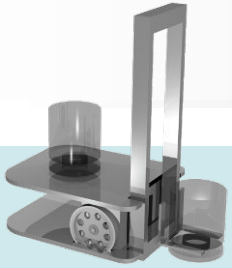
Praktikum „Mobile Roboter“
Roboter „Profi“
Gruppe A

Patrick Fleischmann
Robert Becker
Thomas Pfister
Jonas Mitschang
Marcel Zimmer
Luc Heischbourg

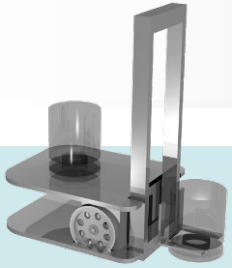
TU Kaiserslautern

Februar 2007

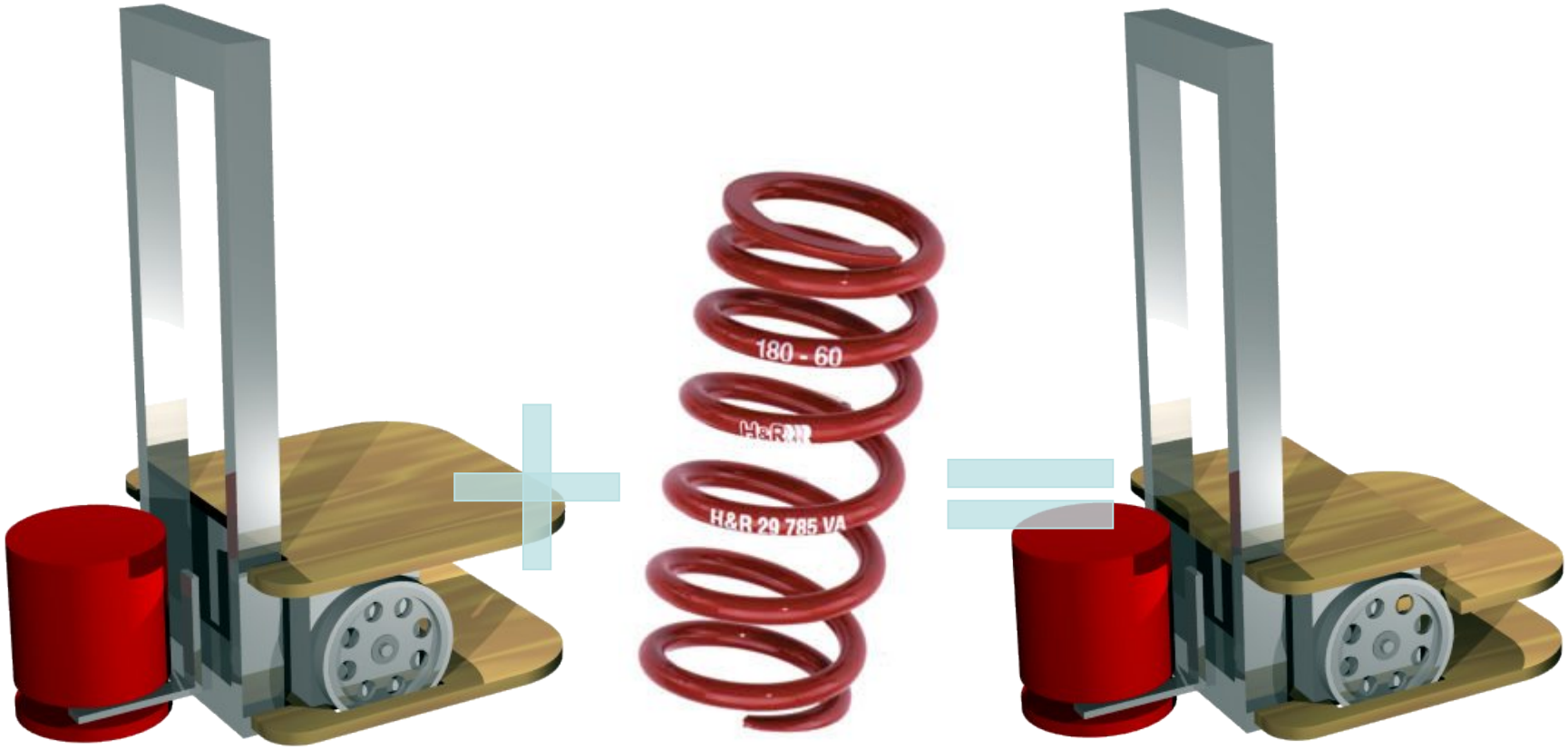


Hardware

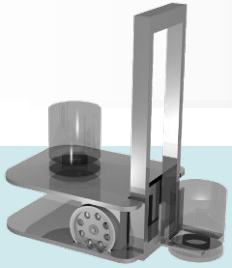
Verkürzung des Roboterchassis um 5cm um die Beweglichkeit des Roboters zu verbessern



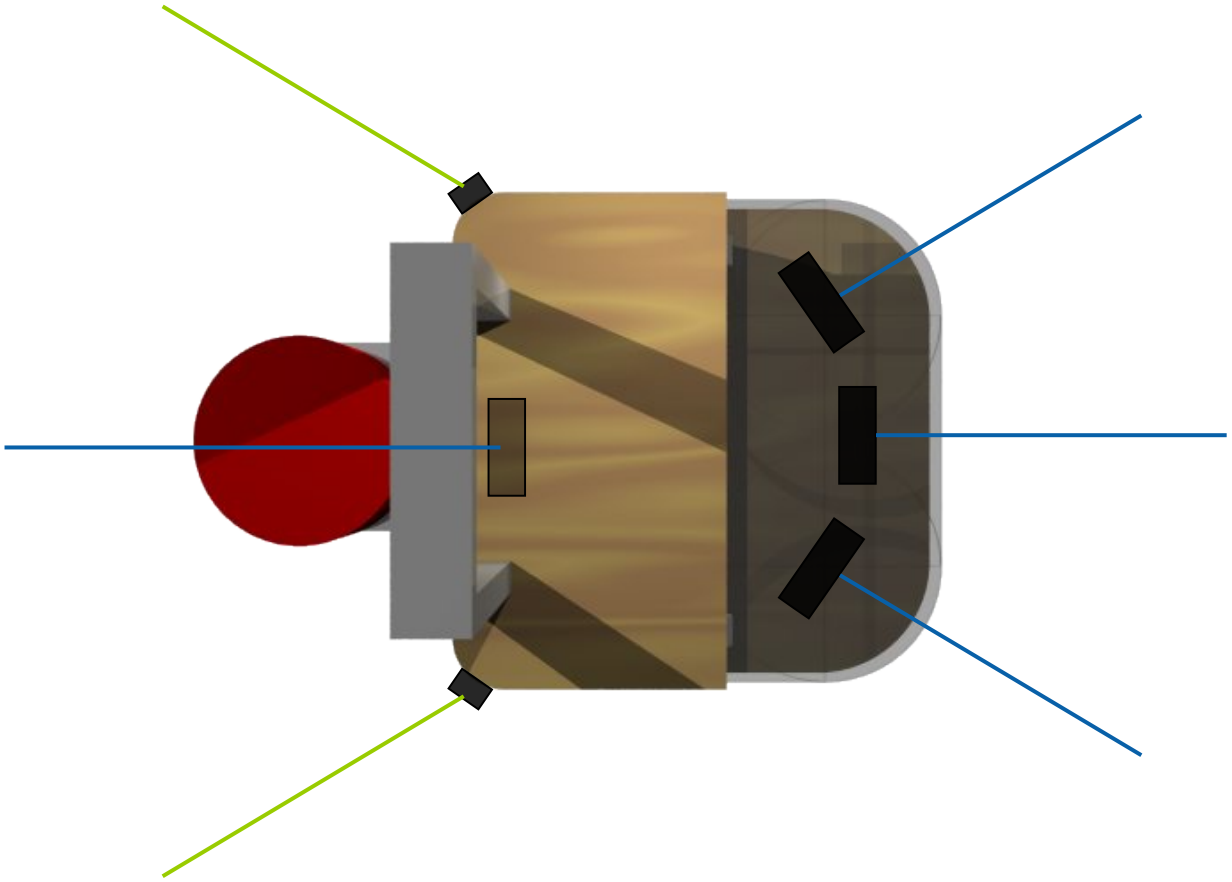
Hardware



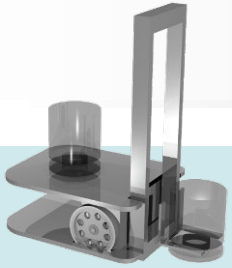
Teilweise Tieferlegung der oberen Plattform zu besseren Unterbringung des Mini-PC und der Akkus, sowie zur Verlagerung des Schwerpunktes



IR-Sensoren - Hardware



Short-Range IR-Sensoren, Long-Range IR-Sensoren



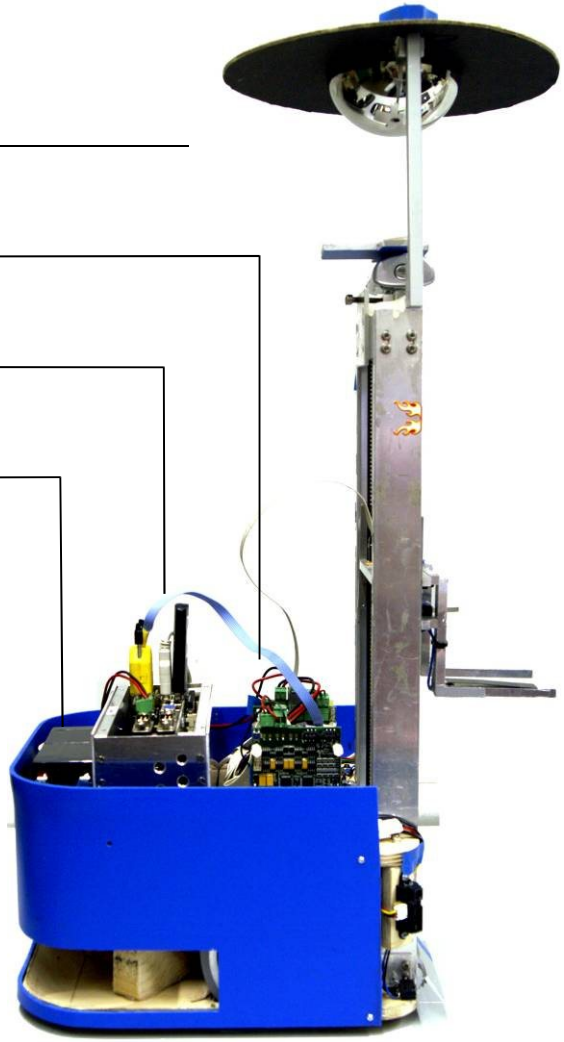
Hardware

Kamerasystem

DSP-Boards

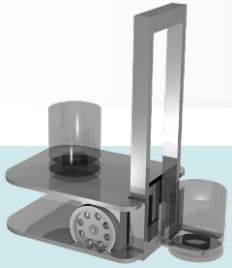
Mini-PC

Akkus



P.R.O.F.I.

Profi
Robot
Observation and
Forklifting
Individual



Übersicht

Sensors
Auswertung Taster
Auswertung IR-Sensoren
Odometrie

VehicleControl
Wegfindung
Kollisionsvermeidung
Hardwareansteuerung

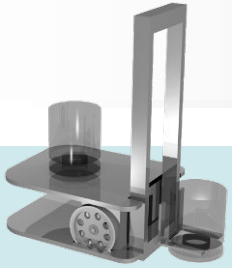
Camera
Objekterkennung

Mapping
Aufbau globaler Karte

ForkControl
Ansteuerung Gabelhw.

GUI
Darstellung Karte
Darstellung Parametern
Eingaben

MasterControl
Verhaltenssteuerung



Camera

Blackboard

```
MapObjectType type;
float pos1_angle;
float pos1_distance;
float pos2_angle;
float pos2_distance;
```

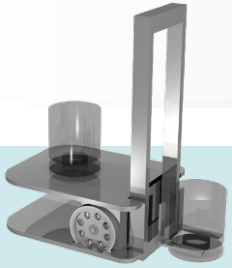
Camera

Objekterkennung

→

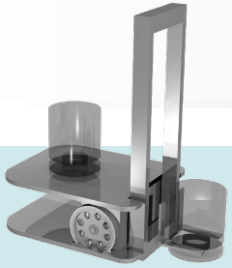
Mapping

Aufbau globaler Karte

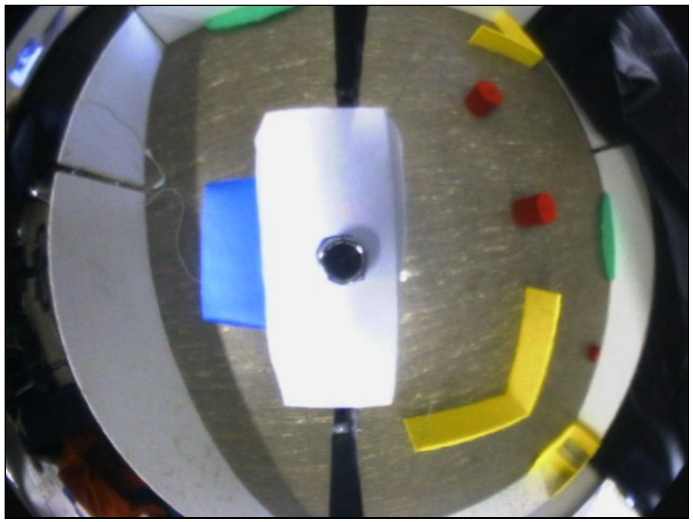


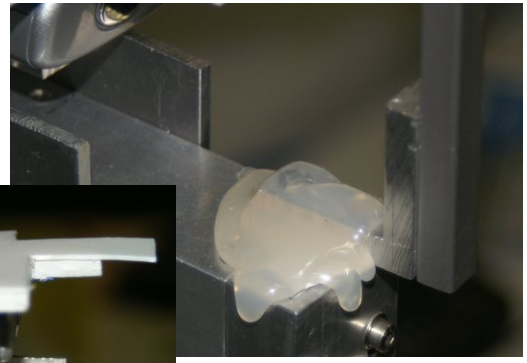
Kamerasystem / „Omnivision“ - Camera

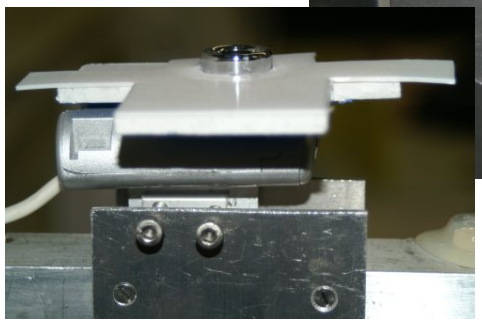
- Technik
  - Indirektes Kamerabild über Spiegelung an einer Kugel (Suppenkelle)
  - Auflösung 640x480 Pixel
  - 1 Bild/s
  - „automatisierter“ Weißabgleich durch weiße Fläche um die Kamera
- Vorteile
  - größeres Sichtfeld in alle Richtungen



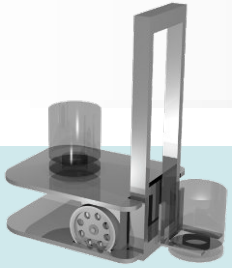
Kamerasystem / „Omnivision“(2) - Camera



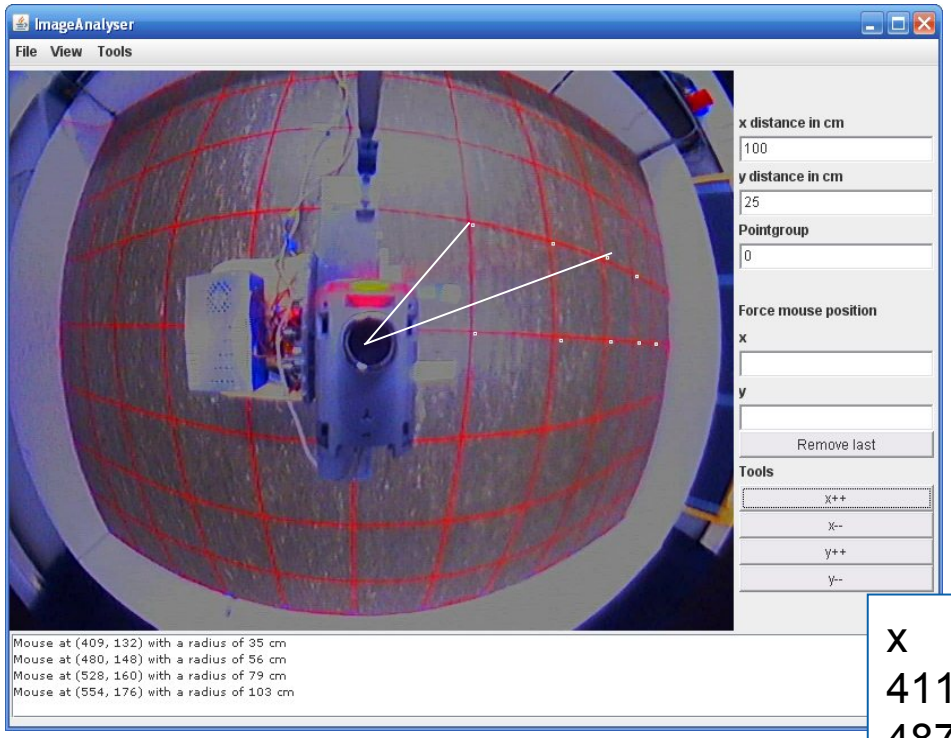




- Nachteile
  - Verzerrungen durch Verarbeitungsfehler / Verschmutzungen des Reflektors
  - Reflektor nicht kugelförmig
  - geringere Auflösung (als nach vorne gerichteter Ansatz)
  - Komplexere Berechnungen
  - Ausrichtung, Befestigung, Justierung aufwändig und fehleranfällig

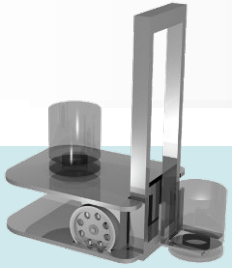


Abstandskalibrierung - Camera



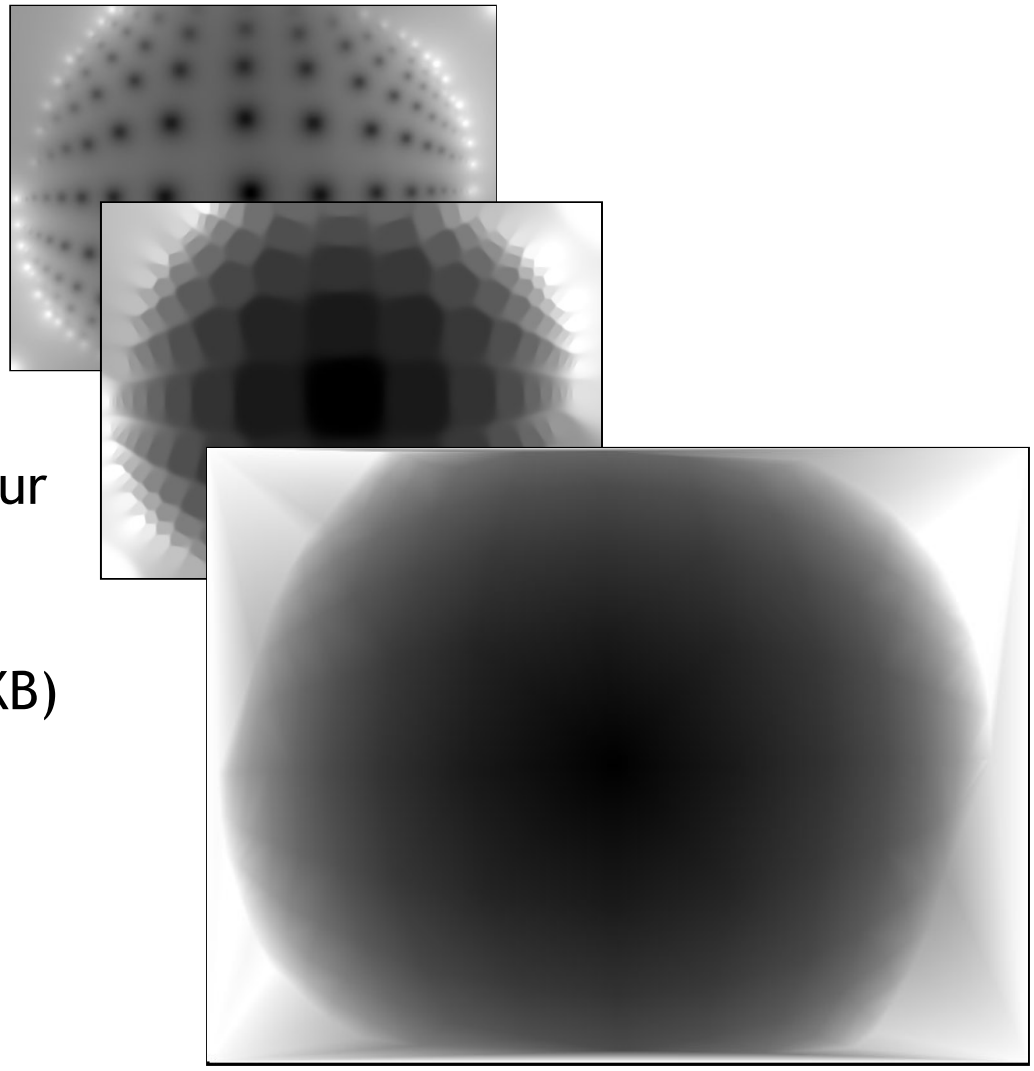
- Aufnahme eines Punkterasters (25x25cm)
- Markieren der Punkte mit Einlesen des Radius (Abstand Rasterpunkt – Roboter)

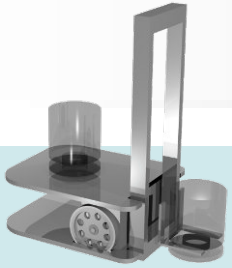
x	y	distance
411	225	25
487	231	50
531	232	75
556	233	100
571	234	125
409	132	35



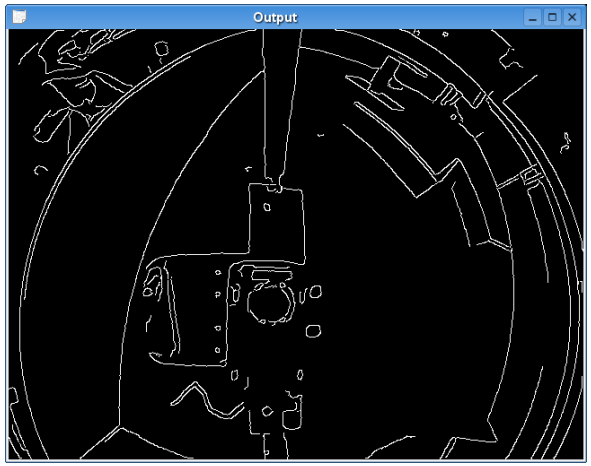
Abstandskalibrierung (2) - Camera

- 3D-Interpolation der Punktematrix mit Matlab
- ✂ → vorgerechnete Abstandswerte
- Vorteile
  - Keine „Berechnungen“ zur Laufzeit
  - geringer Speicherverbrauch (300KB)
- Nachteile
  - Fehler bei der Interpolation



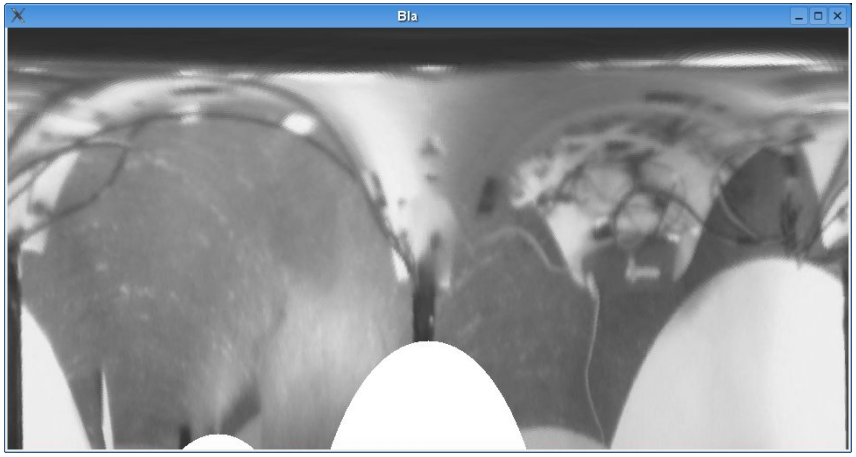


Vorversuche (1) - Camera



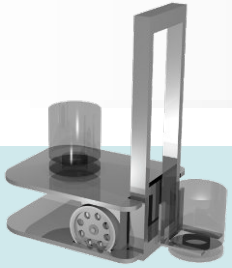
Oben:

Kantendetektion mittels Canny-Algorithmus zur Kantenextraktion



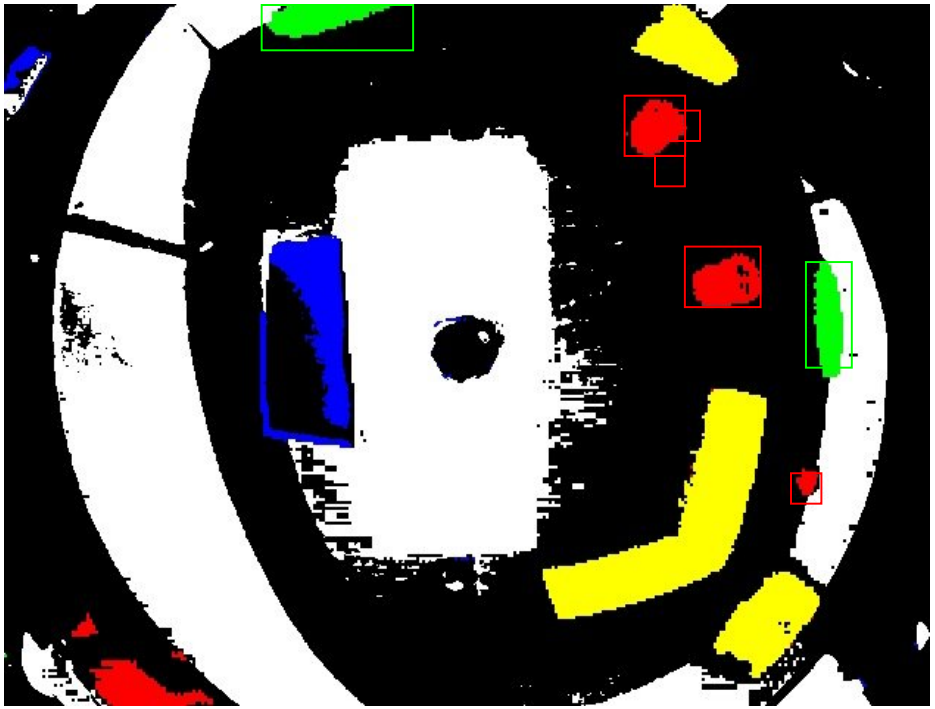
Unten:

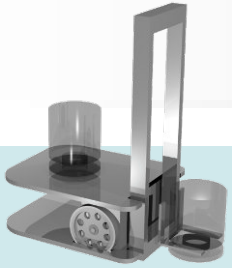
Entzerrung des Kamerabildes (Versuch)



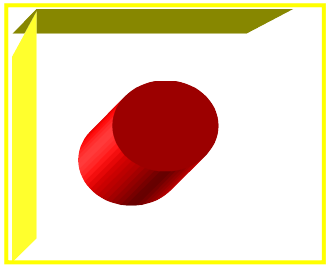
Lastobjekt-, Baseerkennung - Camera

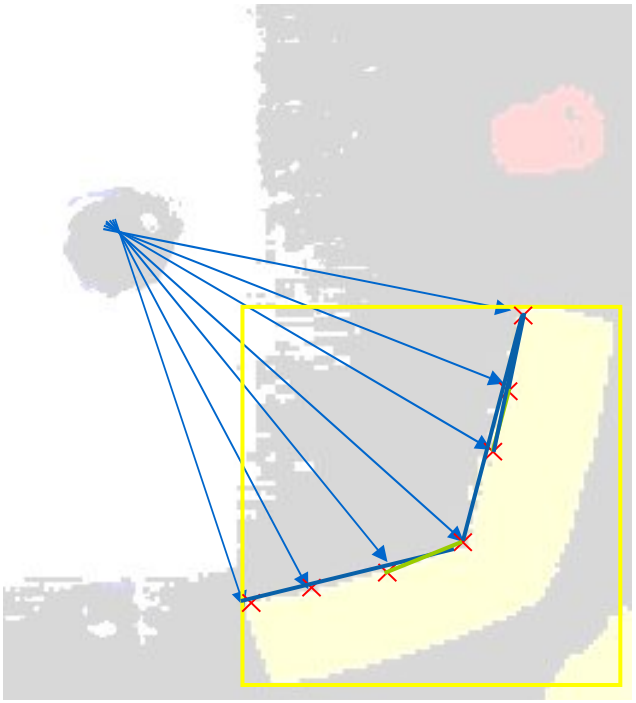
- Einsatz von CMVision-Funktionen (getRegions) zur Lastobjekt-, Basen- und Gegnererkennung
- Mehrere Modifikationen:
  - Ignorieren kleiner Bereiche
  - Zusammenfassen kleiner benachbarter Bereiche
- Übergabe der Objekte in Polarkoordinaten an die Karte (Mapping)



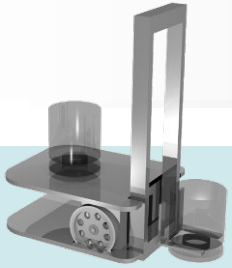


Wanderkennung - Camera

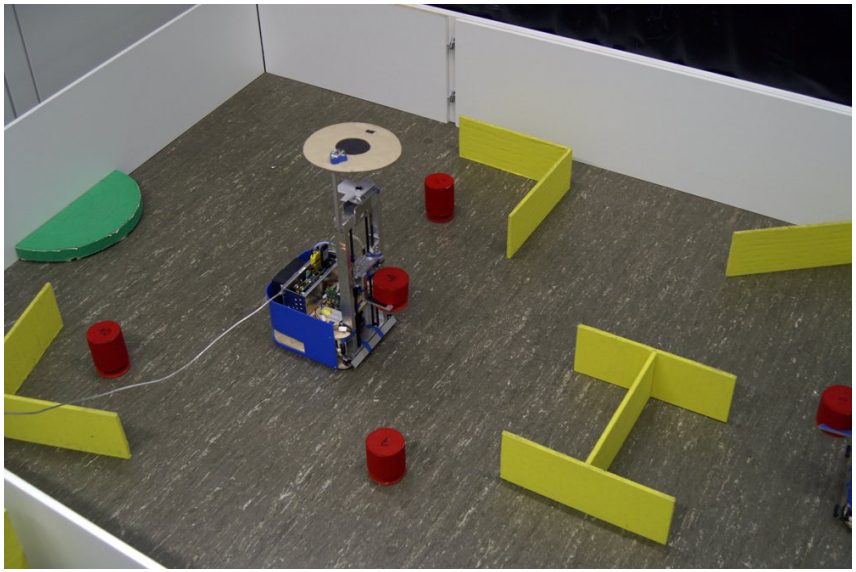


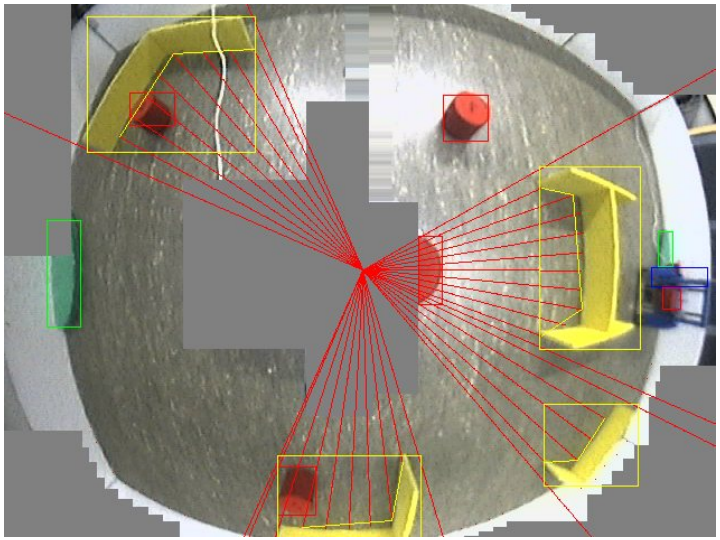


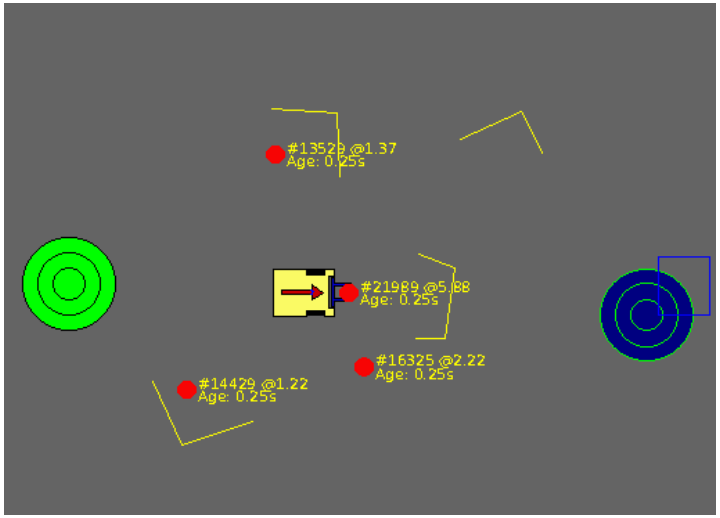
- Bounding Boxes nicht ausreichend (versperrte Wege, nicht erreichbare Lastobjekte)
- „Strahlenalgorithmus“
  - Erkennung einzelner Randpunkte der Wand
  - verbinden der Punkte zu einem Liniensegment
  - Erzeugung eines neuen Liniensegments, falls kritischer Winkel überschritten wird

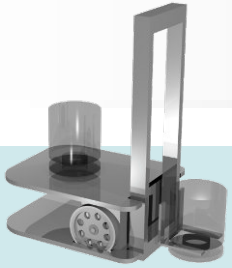


Wanderkennung (2) - Camera





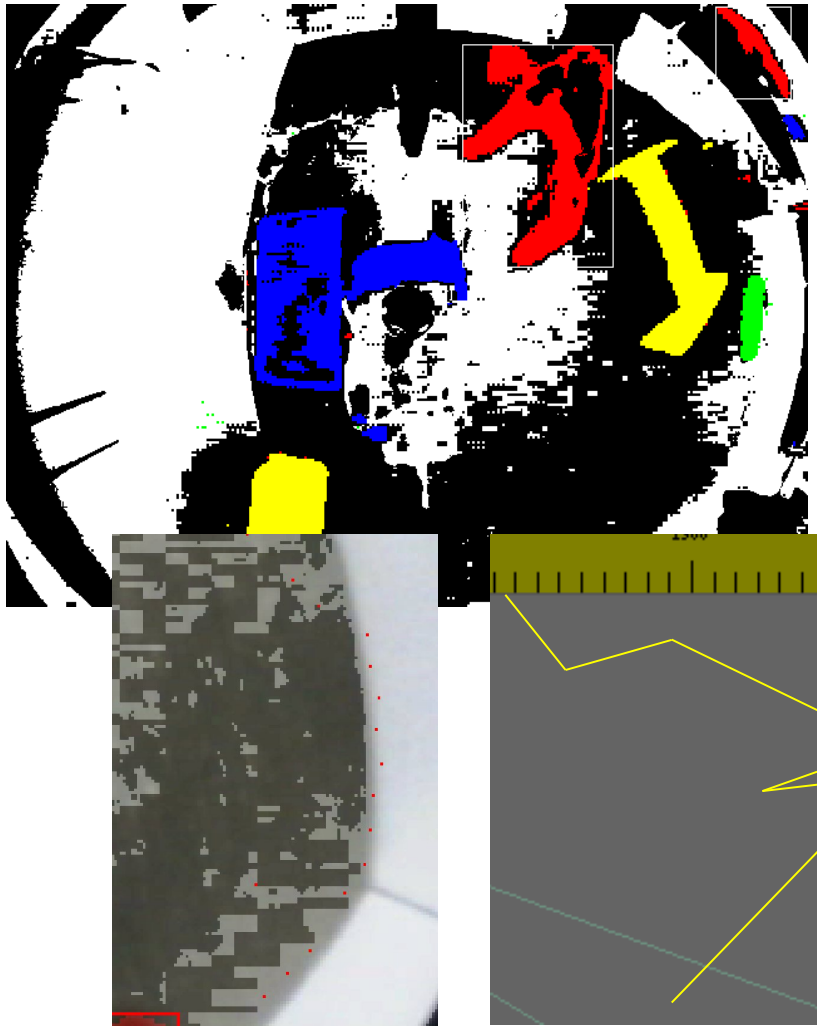


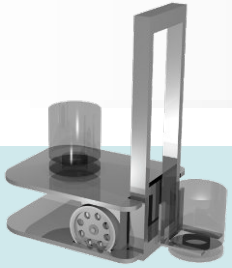


Seitenwanderkennung - Camera

- Notwenig um Objekte außerhalb der Seitenwände zu ignorieren
- Begrenzung des Spielfeldes zu ermitteln
- Problem:
  - Lichtreflexionen auf dem Boden
  - Abstandsberechnung weit entfernter Objekte extrem ungenau (+-20cm)

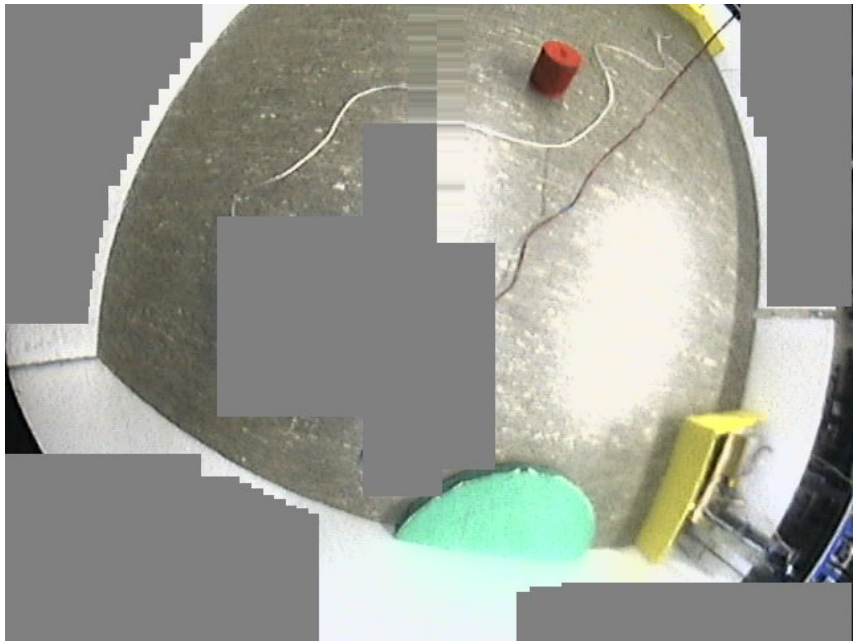
  → Seitenwand instabil in der Karte

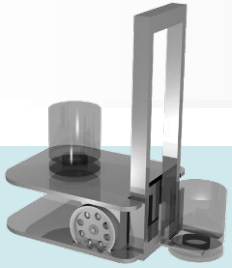




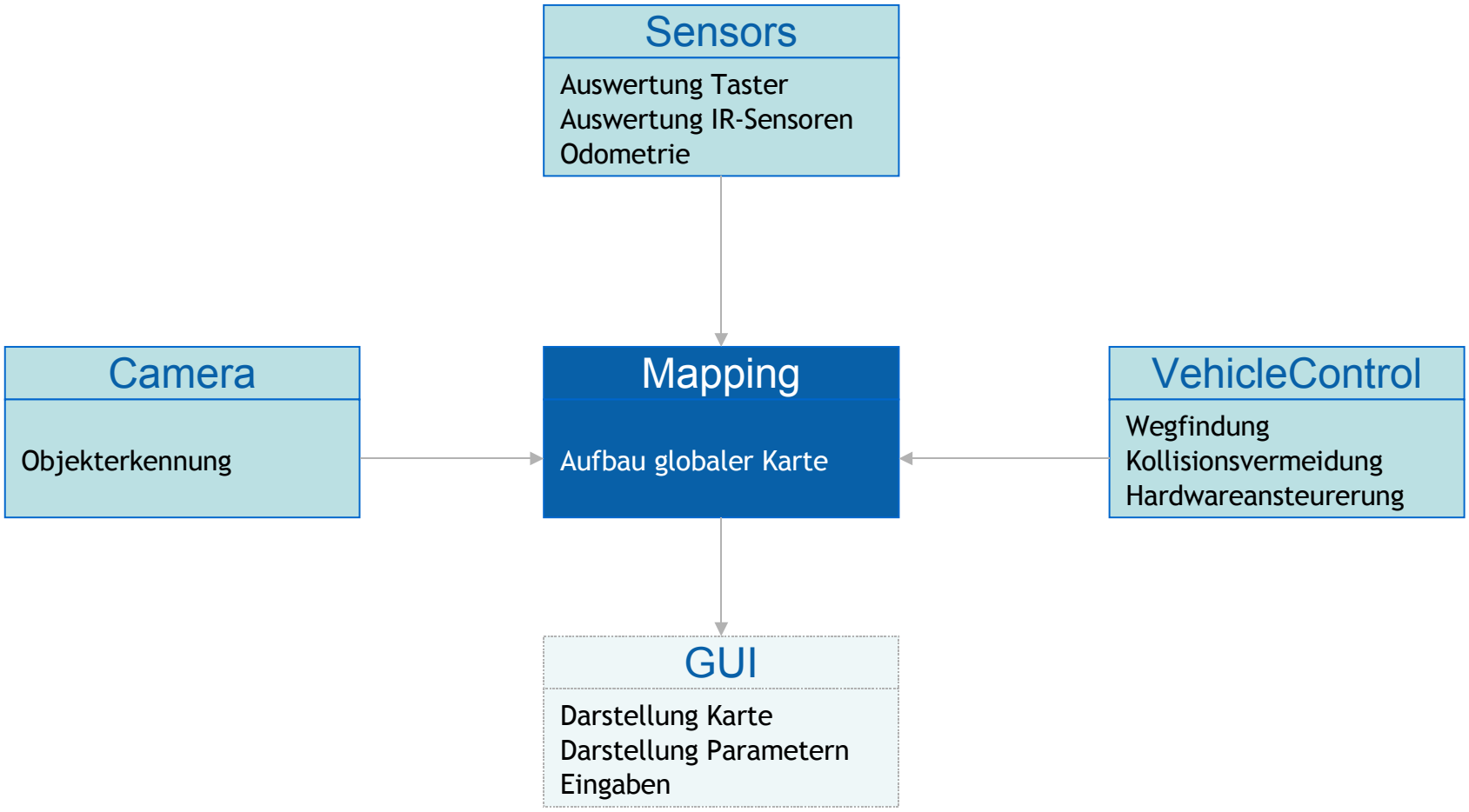
Seitenwanderkennung (2) - Camera

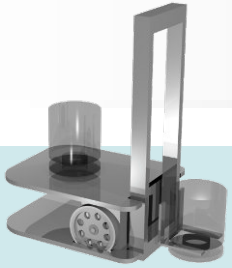
- Deshalb:
  - Beschränkung auf das Verwerfen der Objekte hinter den Seitenwänden
- Effiziente Implementierung durch destruktives Arbeiten auf dem Bild
  - Reduziert CMVision-Aufwand



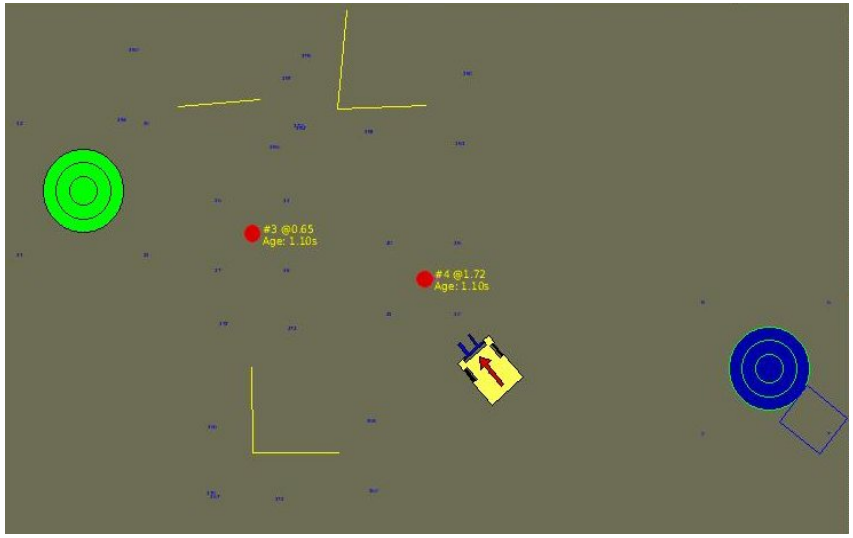


Mapping

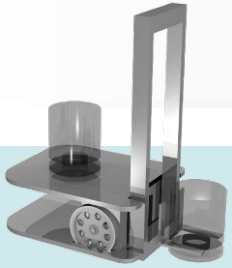

Sensors
Auswertung Taster
Auswertung IR-Sensoren
Odometrie
Camera
Objekterkennung
Mapping
Aufbau globaler Karte
VehicleControl
Wegfindung
Kollisionsvermeidung
Hardwareansteuerung
GUI
Darstellung Karte
Darstellung Parametern
Eingaben



Globale Karte - Mapping

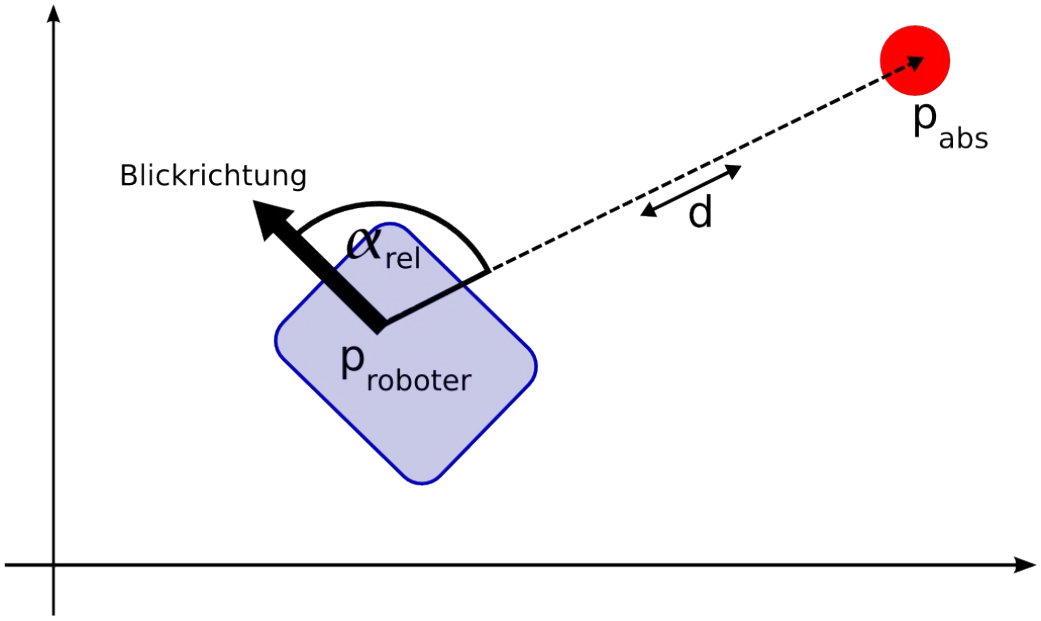


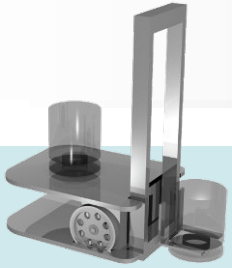
- Vorteile
  - „souveränes“ Navigieren auf dem Spielfeld
  - komplexe Bewertung des Umfelds
  - leichte Wiedererkennung von Objekten
- Nachteile
  - Große Probleme bei ungenauer Odometrie



Erstellen der absoluten Karte - Mapping

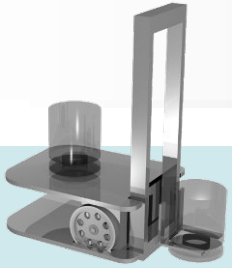
- Kamera liefert Polarkoordinaten relativ zum Roboter
- Umrechnung in absolute karthesische Koordinaten:
  - $p_{abs} = (\alpha_{rel} + \alpha_{blickrichtung}, d).toCartesian() + p_{roboter}$





Erstellen der Karte: Pömpel - Mapping

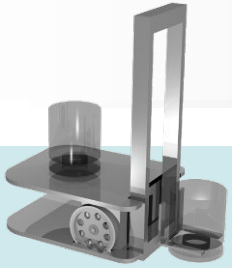
- Wiedererkennungsradius: 20 cm
- Aktualisieren des Alters
- Nach 3s: Entfernen des Objekts
- Ignorieren von Lastobjekten:
  - auf den Basen
  - auf der Gabel



Erstellen der Karte: Basen - Mapping

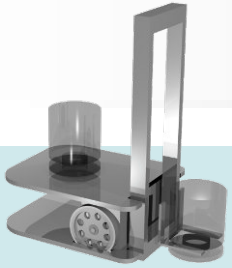
- Basen weit auseinander
- Feste Position auf Karte
- Gute Erkennung durch Kamera

  - Wiedererkennungsradius groß: 1m
  - Eigene Basis
    - Wird sicher erkannt
    - Unbegrenzte Lifetime
  - Lifetime gegnerische Basis: 15s

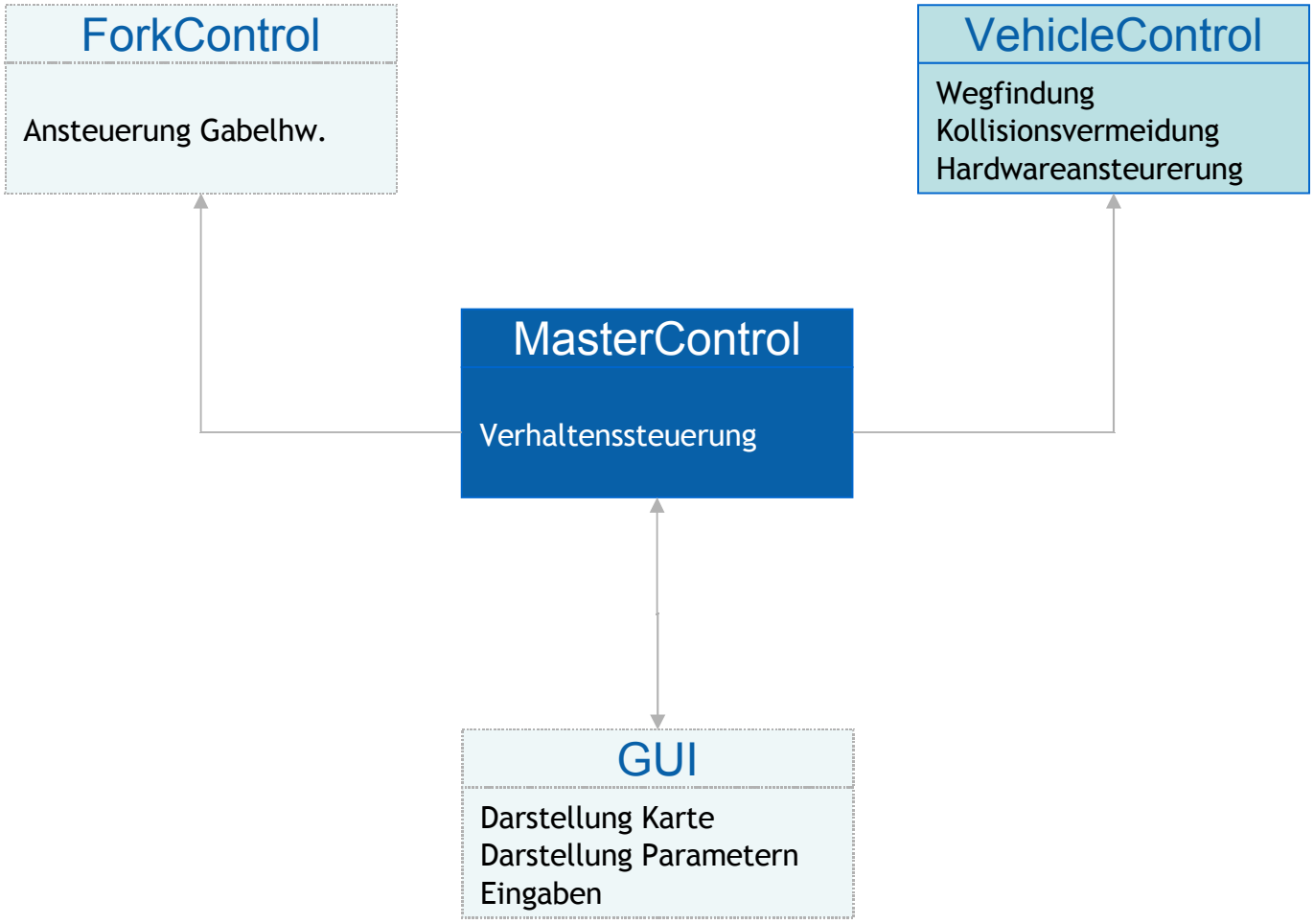


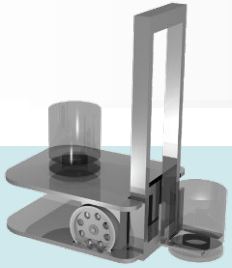
Darstellung der Karte - Mapping

- Basen
  - Implementierung gefüllter Kreise für das MCA GeometryBackboard
  - Farbliche Unterscheidung
- Pömpel
  - Implementierung von Textausgabe für das MCA GeometryBackboard
  - Anzeige von ID, Rating und Alter
  - Ausblenden sterbender Pömpel
- Hindernisse
- Wegpunkte mit A* Graph
- Roboter



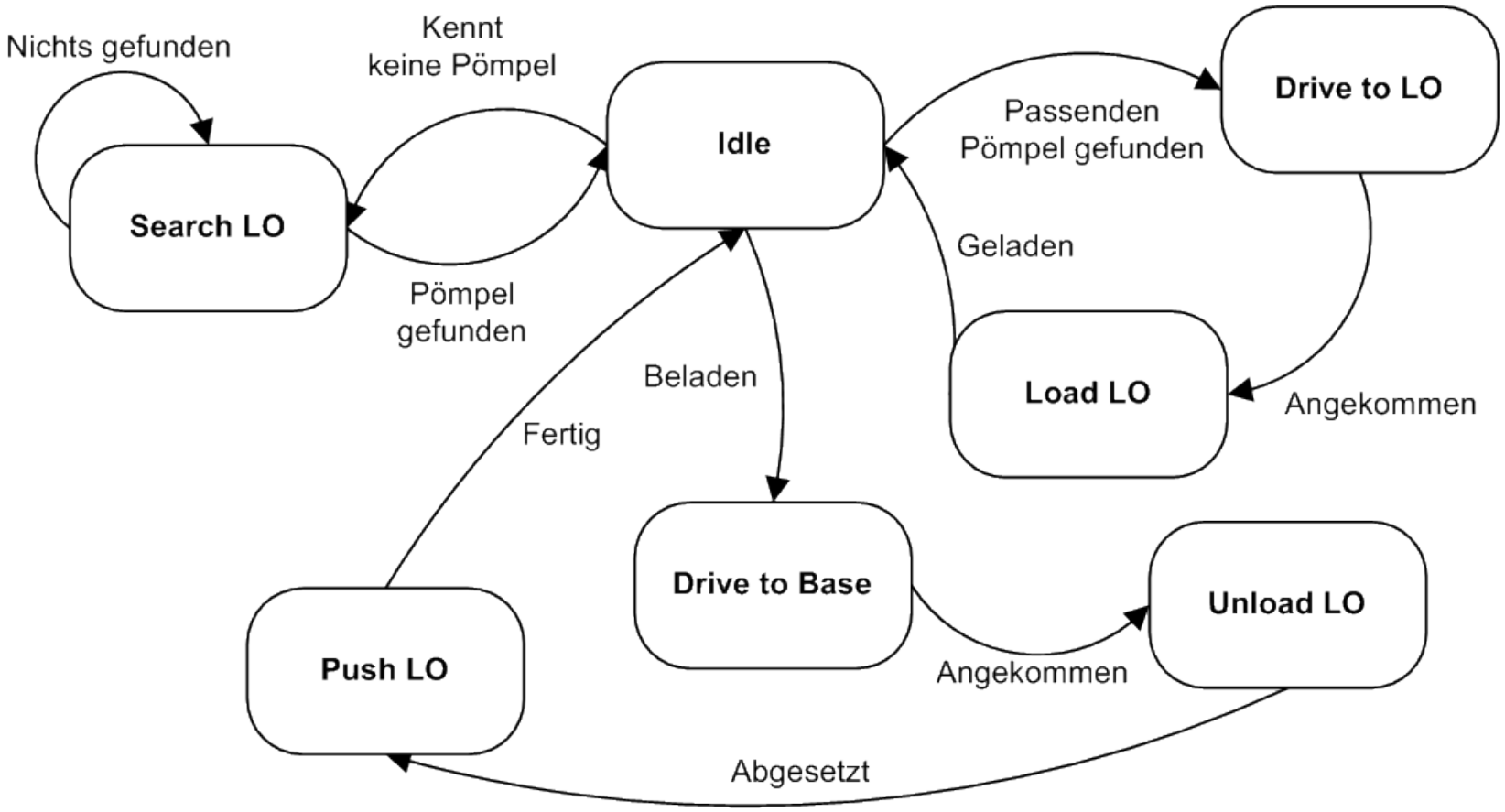
MasterControl

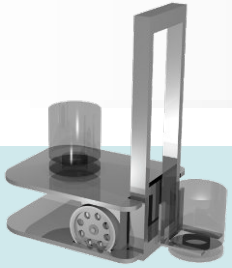




Zustände - MasterControl

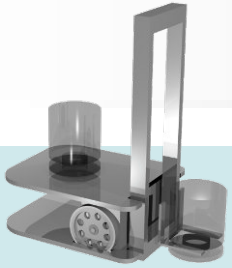
- Vereinfachtes Zustandsdiagramm





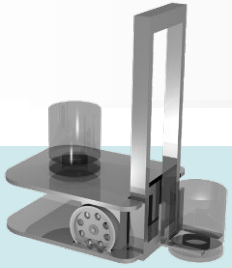
Drive Modes - MasterControl

- Verschiedene „Fahr-Modi“:
  - **DRIVE_MODE_STOP:** Stop
  - **DRIVE_MODE_GOTO_OBJ:** Fahre Objekt an
  - **DRIVE_MODE_GOTO_XY:** Fahre Koordinaten an
  - **DRIVE_MODE_CUSTOM_DRIVE:** Direkte Steuerung (Joystick)
  - **DRIVE_MODE_ROTATE:** Drehe um relativen Winkel
  - **DRIVE_MODE_DISTANCE:** Fahre bestimmte Stecke geradeaus
  - **DRIVE_MODE_ROTATE_TO_YAW:** Drehe auf absoluten Winkel
- Parameter über zusätzliche Kanten

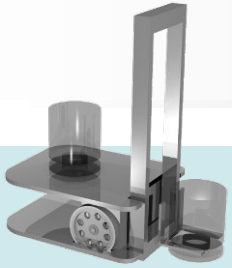


Auf der Suche nach Pömpeln - MasterControl

- Dank Omnivision Erkennung nahezu aller Pömpel
- Bewertung:
  - Anzahl fehlgeschlagener Anfahrversuche (CollisionAvoidance)
  - Alter des Lastobjekts
  - Entfernung vom eigenen Roboter
  - Nähe zum Gegner
  - Entfernung zu Hindernissen

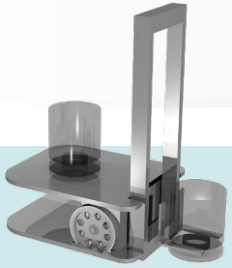


VehicleControlSystem (VCS)

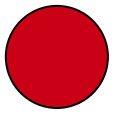


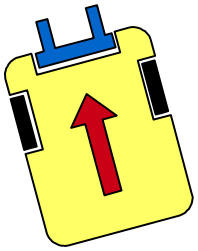
Bahnplanung - VCS

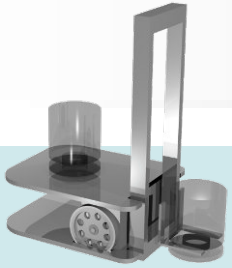
- Aufbau eines Sichtbarkeitsgraphen
- Suchen des kürzesten Weges durch den A*-Algorithmus
- Abfahren der einzelnen Wegpunkte
- Kollisionsvermeidung durch „Obstacle-Avoidance“



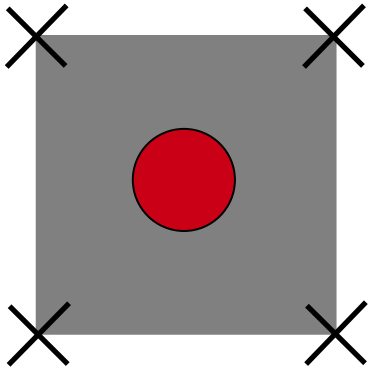
Sichtbarkeitsgraph - VCS

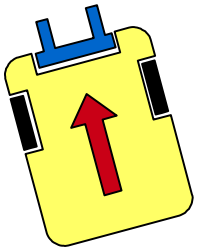


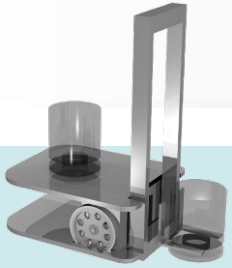




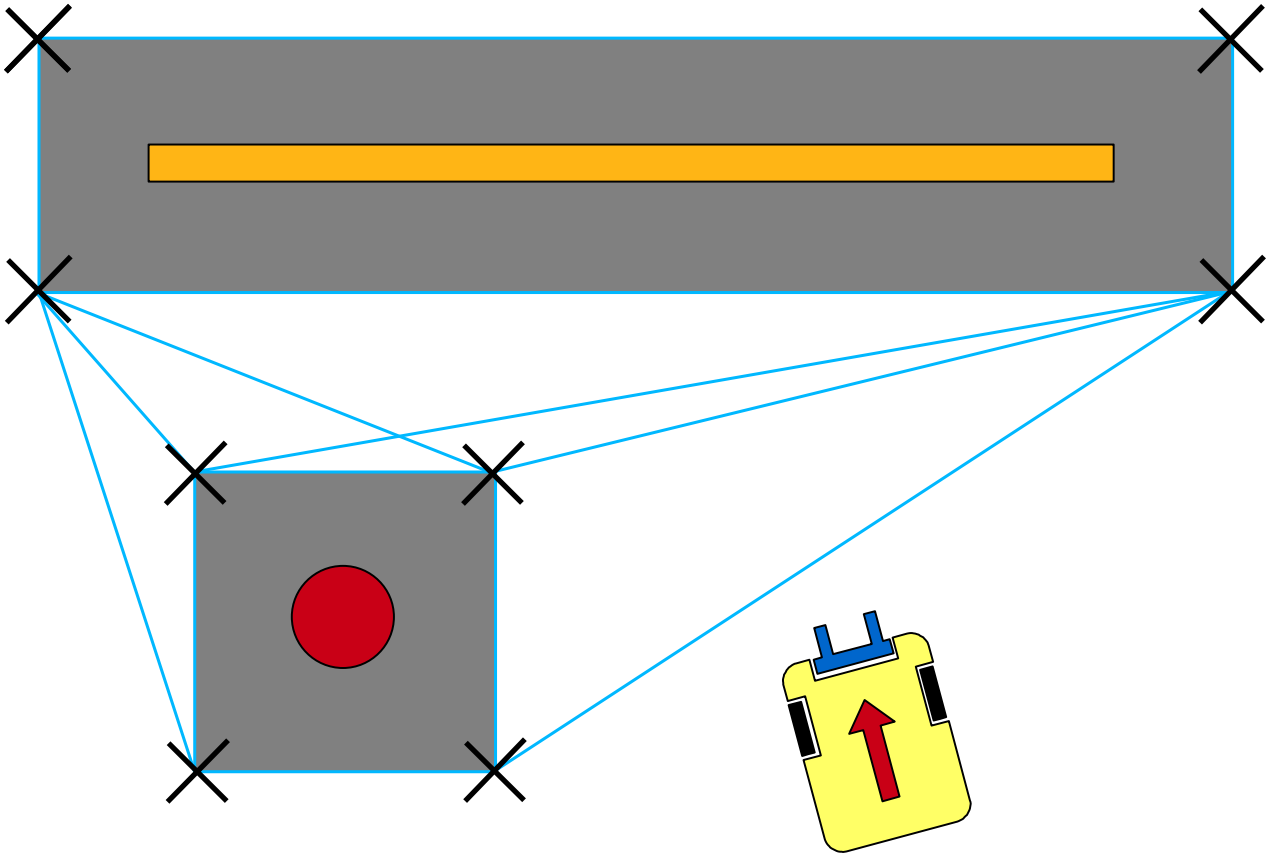
Sichtbarkeitsgraph - VCS

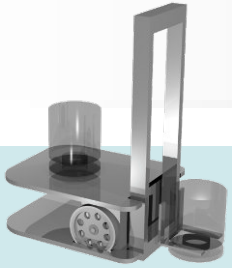




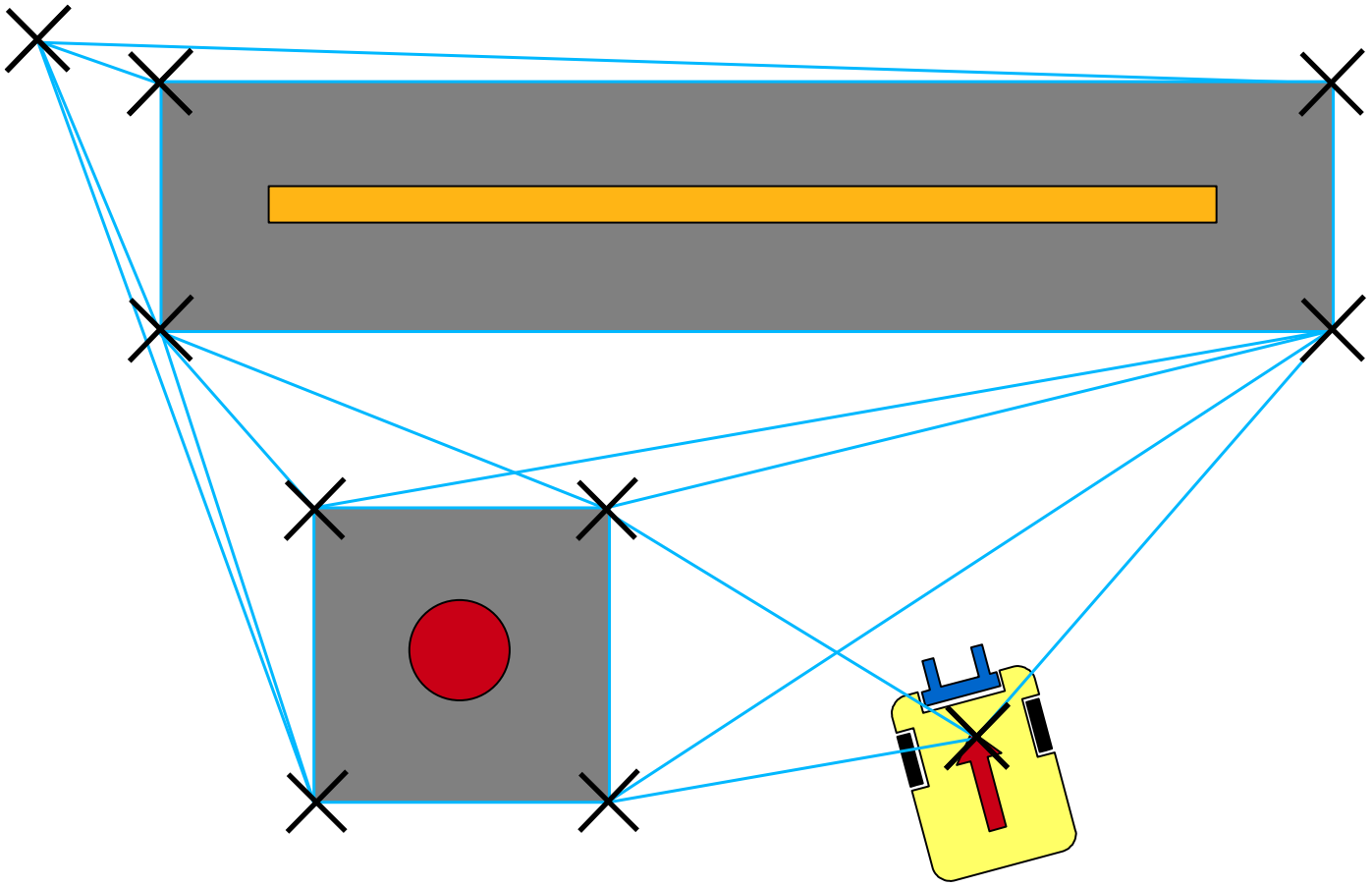


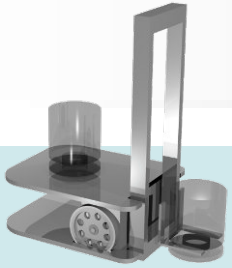
Sichtbarkeitsgraph - VCS



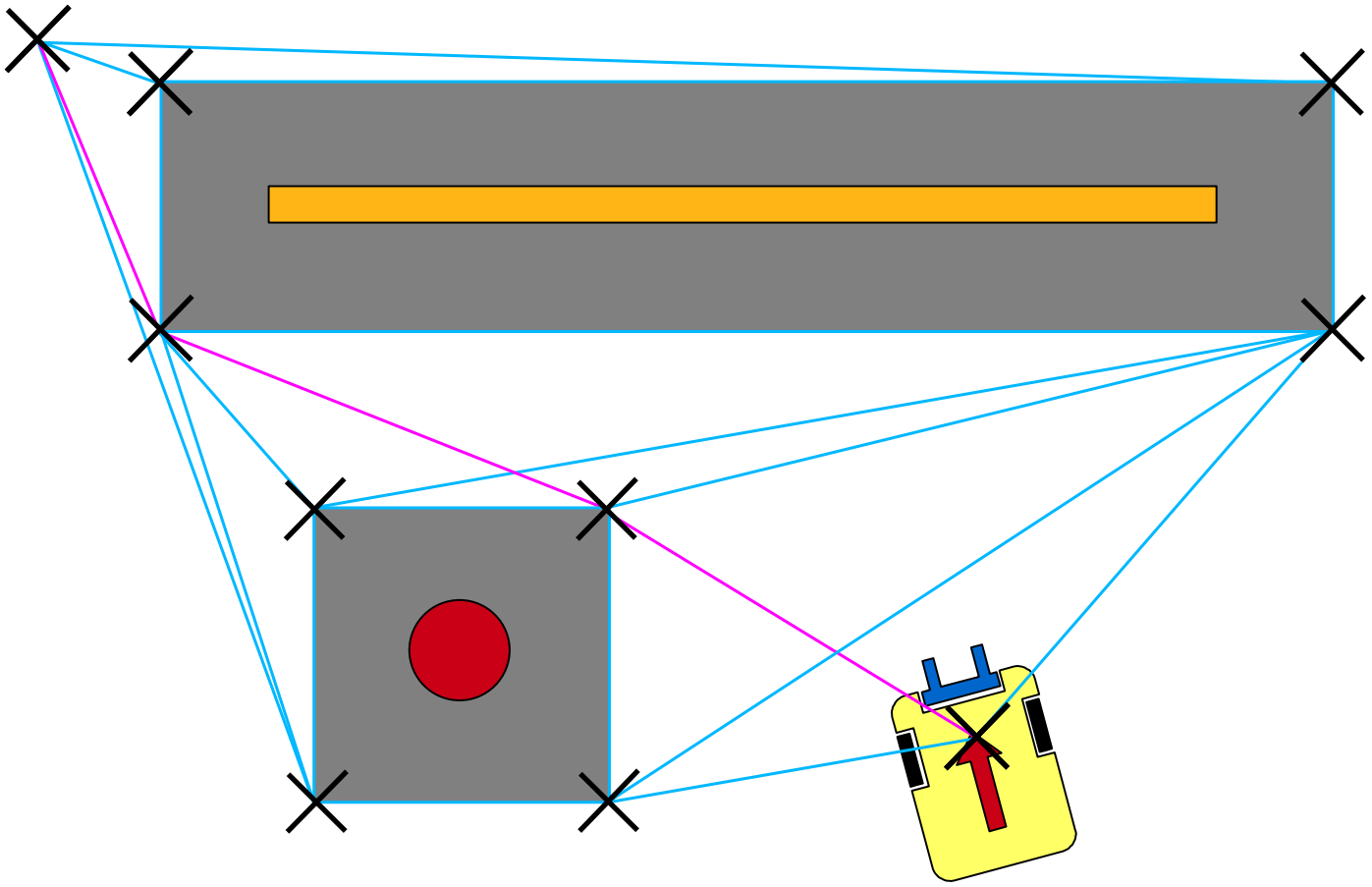


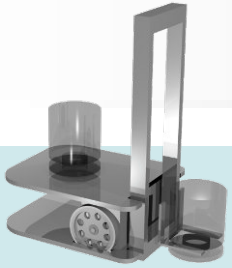
Sichtbarkeitsgraph - VCS





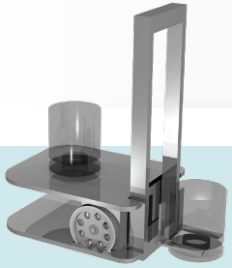
Sichtbarkeitsgraph - VCS



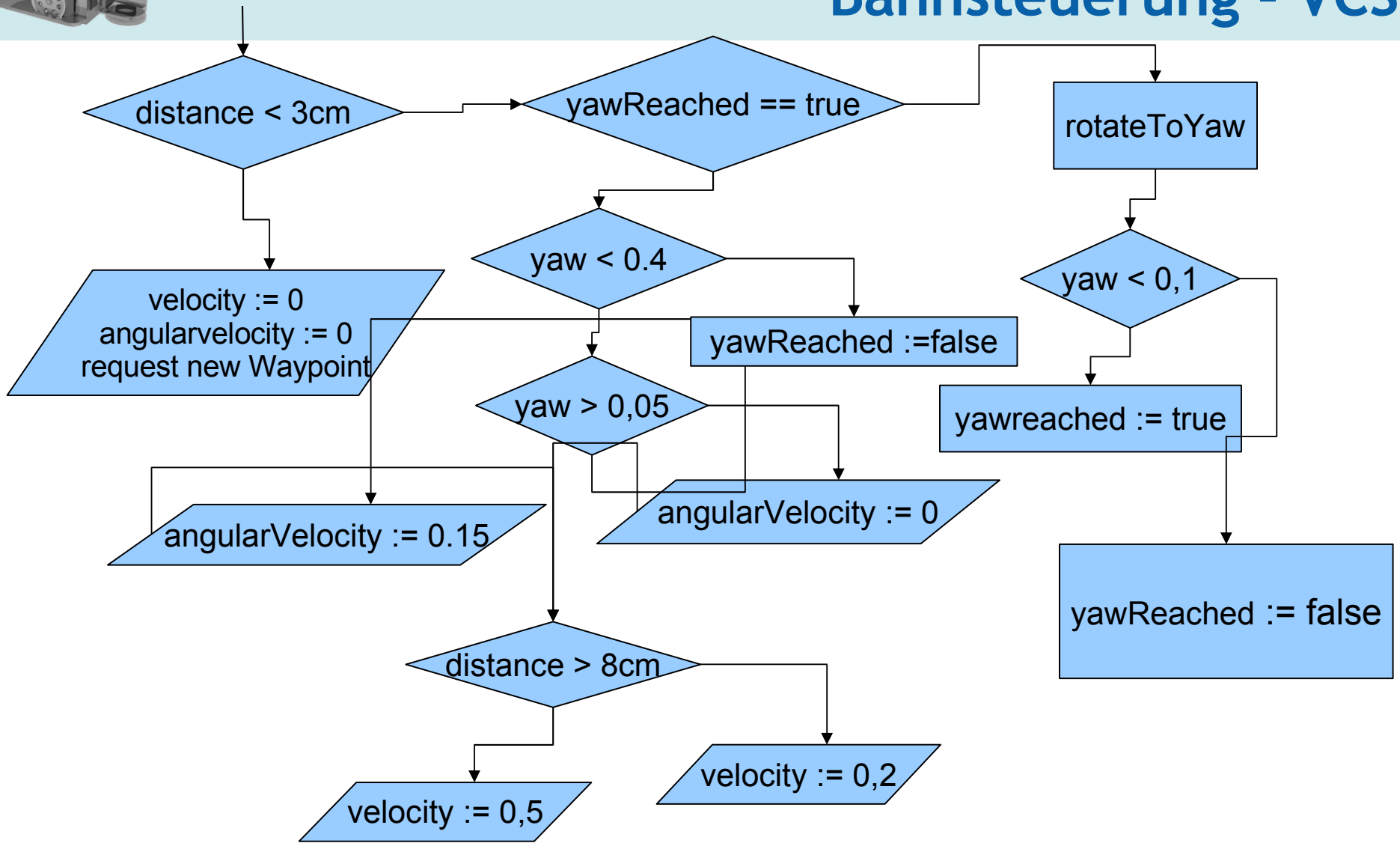


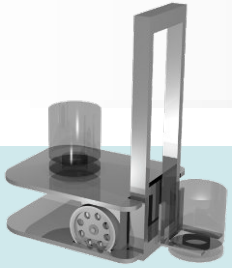
A Star - VCS

- Der A*-Algorithmus dient der Berechnung des kürzesten Weges zwischen zwei Knoten in einem kantengewichteten Graphen.
- Der A*-Algorithmus basiert auf einer zweiteiligen Auswertefunktion für jeden Knoten. Die Auswertefunktion setzt sich aus der Wegstrecke zum Startknoten und einer Schätzfunktion für die zu erwartende kürzeste restliche Wegstrecke zusammen.

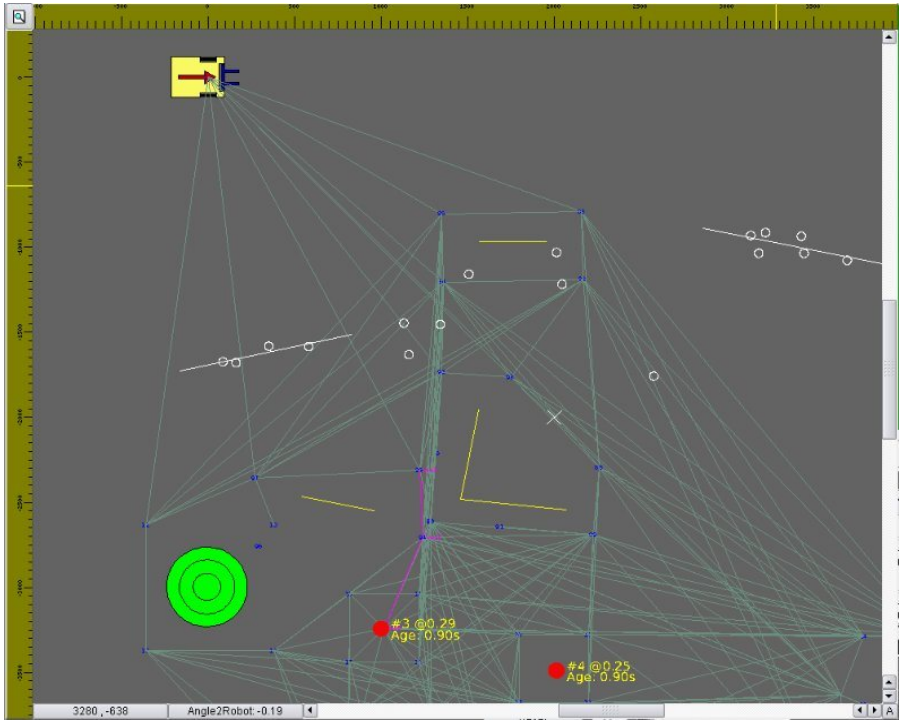


Bahnsteuerung - VCS



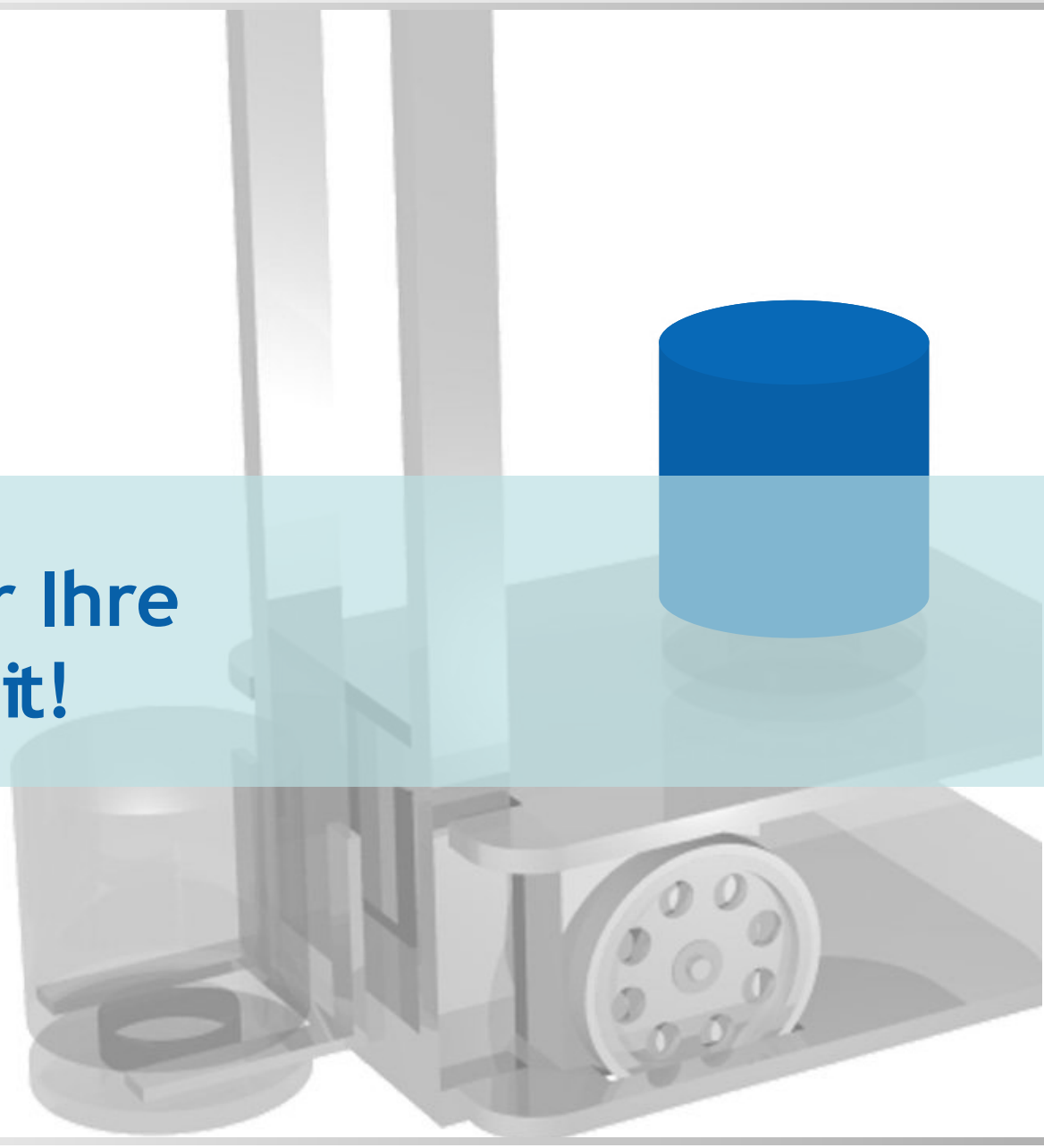


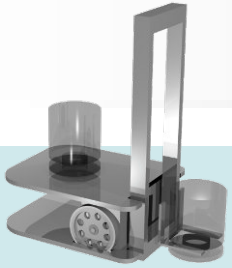
Bahnplanung - VCS



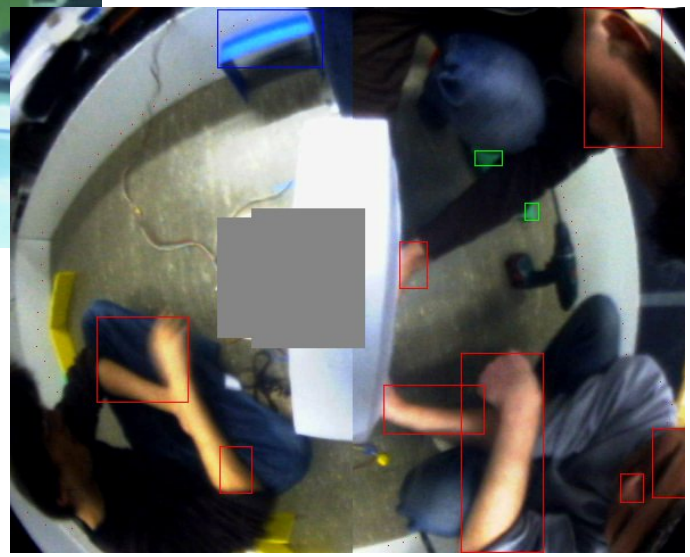
- Vorteile
  - finden des kürzesten Weges
  - Voraussagen ob der Zielpunkt erreichbar
- Nachteile
  - weniger geeignet für dynamische Umgebungen
  - nicht immer schnellster Weg

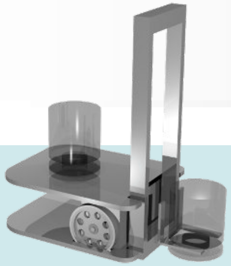
Vielen Dank für Ihre Aufmerksamkeit!





Outtakes - Anhang





IR-Sensoren (Kalibrierung) - Anhang