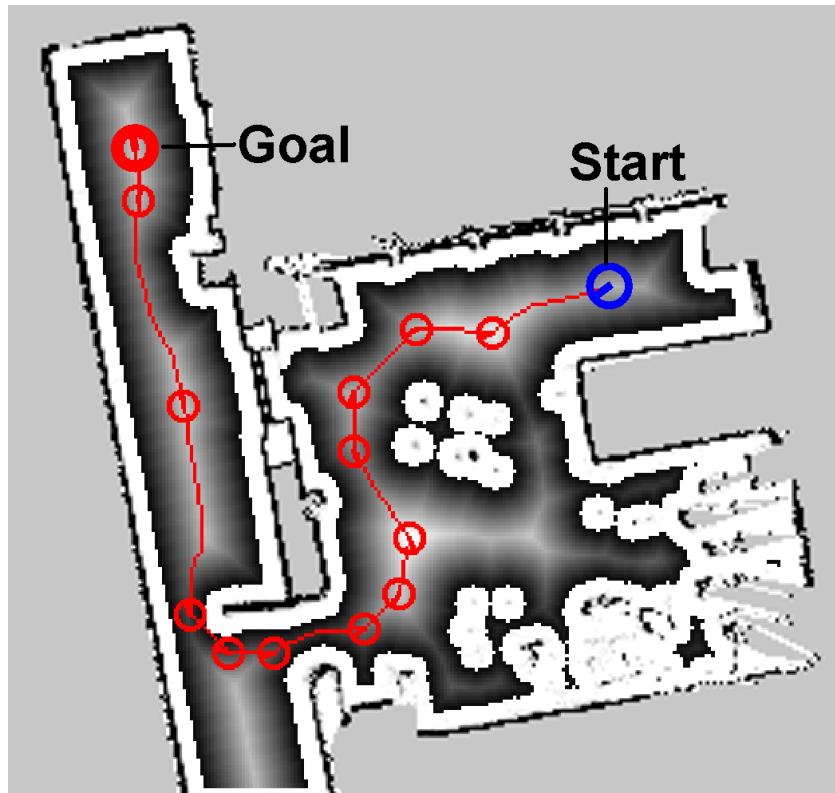
Aggregierte Kosten



Mit Heuristik

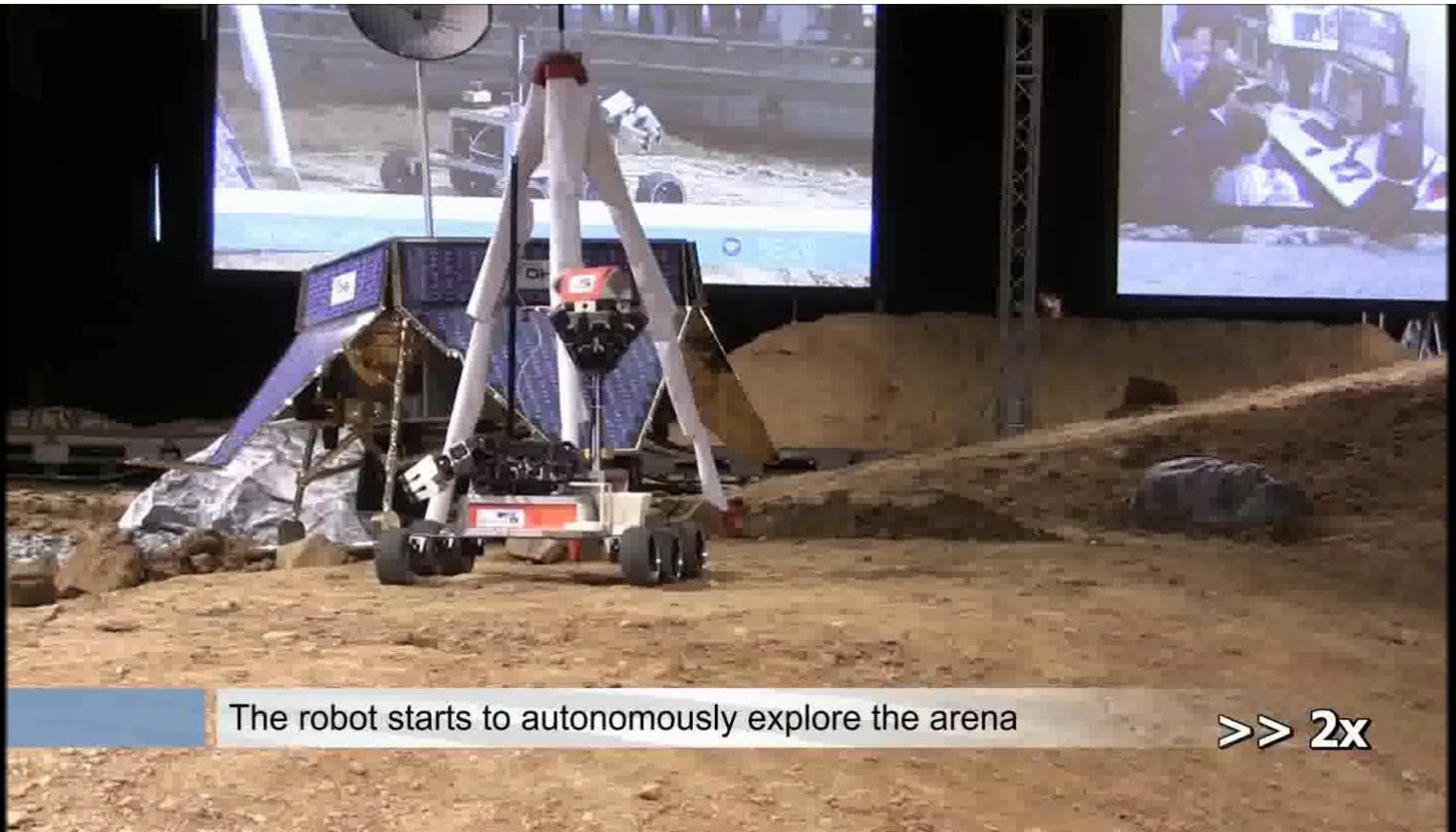
Pfadplanung in Büroumgebung

- Omnidirektionales Fahrwerk mit vier lenkbaren Radpaaren
- Zellkosten steigen in der Nähe der Wände
- A* findet kostengünstigsten Pfad
- Ausrichtung in Engstellen
- Neuplanung wenn Pfad blockiert



RoboCup 2013

NimbRo Explorer @ DLR SpaceBot Cup



The robot starts to autonomously explore the arena

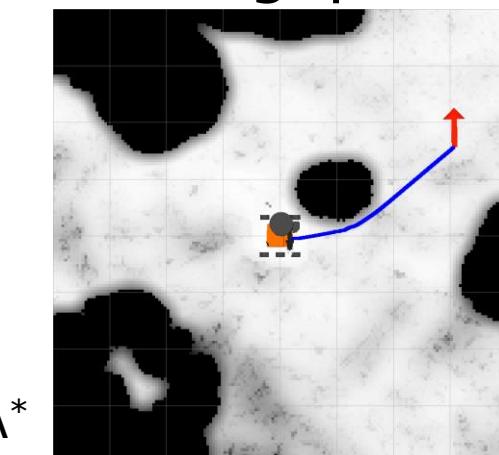
>> 2x

Lokale Navigationsplanung

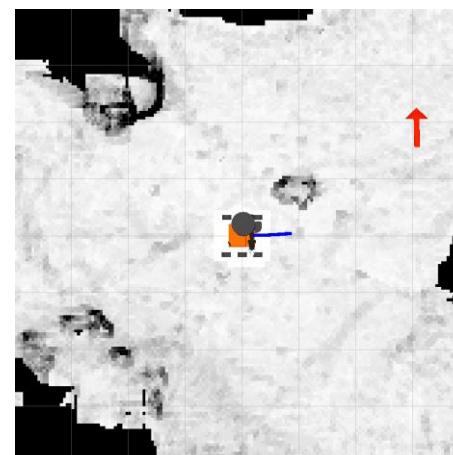
- Lokales Geländemodell aus Rundum-RGB-D-Kamera



- Zellkosten und geplanter Pfad



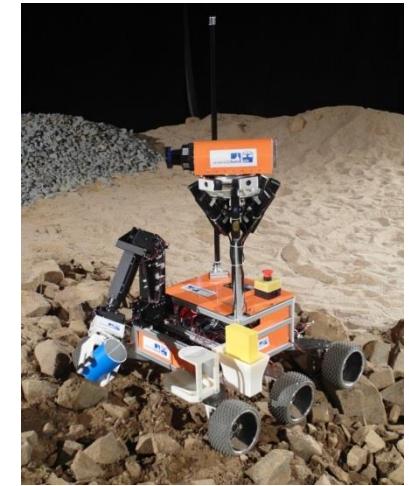
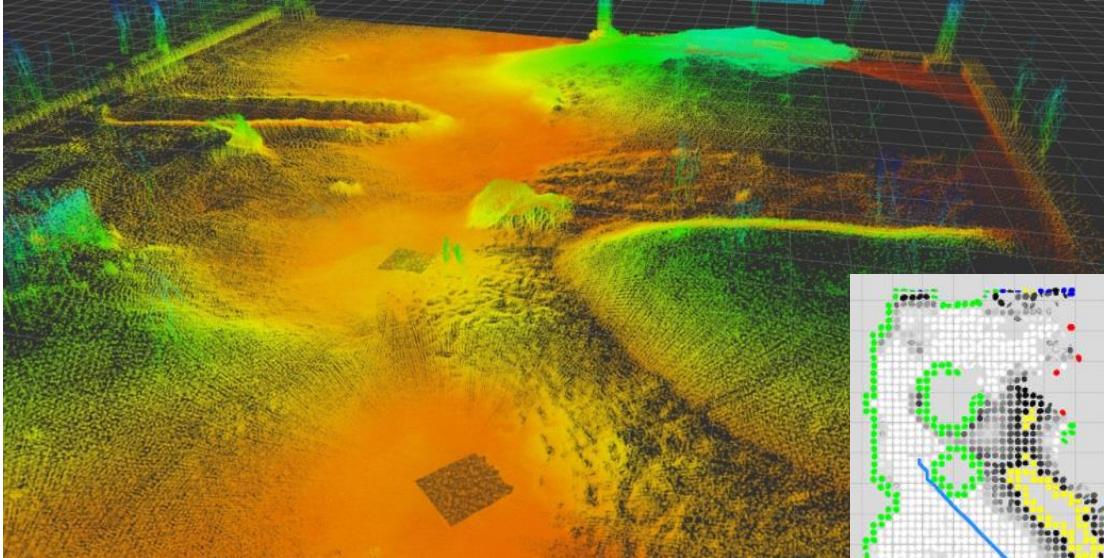
A^*



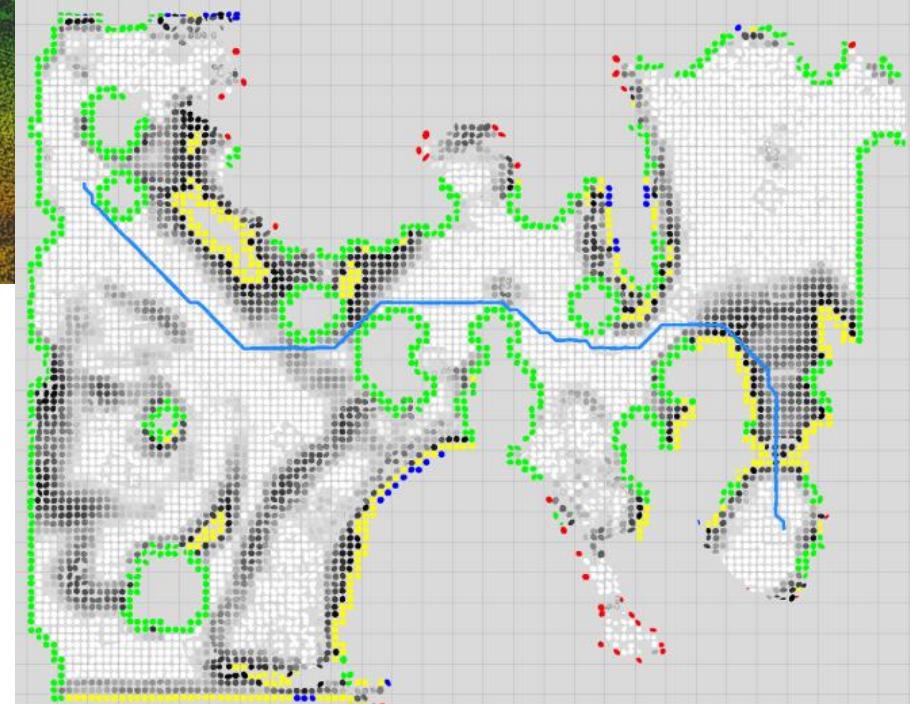
Rollout

Pfadplanung in unebenem Gelände

- 3D-Karte aus mehreren registrierten 3D-Laserscannerscans

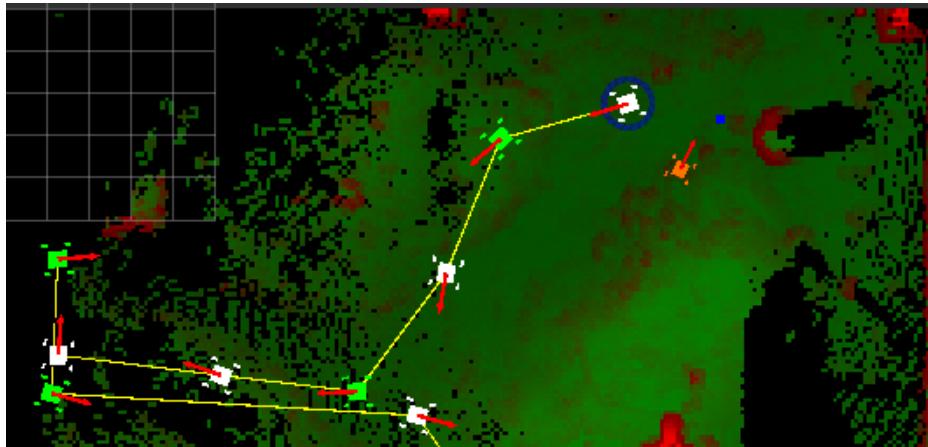
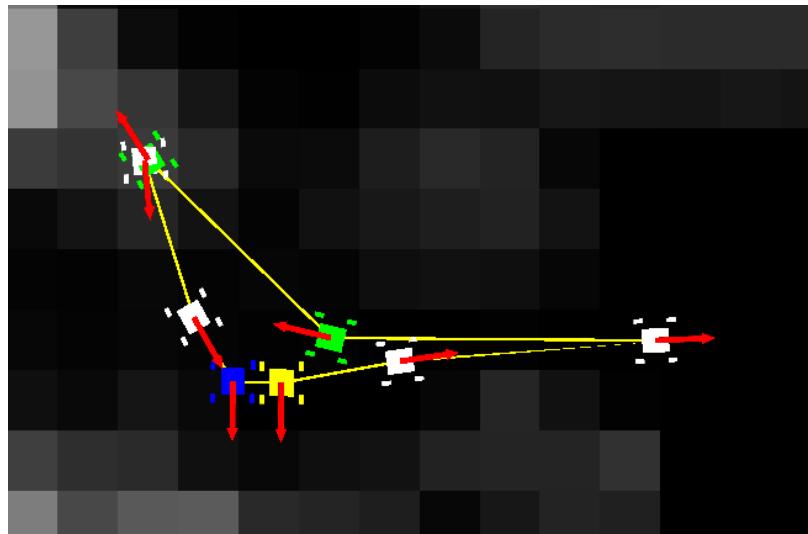


- Zellkosten abgeleitet von lokalen Geländeeigenschaften



Missionsplanung

- Spezifikation von Wegpunkten relativ zu initialer Karte
- Scan- und Objektsuchposen
- Interaktive Verfeinerung bei Checkpunkten

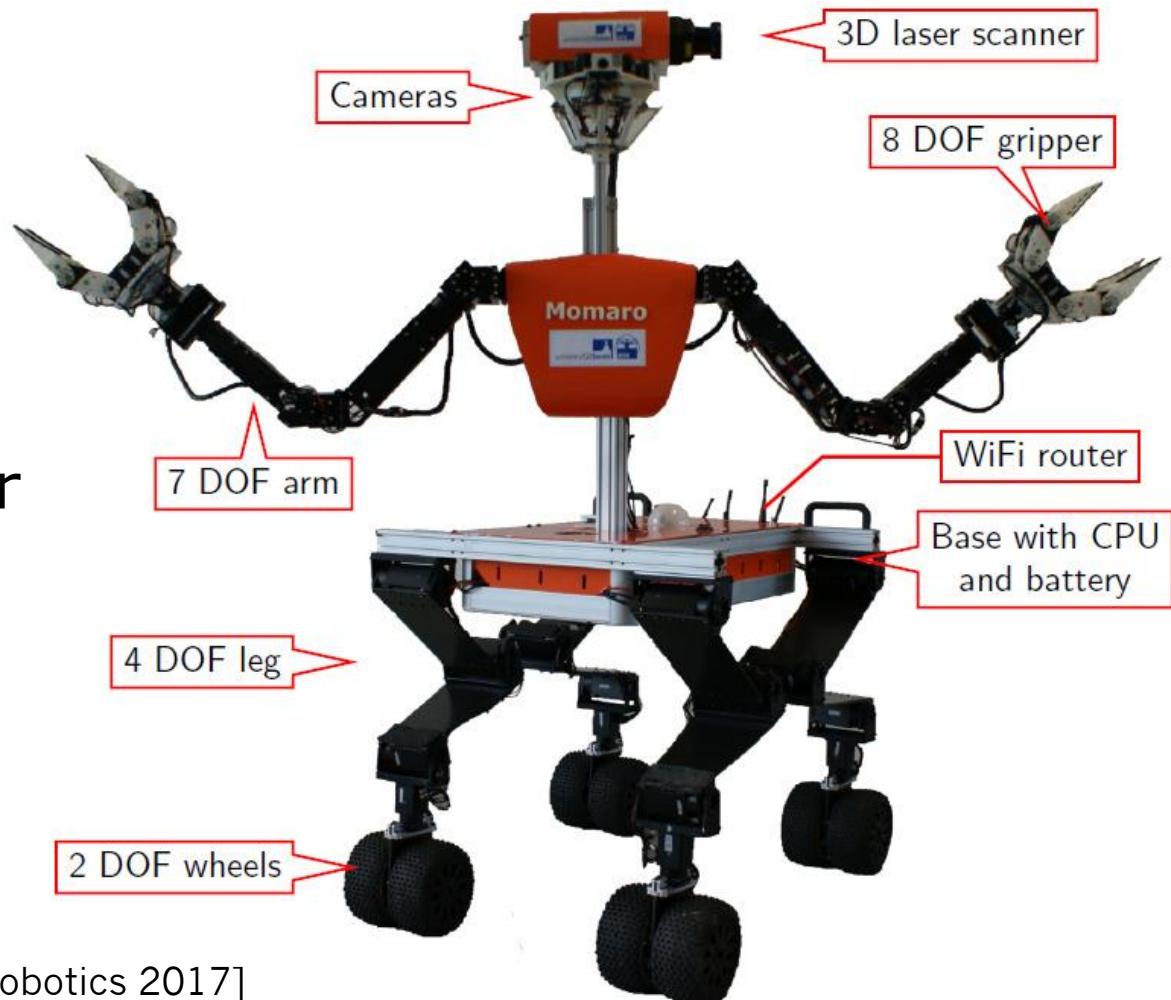


- Posen sind Ziele für allozentrische Pfadplanung
- Abarbeitung braucht Lokalisierung!

Mobiler Manipulations-Roboter Momaro



- Vier nachgiebige Beine mit lenkbaren Radpaaren
- Menschenähnlicher Oberkörper
- Sensorkopf
 - 3D-Laserscanner
 - IMU, Kameras



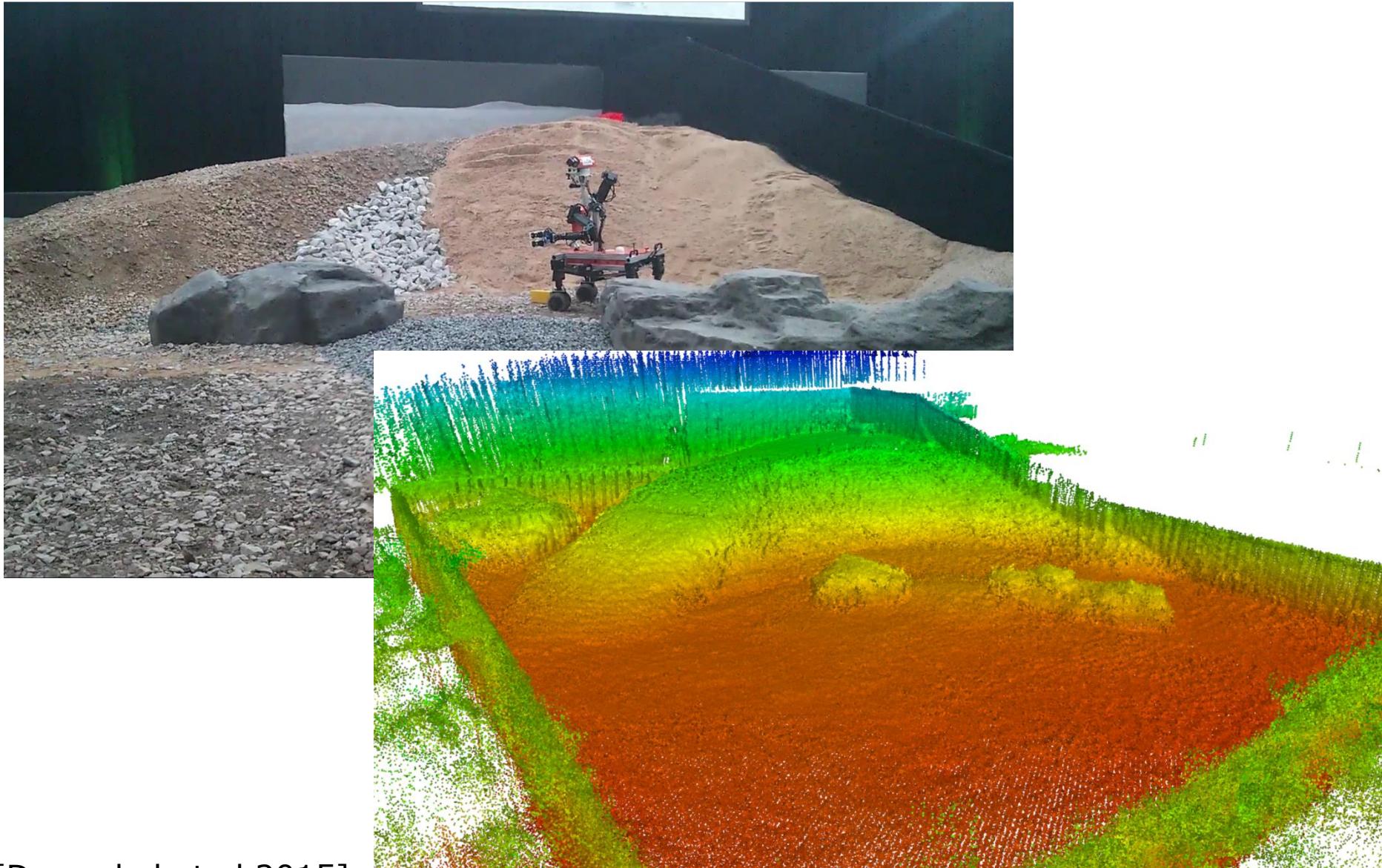
[Schwarz et al. Journal of Field Robotics 2017]

DLR SpaceBot Camp 2015



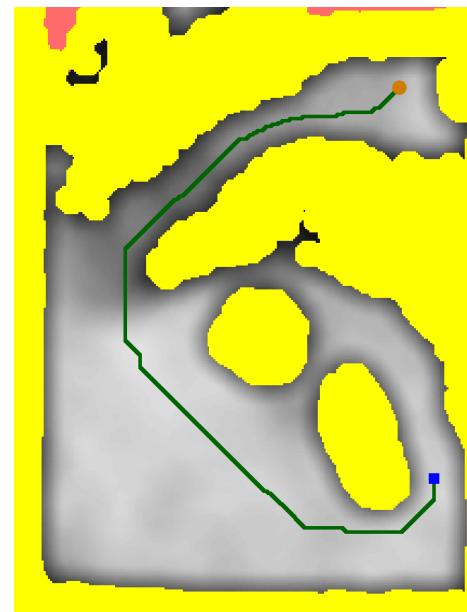
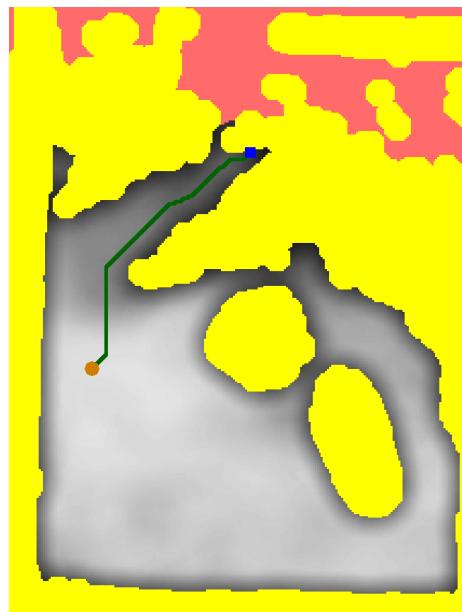
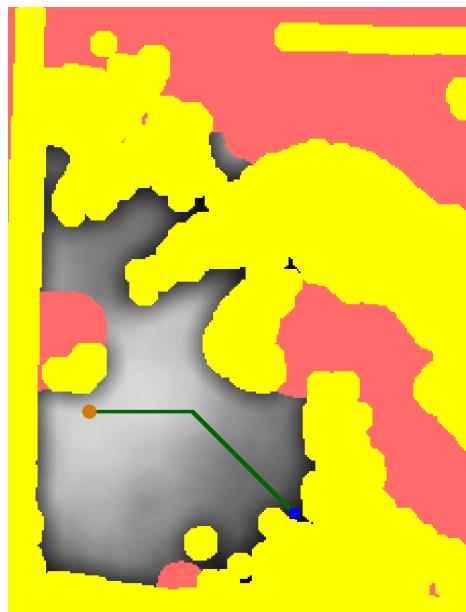
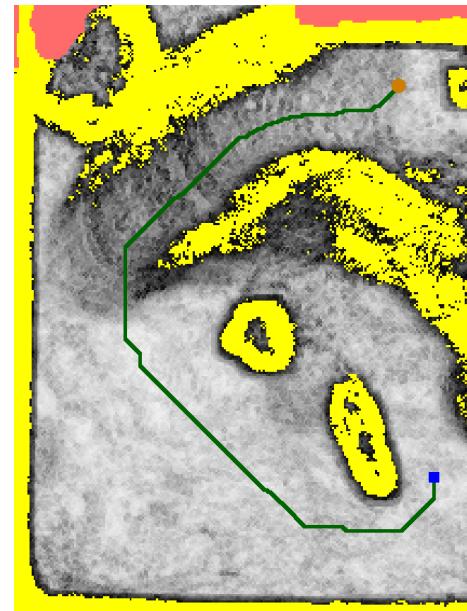
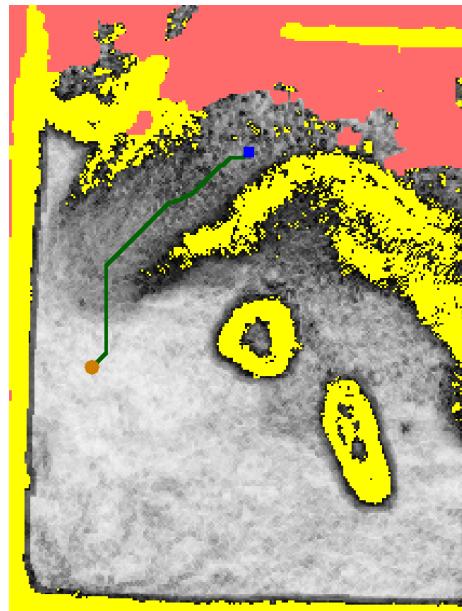
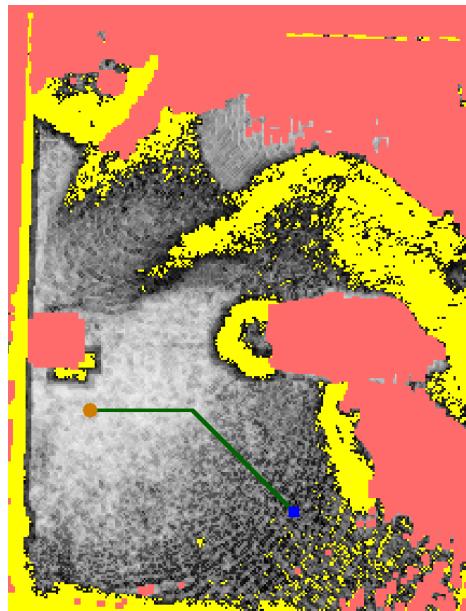
8X

3D-Laser-basierte Karte

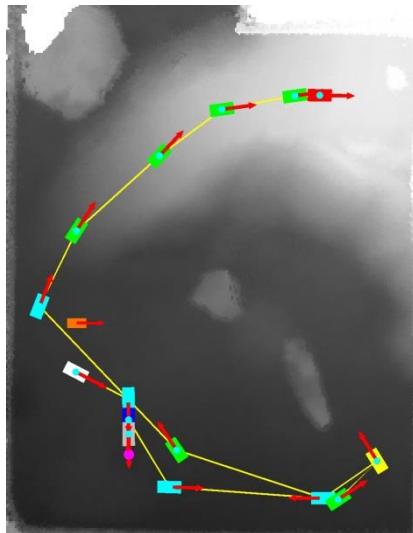


[Droeschel et al 2015]

Exploration und Pfadplanung



Überwacht Autonome Missionsausführung



Missionsplan



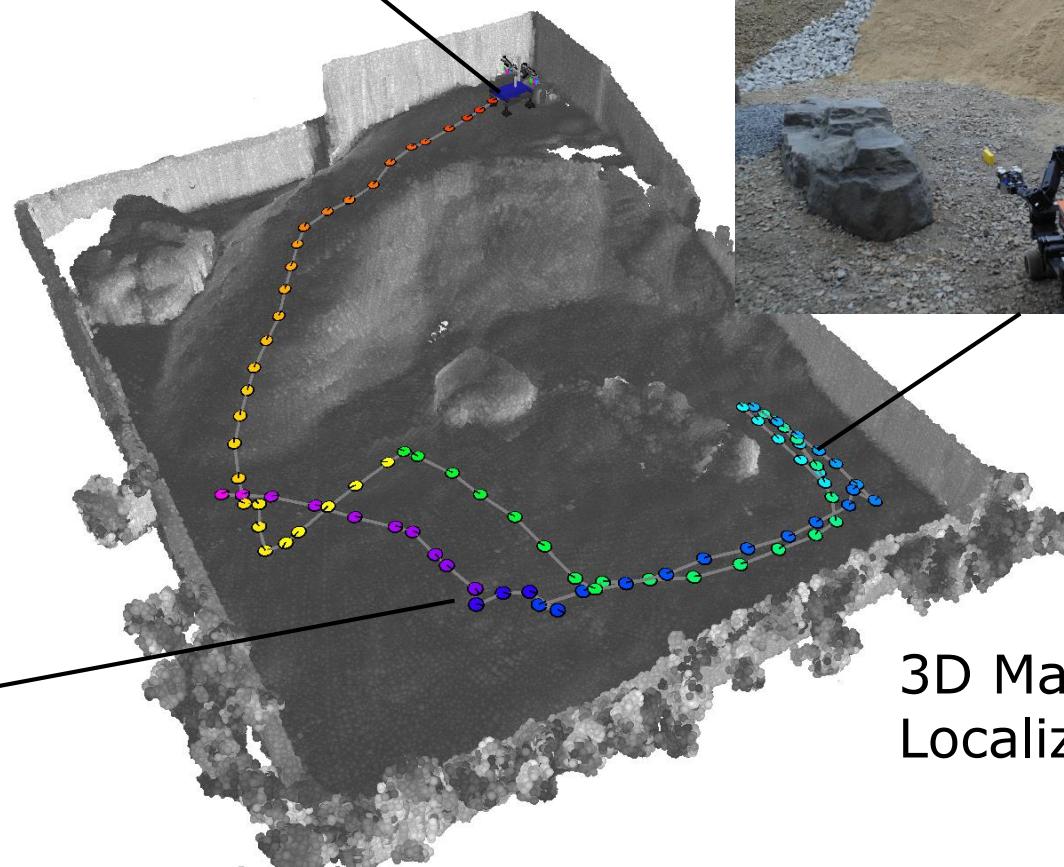
Assembly



Collecting battery



Taking soil sample

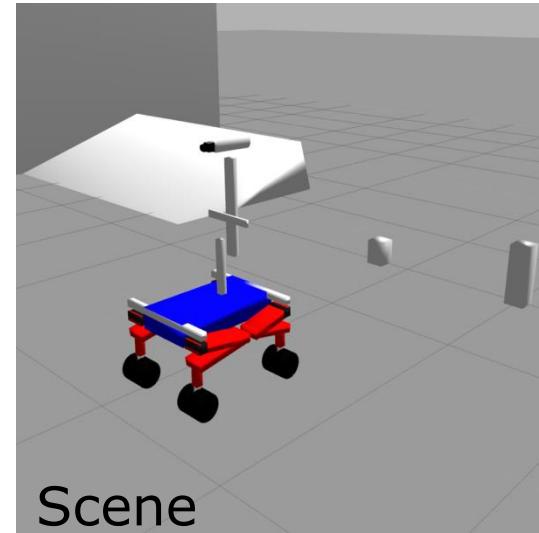


3D Map +
Localization

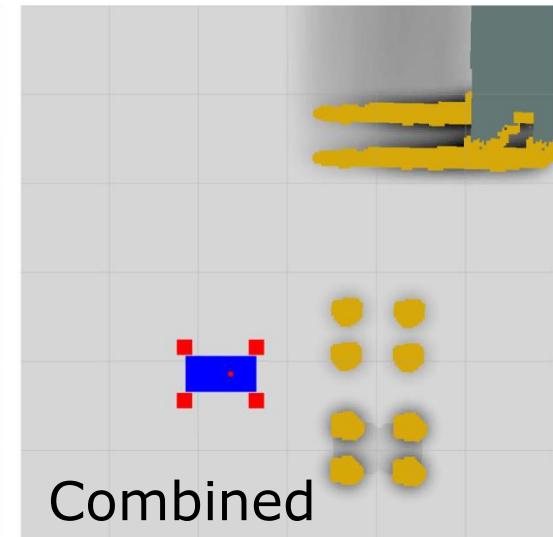
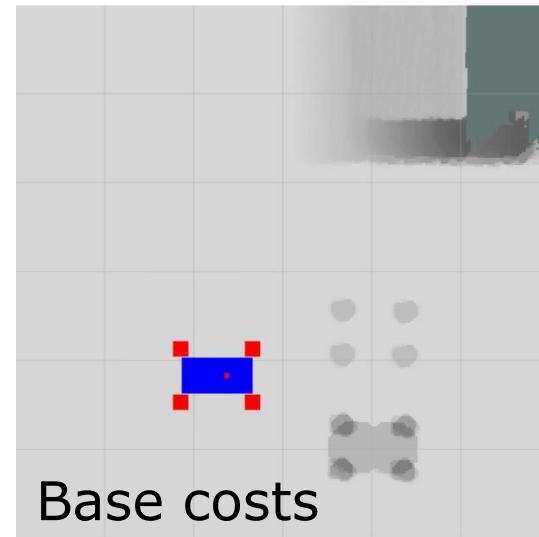
Berücksichtigung der Roboterform

- Kosten für individuelle Radpaare abgeleitet aus lokalen Höhenunterschieden
- Kosten für die Basis
- Nicht-lineare Kombination ergibt 3D (x, y, θ) Kosten

[Klamt and Behnke, 2017]

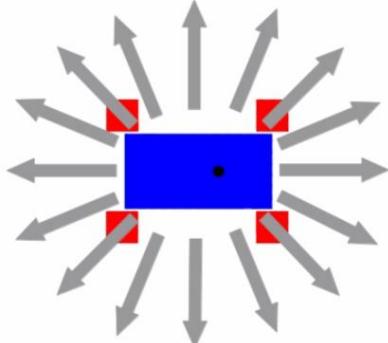


Scene

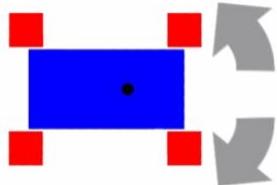


3D-Navigationsplanung (x, y, θ): A*

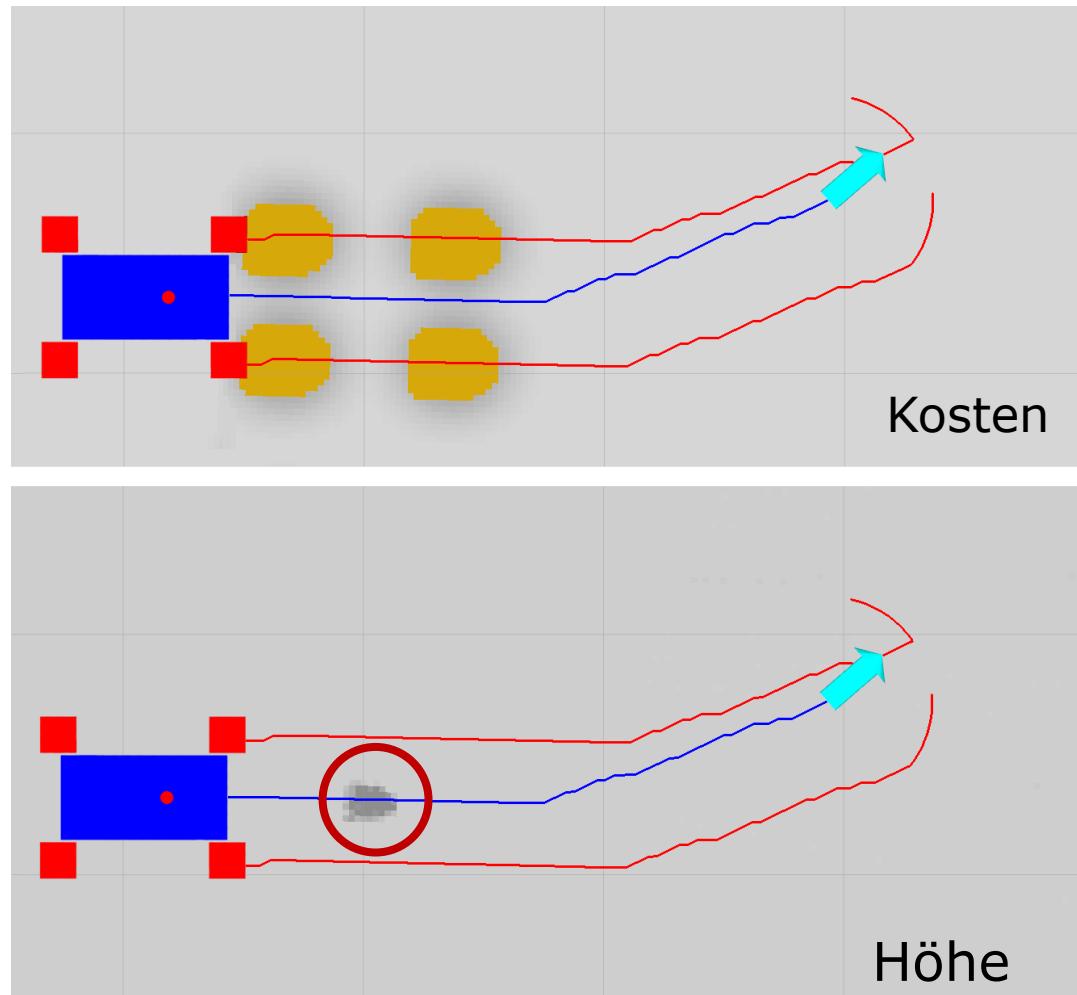
- 16 Fahrtrichtungen



- Orientierungsänderungen



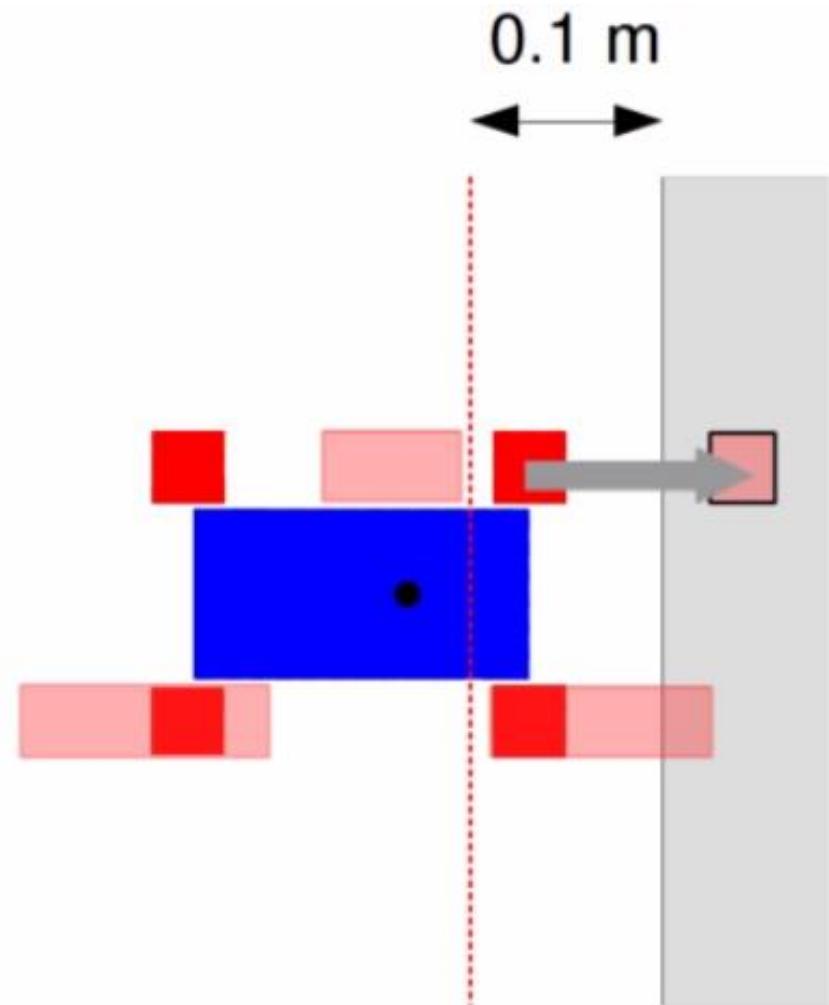
=> **Hindernis zwischen den Rädern**



[Klamt and Behnke, IROS 2017]

Überwindung von Höhenunterschieden durch Schritte

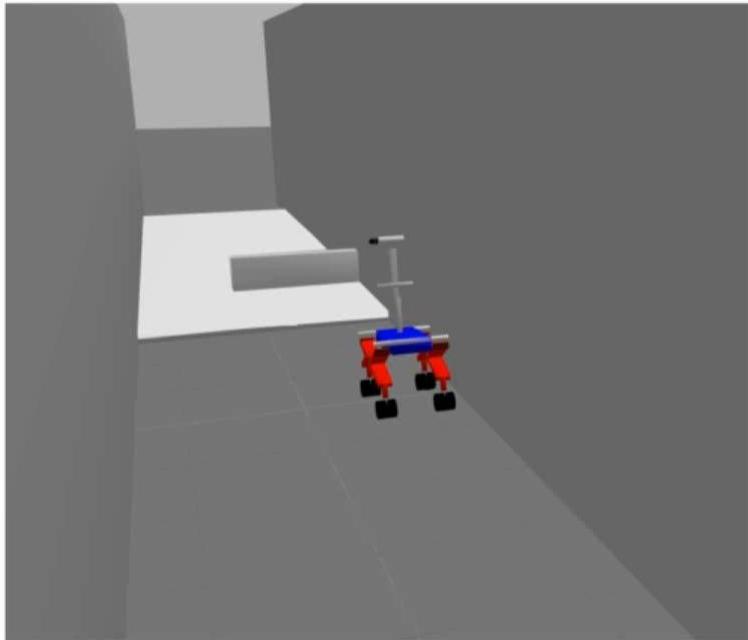
- Wenn nicht befahrbarer Höhenunterschied vor einem Radpaar
- Aufsetzpunkt nach dem Höhenunterschied muss befahrbar sein
- Die drei anderen Beine müssen fahrbar sein



[Klamt and Behnke, IROS 2017]

Hybrider Lokomotionsplan kombiniert Fahren und Schritte

Path Planning Example



Scenario: Momaro has to step up a height difference and manoeuvre around a small wall.

[Klamt and Behnke, 2017]

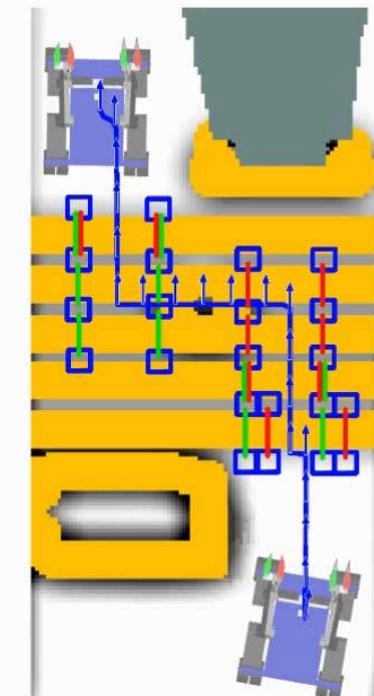
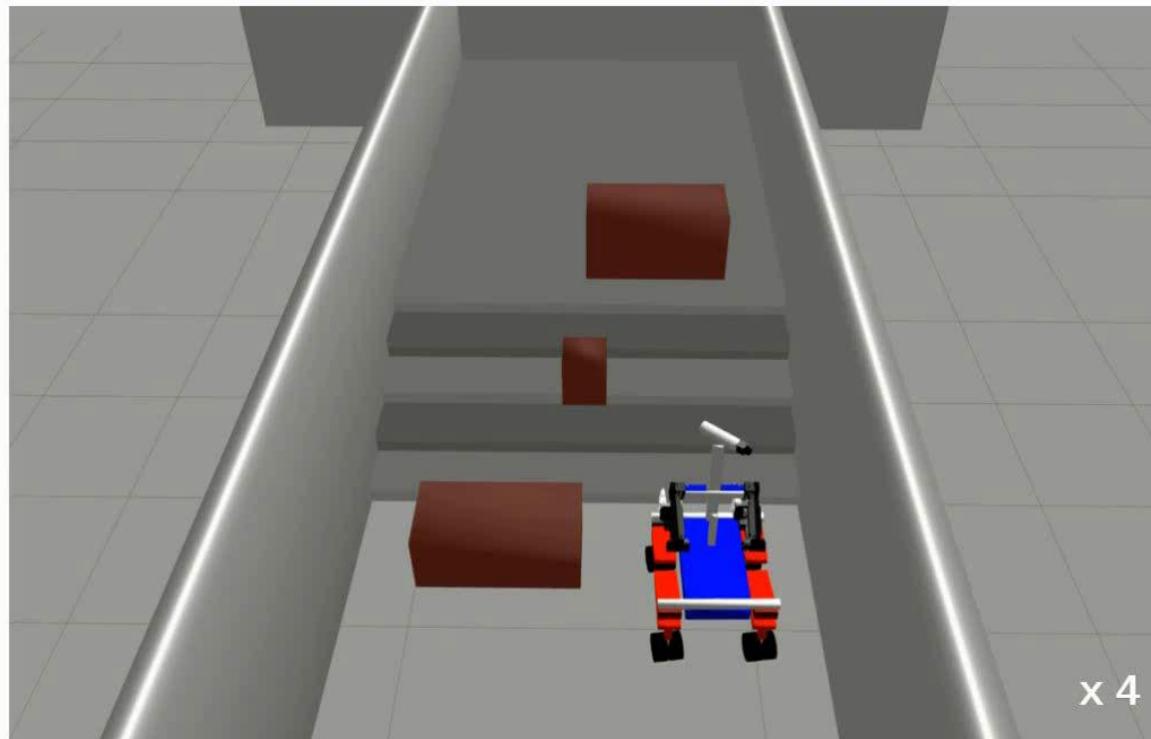
Realisierung der Schritte

Expanding Abstract Steps to Detailed Motion Sequences



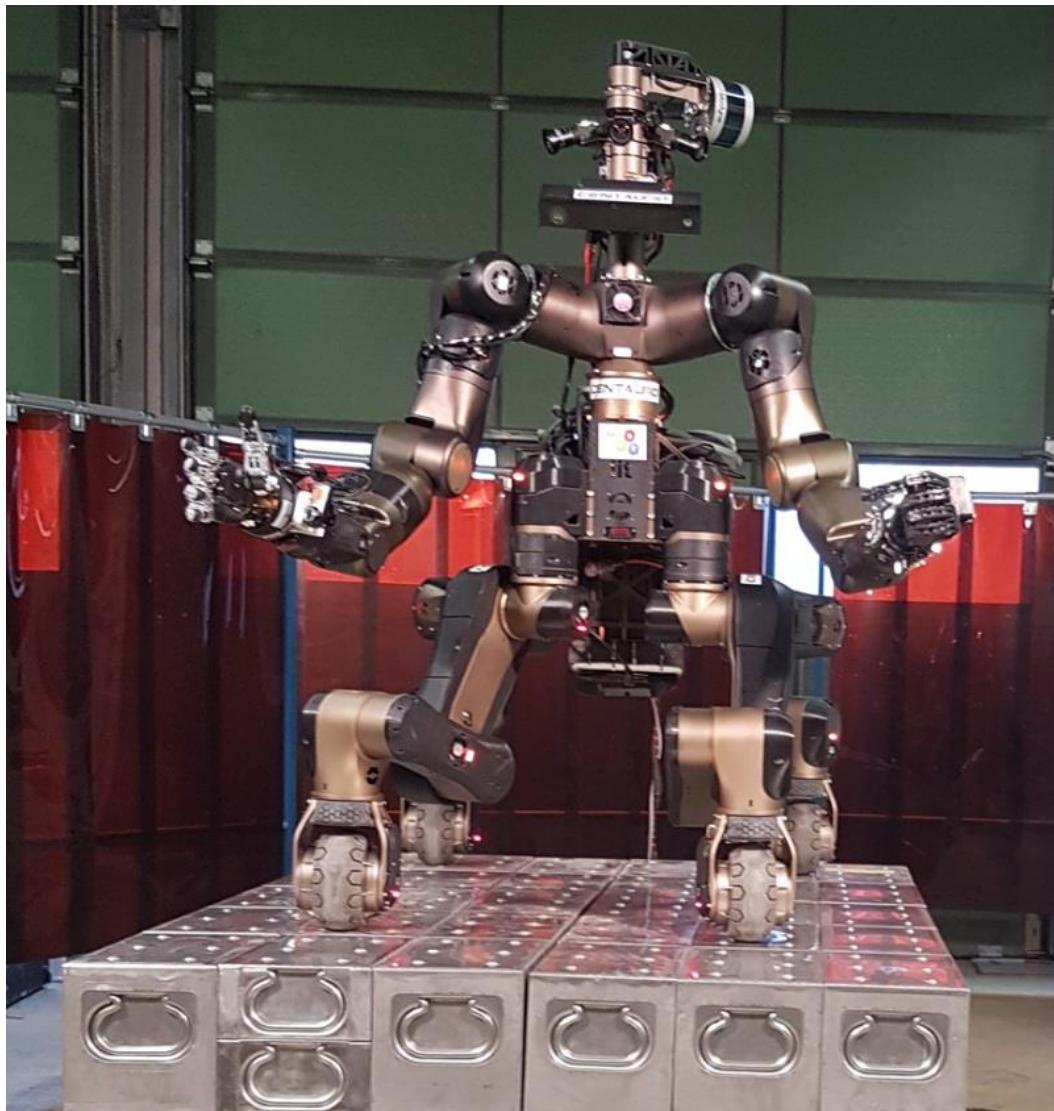
Treppe mit zusätzlichen Hindernissen

Planning for Challenging Scenarios



Centauro-Roboter

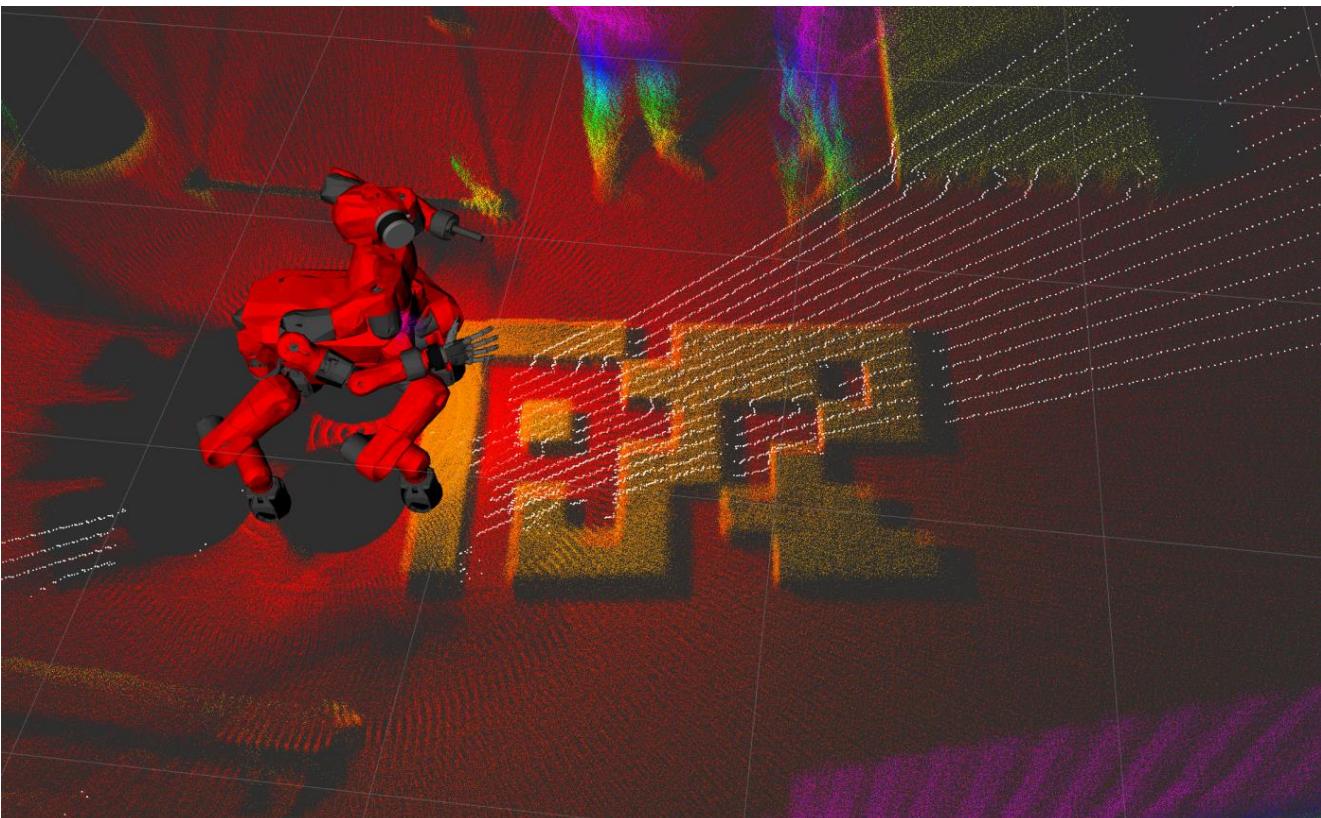
CENTAURO



- Seriell-elastische Aktuatoren
- 42 große Gelenke
- Schunk-Hand
- 3D-Laserscanner
- RGB-D-Kamera
- Farbkameras
- Zwei GPU-PCs

[Tsagarakis et al.,
IIT 2017]

3D-Kartierung und Lokalisierung

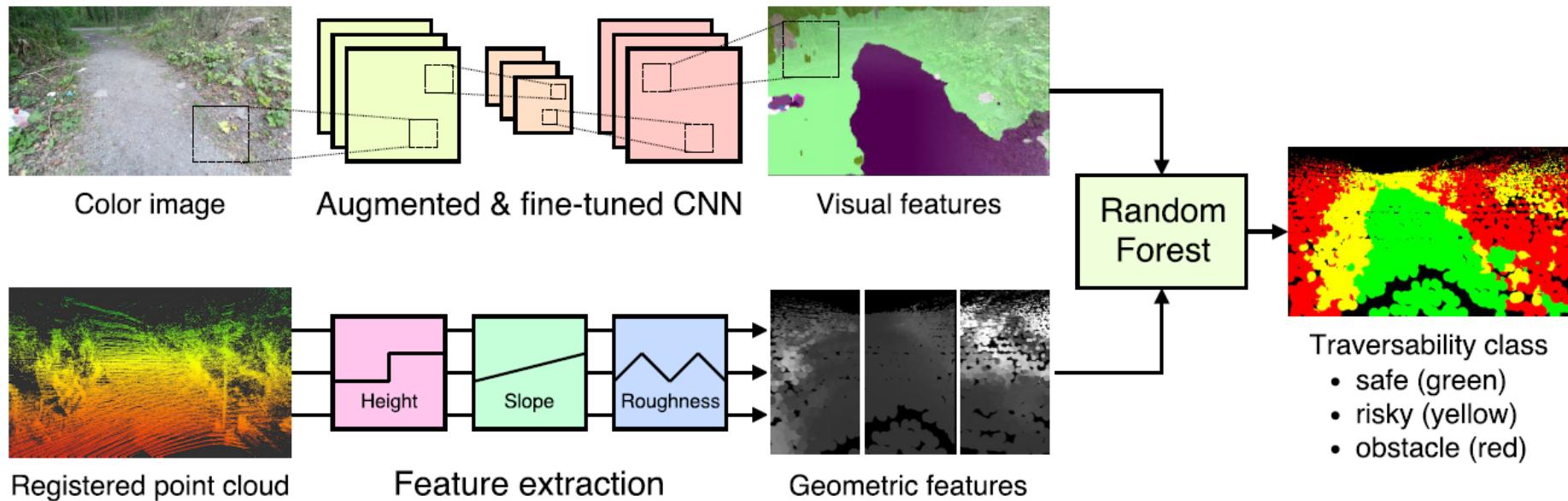


Laufen über unebenes Gelände



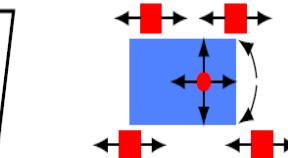
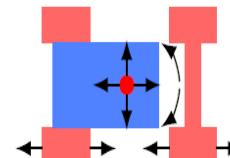
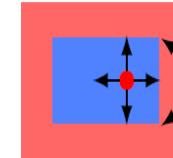
8

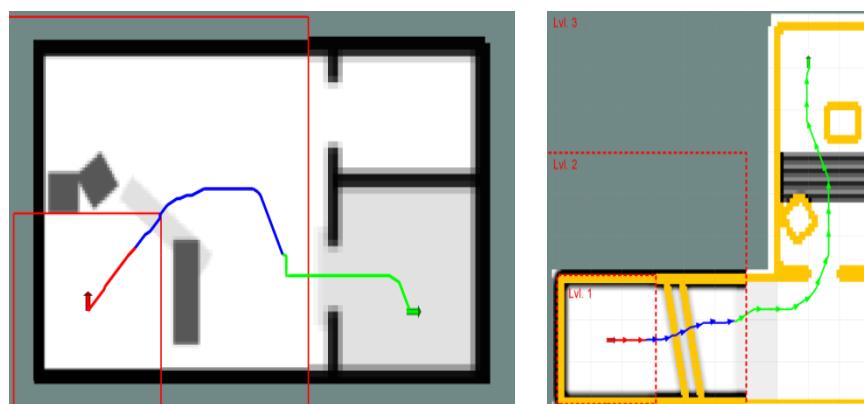
Terrain-Klassifikation



[Schilling et al., IROS 2017]

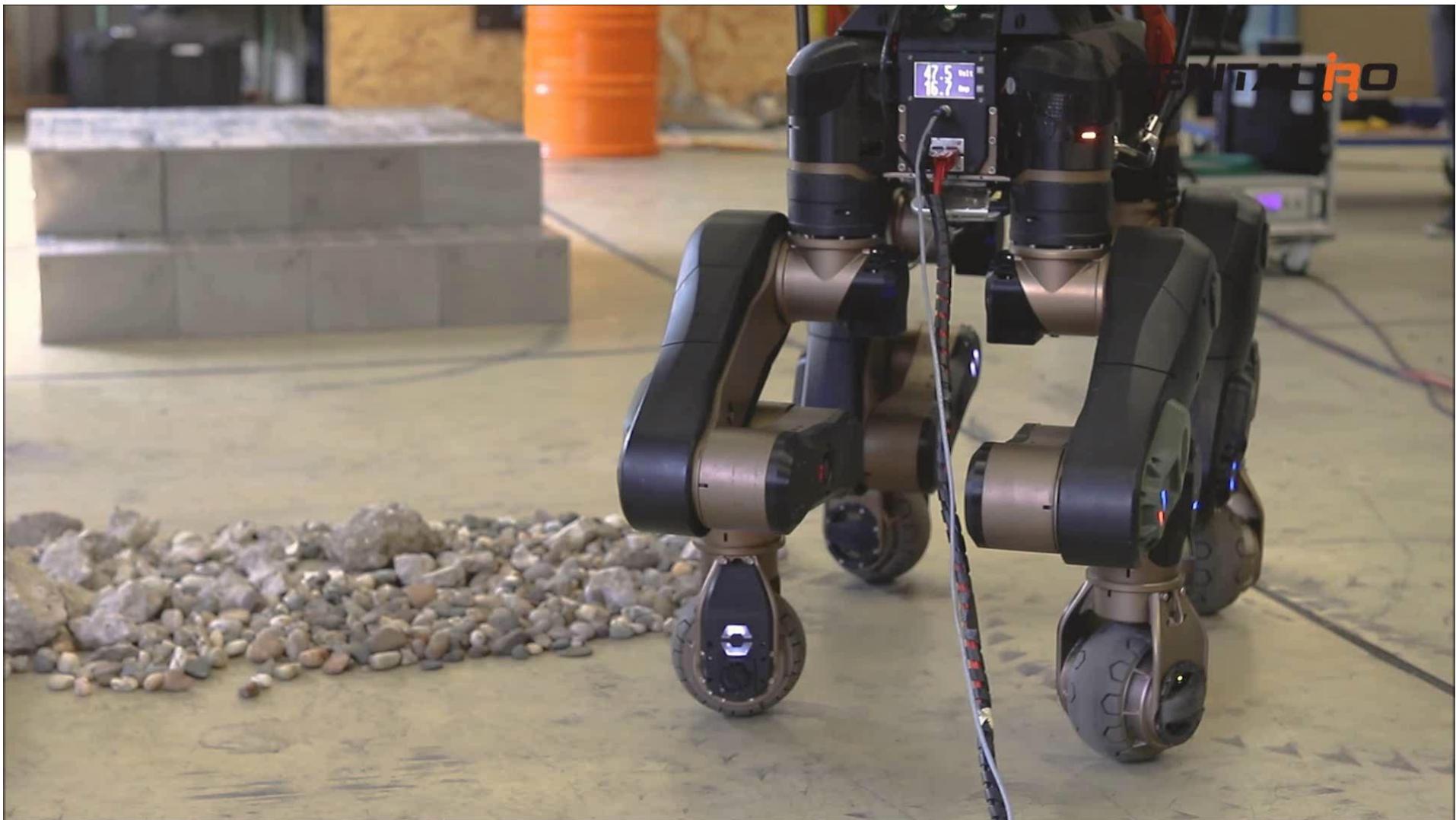
Planen auf mehreren Abstraktionsstufen

Level	Map Resolution	Map Features	Robot Representation	Action Semantics
1	• 2.5 cm • 64 orient.	• Height		• Individual Foot Actions
2	• 5.0 cm • 32 orient.	• Height • Height Difference		• Foot Pair Actions
3	• 10 cm • 16 orient.	• Height • Height Difference • Terrain Class		• Whole Robot Actions



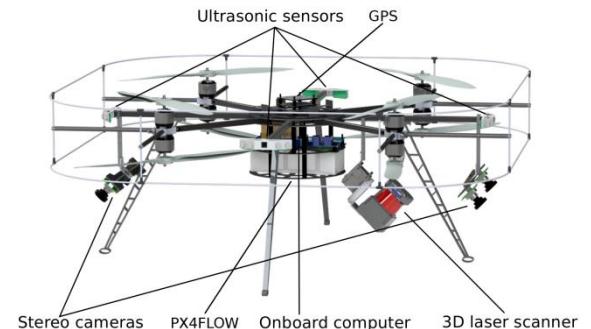
[Klamt and Behnke,
IROS 2017, ICRA 2018]

Evaluation @ KHG: Hybride Fortbewegung

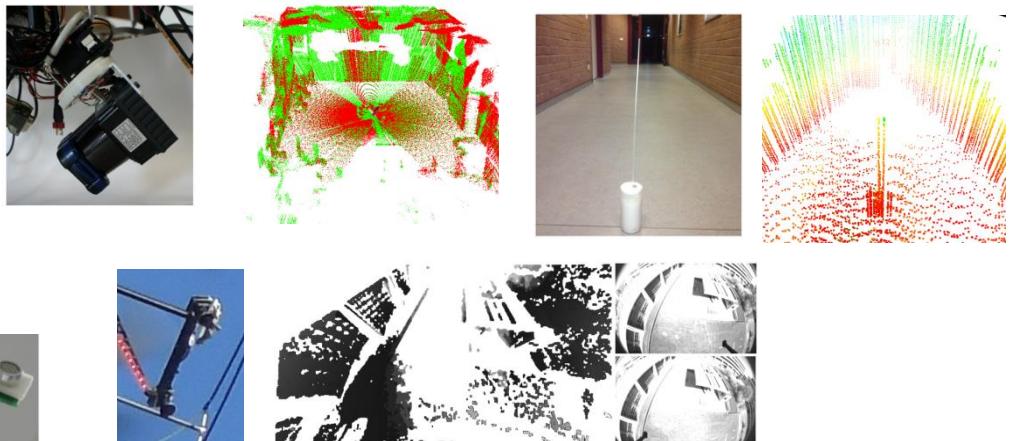


Autonomer Flug in Hindernisnähe

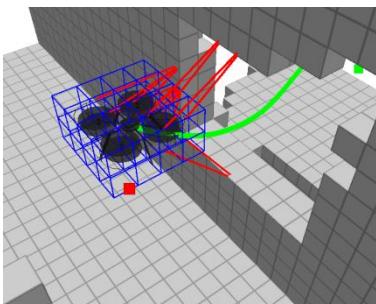
- Flugroboter mit viel Sensorik und Rechenleistung
- Multimodale Hindernisdetektion
 - 3D-Laserscanner



- Stereo-Kameras
- Ultraschall

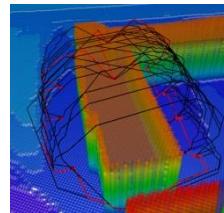


- Lokale Hindernisvermeidung

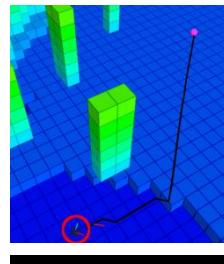


Navigationsarchitektur

■ Geschichtete Planung: Von abstrakt zu konkret



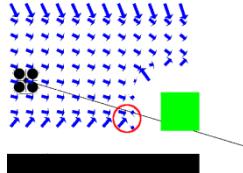
Missionsplanung



Allozentrische Planung

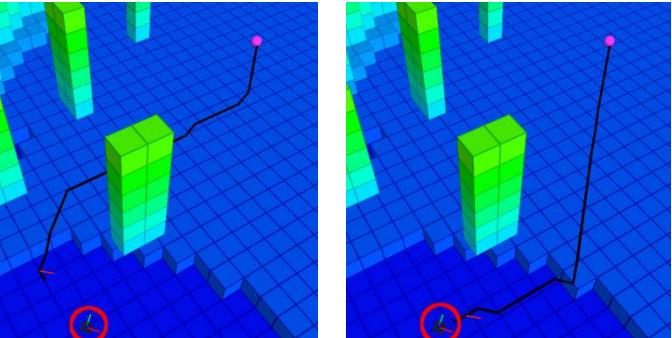
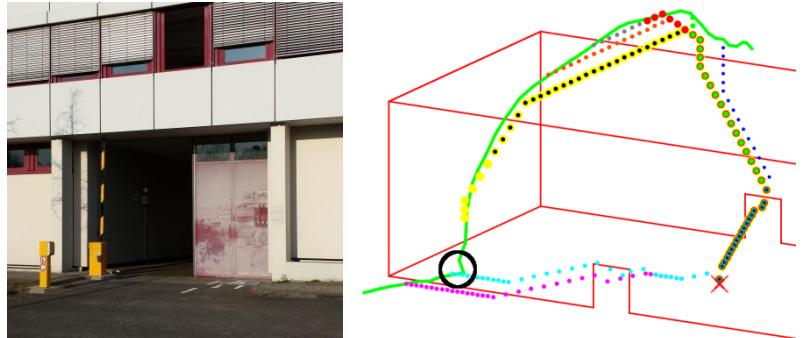
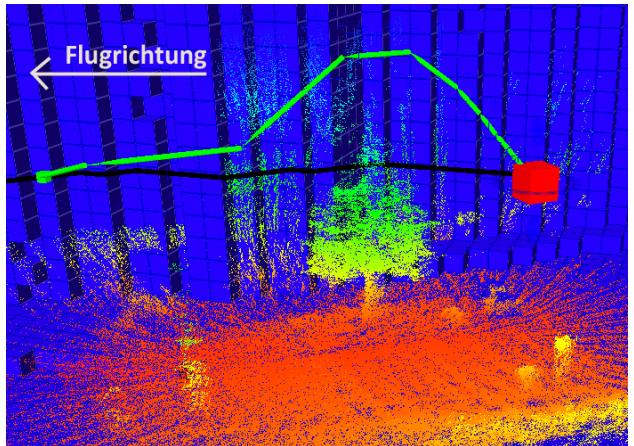


Egozentrische Planung



Hindernisvermeidung

Navigationsplanung

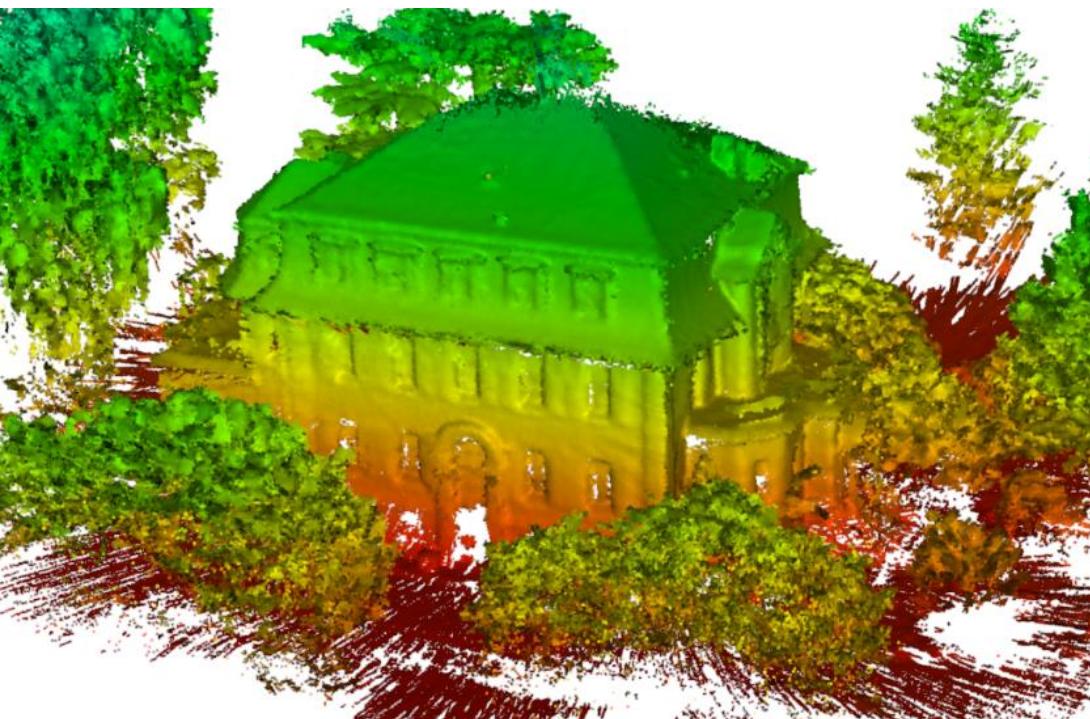
- Kontinuierliche allozentrische Planung
 - A*-Planer in Voxel-Gitter findet kostenoptimale Pfade
- Reaktion auf Positionsabweichung Reaktion auf Belegtheitsänderung
- 
- 
- Lokale Multiresolutionsplanung
 - A*-Planer in lokaler Multi-resolutionskarte
 - Neuplanung mit hoher Rate
 - Umfliegen lokal wahrnommener Hindernisse
- 

Autonomer Flug in Hindernisnähe

Mapping on Demand
Autonomous Flight to Planned View Poses

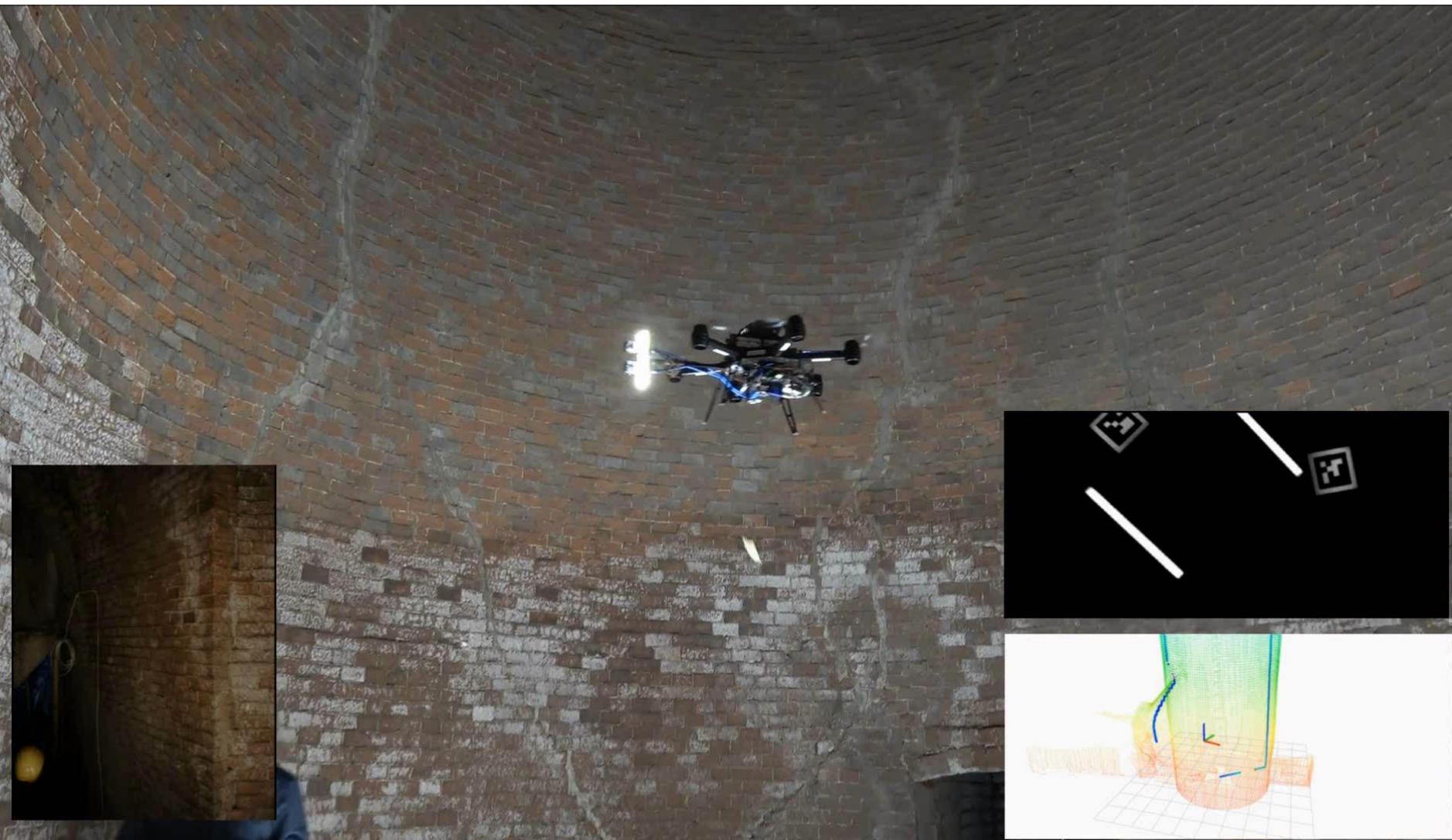
3D-Kartierung

- Registrierung von Laser-Distanzmessungen



[Droeschel et al. JFR 2016]

European Robotics Challenge 3: ChimneySpector



DJI Matrice 600 mit Velodyne Puck Lidar



Autonome Navigation in Innenräumen



Fully Autonomous indoor flight without external tracking.

InventAIRy: Detektierte Etiketten im Regal



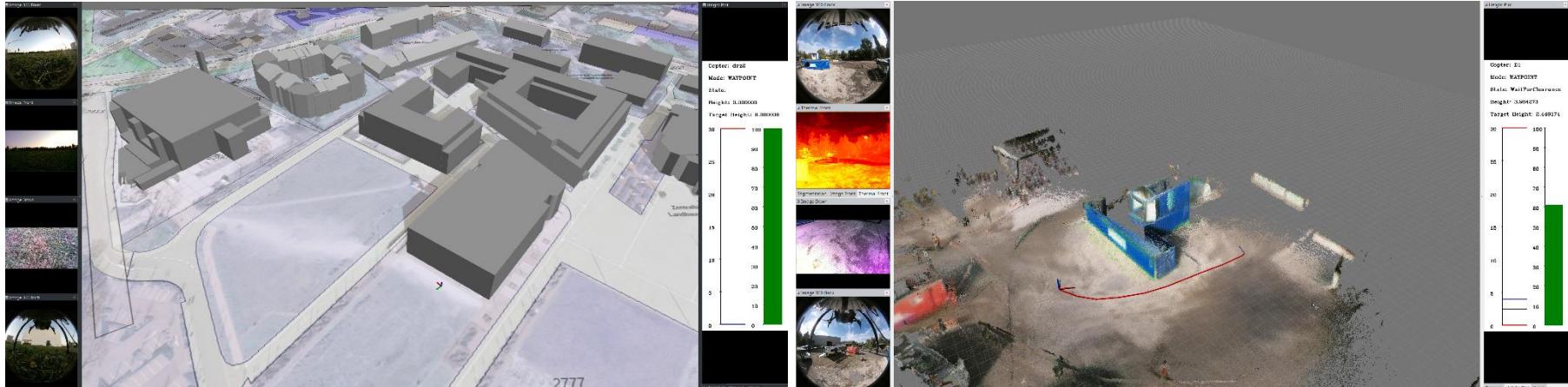
Deutsches Rettungsrobotik-Zentrum

- Basis: DJI Matrice M210 v2
- Intel NUC 11th Gen zur Onboard-Verarbeitung
- 20 Hz 3D-LiDAR Ouster OS0-128
- 360° Kamera Insta360 Air
- FLIR Boson Thermalkamera
- 2x Realsense D455 RGB-D Kamera



Operator-Interface

- Anzeige von CityGML-Modellen, Satellitenbildern und Straßenkarten aus dem Lagebildsystem
- Intuitive Steuerung mehrerer Roboter an der Bodenstation mit Gamepad und Maus
- Visuelle und haptische Rückmeldung des aktuellen Zustandes und der nächsten geplanten Aktionen für wechselbare Steuerungsmodi mit unterschiedlichen Autonomiegraden



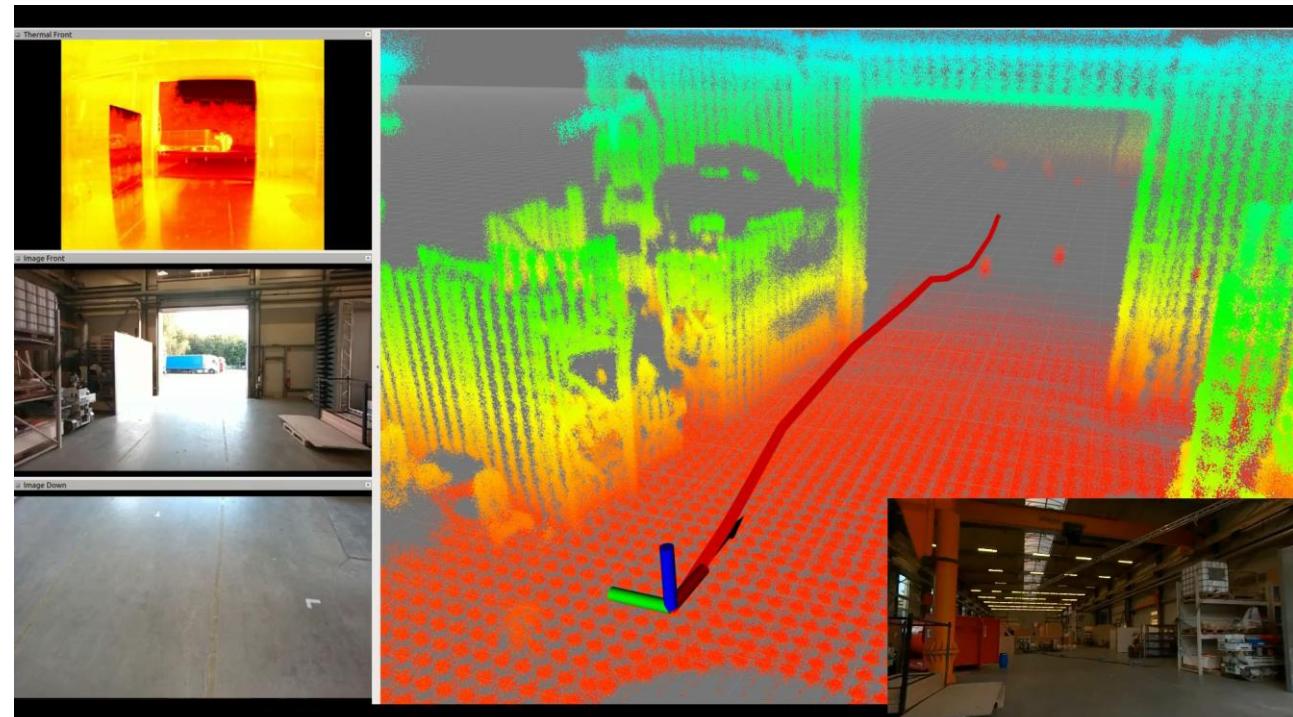
Hindernisvermeidung

- Aktuelle Zielpose wird per Gamepad verschoben und in Echtzeit angeflogen
- Aktive Hindernisvermeidung passt Zielposen an, sodass
 - zu dichte Annäherung an Hindernisse vermieden wird
 - Flugrichtung innerhalb des Sensorsichtfeldes liegt
- Haptische Rückmeldung bei Eingreifen der Hindernisvermeidung



Hindernisvermeidung

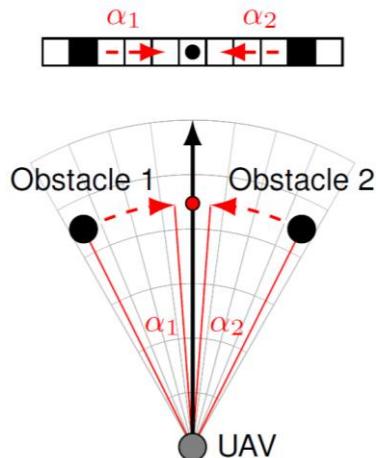
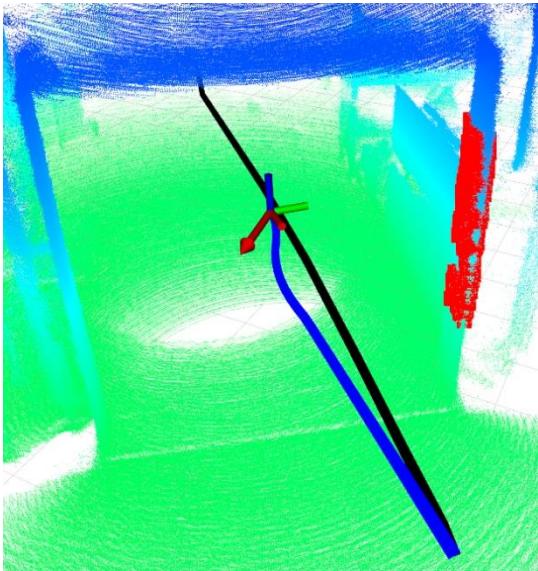
- Aktuelle Zielpose wird per Gamepad verschoben und in Echtzeit angeflogen
- Aktive Hindernisvermeidung passt Zielposen an, sodass
 - zu dichte Annäherung an Hindernisse vermieden wird
 - Flugrichtung innerhalb des Sensorsichtfeldes liegt
- Haptische Rückmeldung bei Eingreifen der Hindernisvermeidung



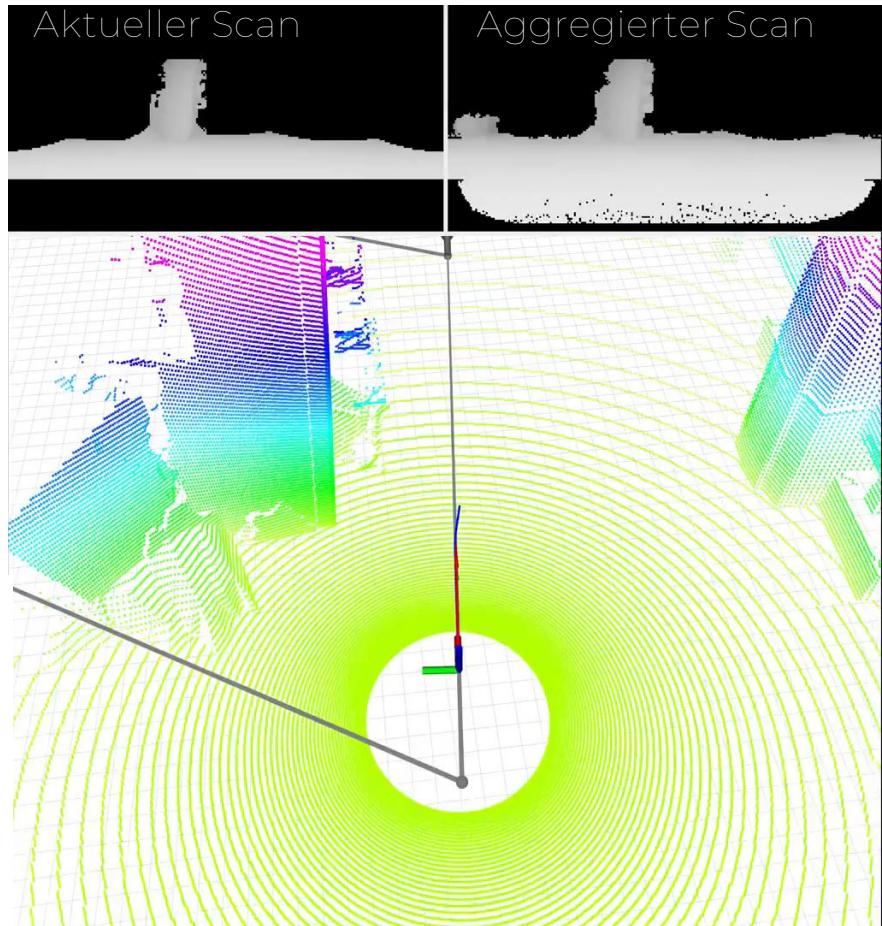
[Beul et al. SSRR 2020]

Hindernisvermeidung

- LiDAR-Scans zu Entfernungsbild aggregieren
- Anpassen der Flugrichtung mit Potentialfeld-Methode
- Vorhersage von Flugbahn und Entfernungsbild
- Skalierung der Geschwindigkeit basierend auf Zeit bis Berührung



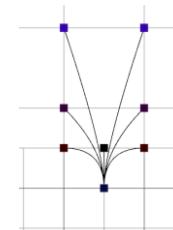
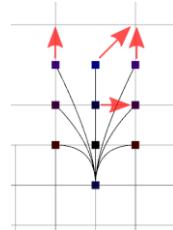
Anpassen der Flugrichtung



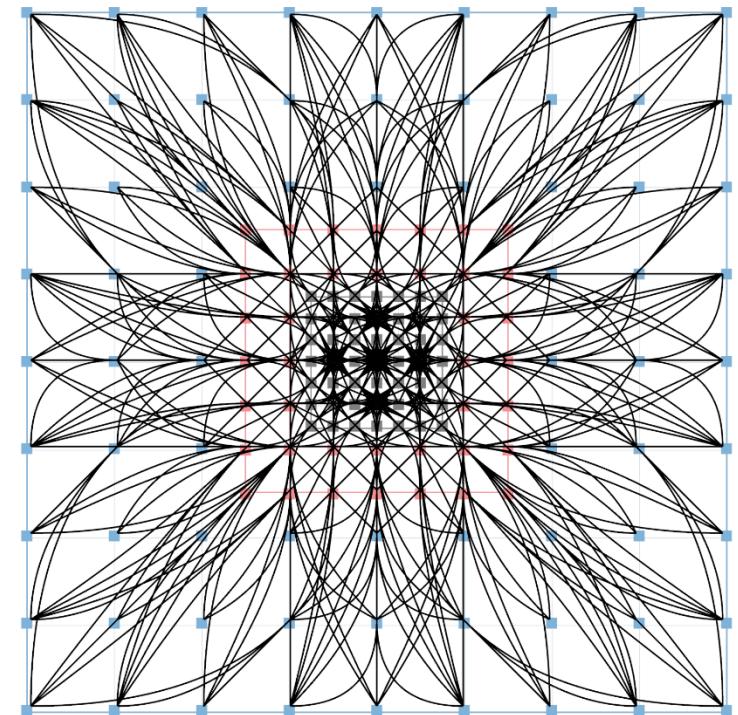
[Schleich and Behnke, IROS 2022]

Dynamische Trajektorienplanung

- Suchbasierte Planung in hochdimensionalem Zustandsgraphen
- Berücksichtigung von Geschwindigkeit und Beschleunigung
- Anpassung der Bewegungsprimitive an lokales multiresolutionales Gitter

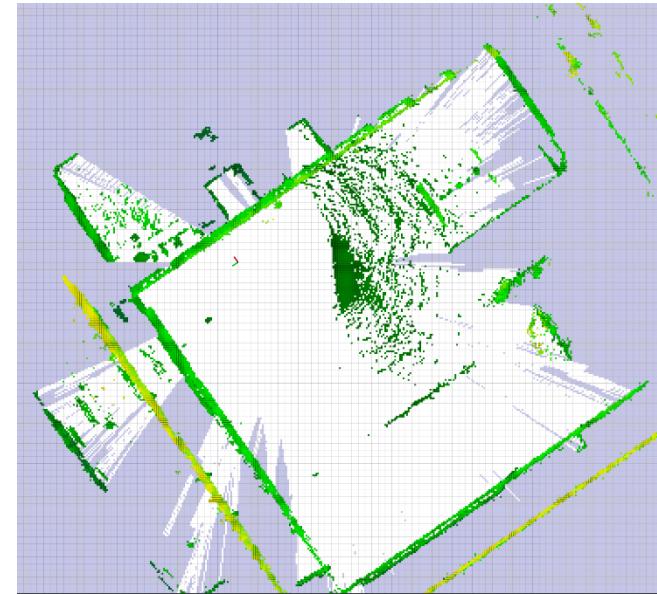
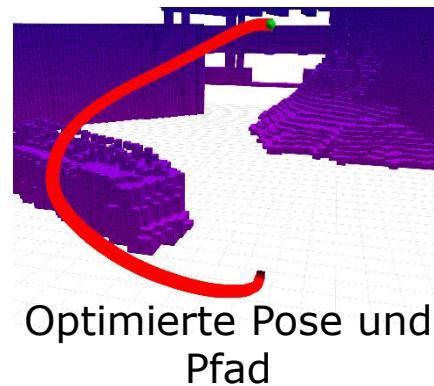
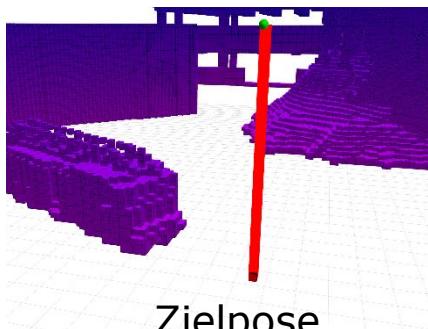


- Optimierung von Flugzeit und Kontrollkosten
- Neuplanung mit ~ 1 Hz



Planung mit Sichtbarkeitsbeschränkung

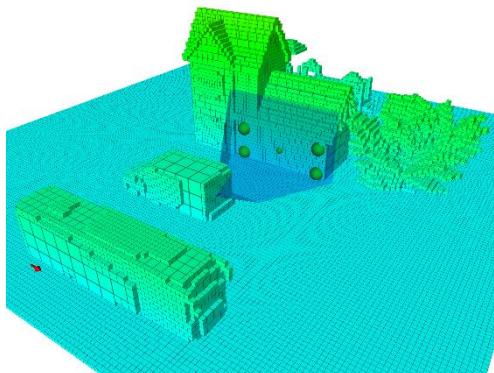
- Zusätzliche Kosten für Flug durch unkartierte Bereiche
- Berücksichtigung des Sensorsichtfeldes bei der Planung:
 - Steigratenbeschränkung durch Kopplung von vertikaler und horizontaler Fluggeschwindigkeit
 - Beschränkung auf Flug in Vorwärtsrichtung mit begrenzter Rotationsgeschwindigkeit



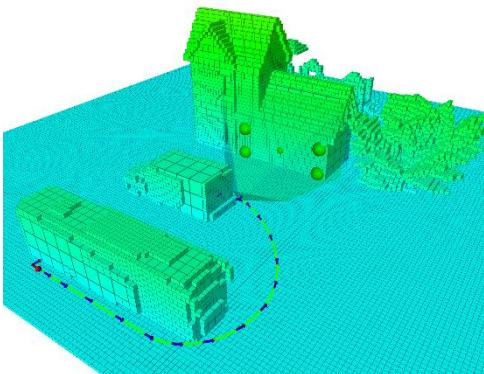
Hinderniskarte

Planung von Beobachtungsposen

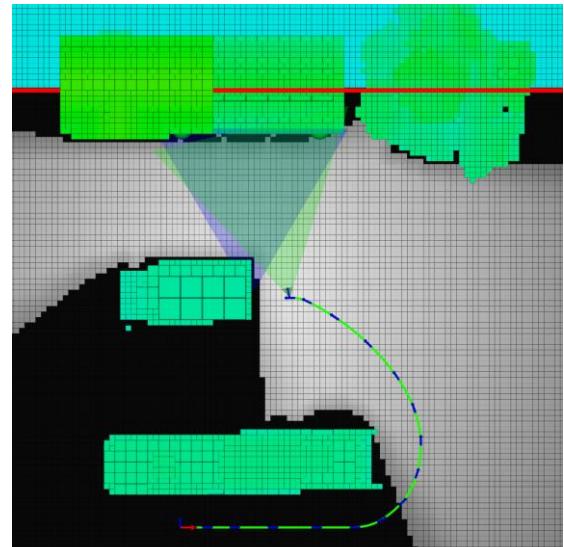
- Ableitung von Beobachtungsposen zu Zielobjekten
- Optimierung der Posen hinsichtlich Sichtbarkeit, Hindernissabstand und Erreichbarkeit



Zielbeobachtungs-
pose



Optimierte Pose und
Pfad



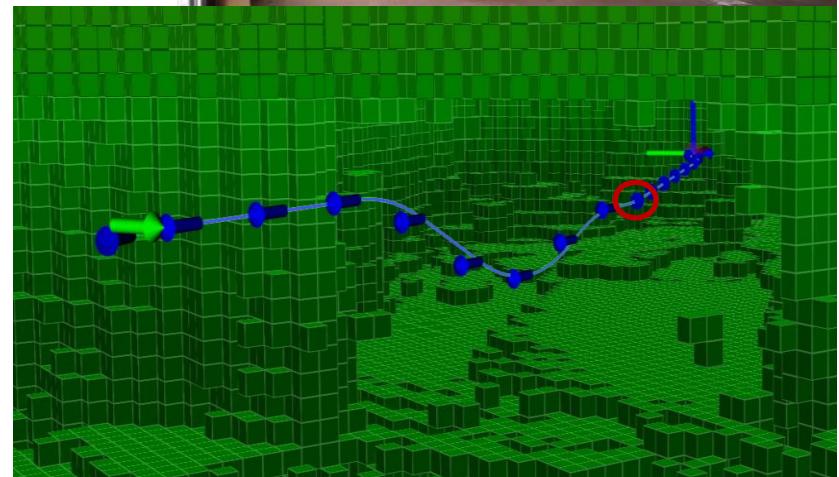
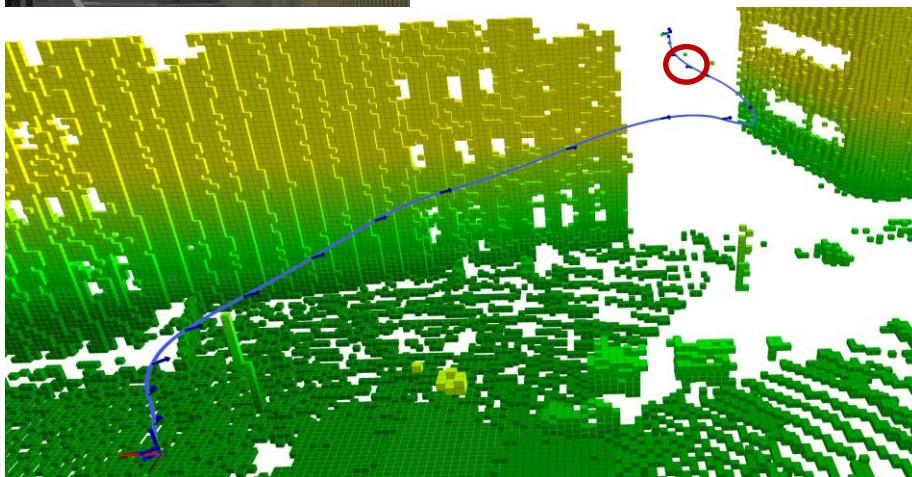
Sichtbarkeit

Wegpunkt-Navigation



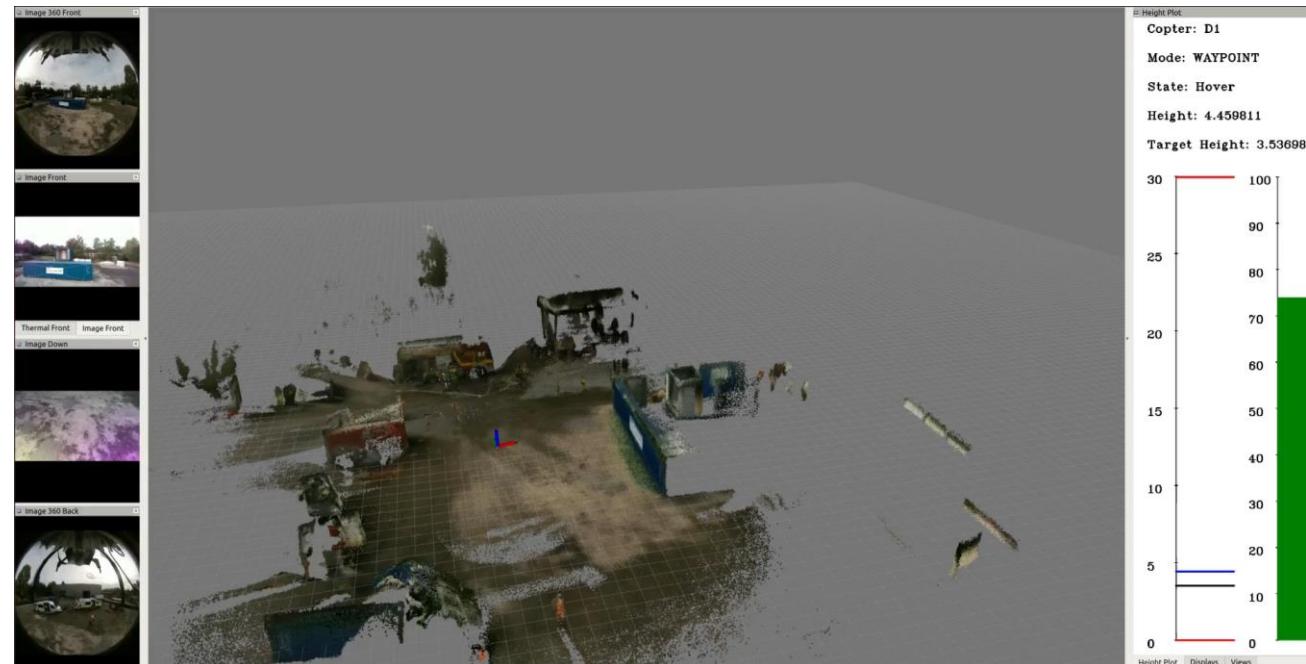
Campus
Poppelsdorf

Desaster
Szenario

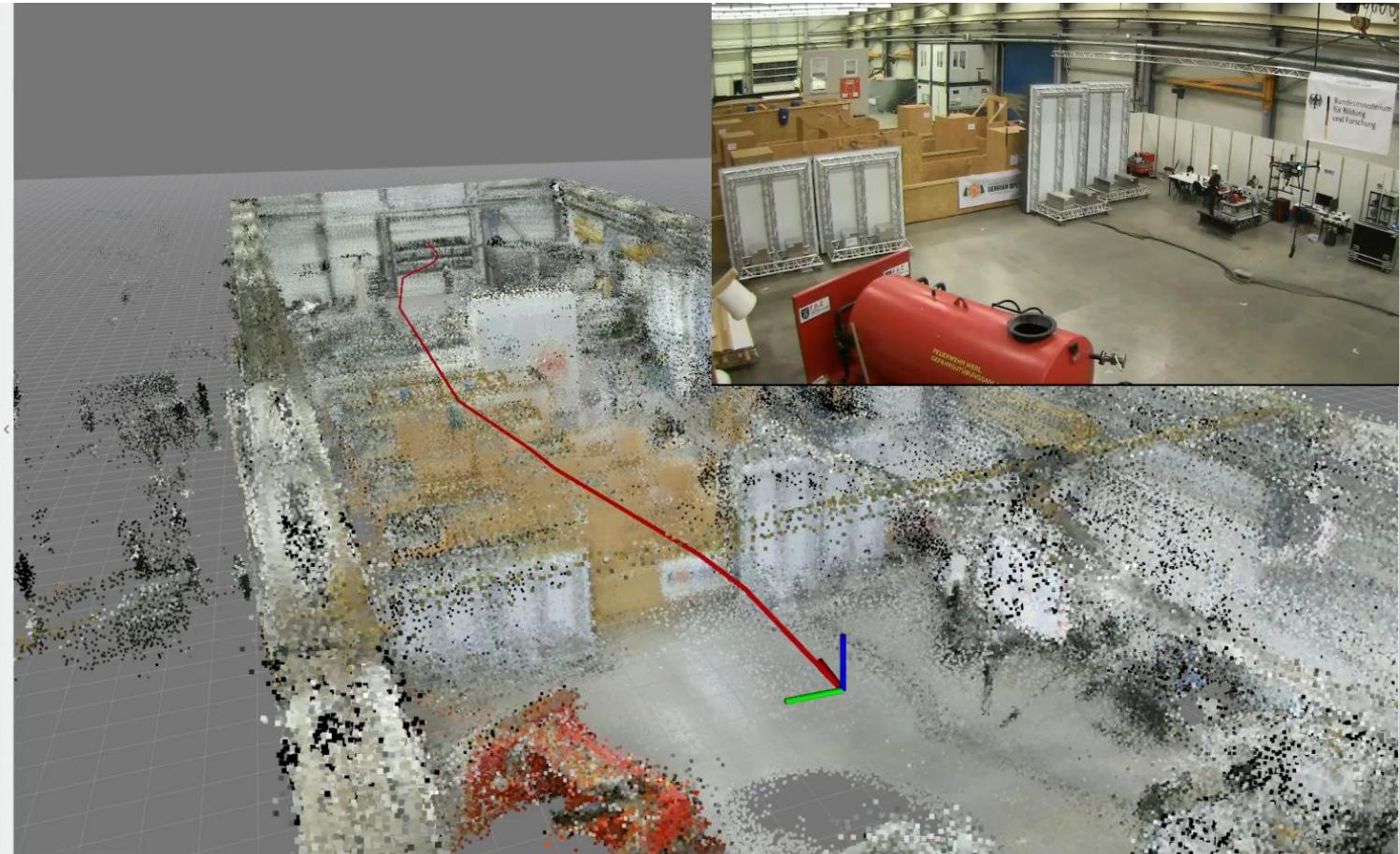
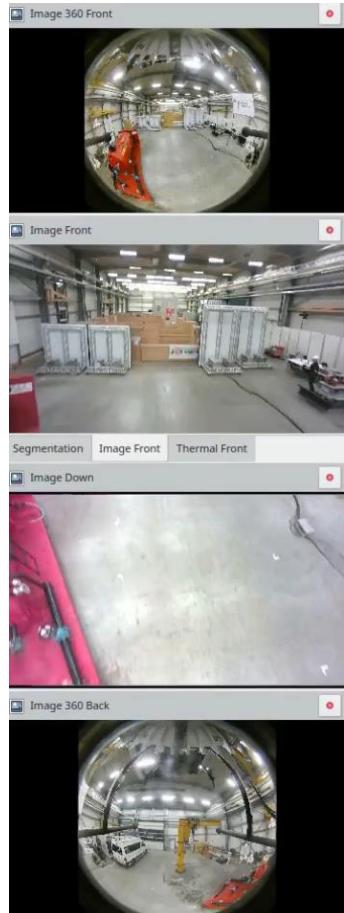


Wegpunkt-Navigation

- Vorgabe von Wegpunkten per Gamepad in der aktuellen 3D-Karte
- Anzeige der geplanten Trajektorie und autonomer Abflug nach Freigabe durch Operator
- Hindernisvermeidung passt Trajektorie an



Wegpunkt-Navigation



[Schleich and Behnke, ICRA 2021]

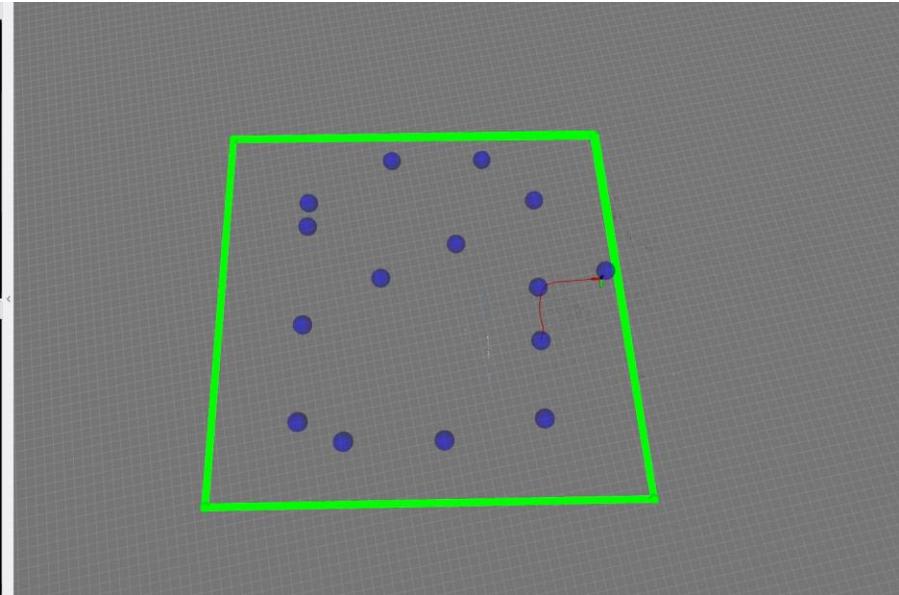
Exploration

- Markierung des Zielgebiets auf Satellitenbildern oder Straßen-karten
- Prüfung von initialem Explorationsmuster (Mäander, Spirale...) auf Hinderniskollisionen
- Lösung eines Traveling-Salesman-Problems auf Bodenstation bestimmt optimale Reihenfolge
- UAV plant kontinuierlich onboard dynamische Trajektorien durch die nächsten Wegpunkte



Exploration

- Zielbereich vorgegeben (grünes Rechteck)
- UAV plant kontinuierlich onboard dynamische Trajektorien (rote Linie) durch die nächsten Wegpunkte (blau)
- Kontinuierliche Neuplanung auf der Bodenstation mit 2 Hz zur Anpassung an Hindernisse



Exploration

- Umgebungsmodell nach Exploration mit
 - initialem Muster (schwarz)
 - angepasstem Pattern (blau)
 - exploriertem Pfad (rot)
- Pattern-Wegpunkte werden lokal entsprechend der Hindernisse verschoben

