Bahnplanung

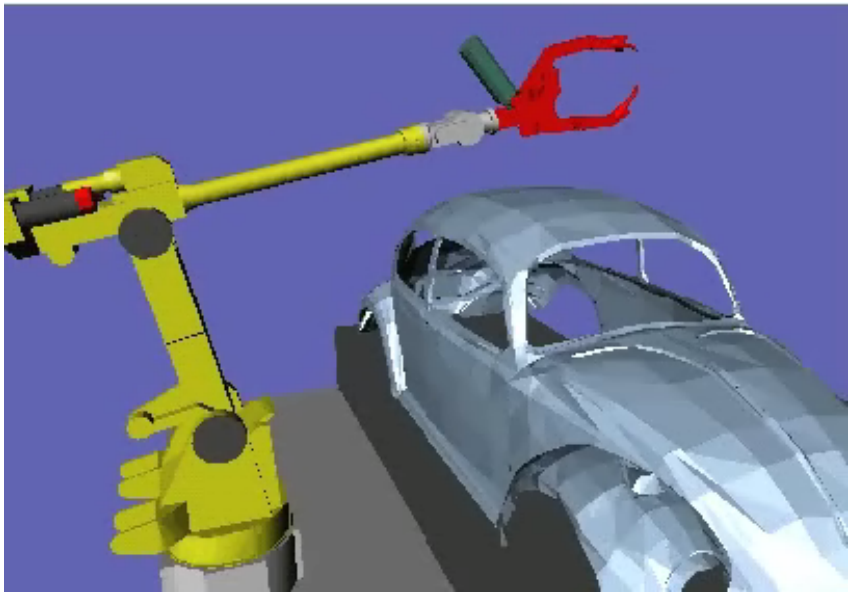
Prof. Dr.-Ing. Rüdiger Dillmann

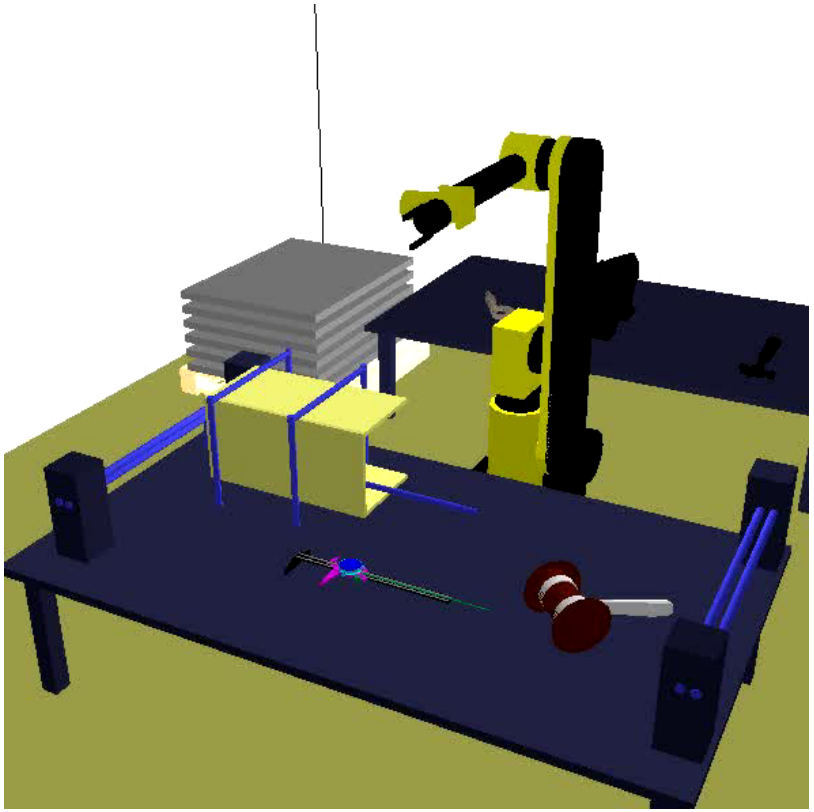
Inhalt

- Motivation
- Grundlagen der Bahnplanung
- Planungsverfahren für
  - Mobile Roboter
  - Manipulatoren
  - Robotergreifer

Motivation

Bahnplanung zur Erzeugung von kollisionsfreien Bewegungen

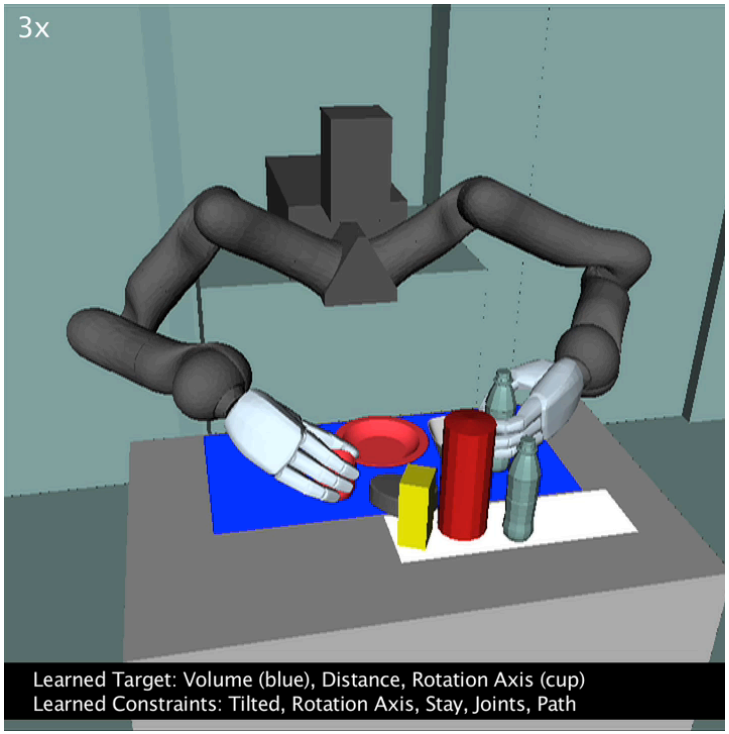


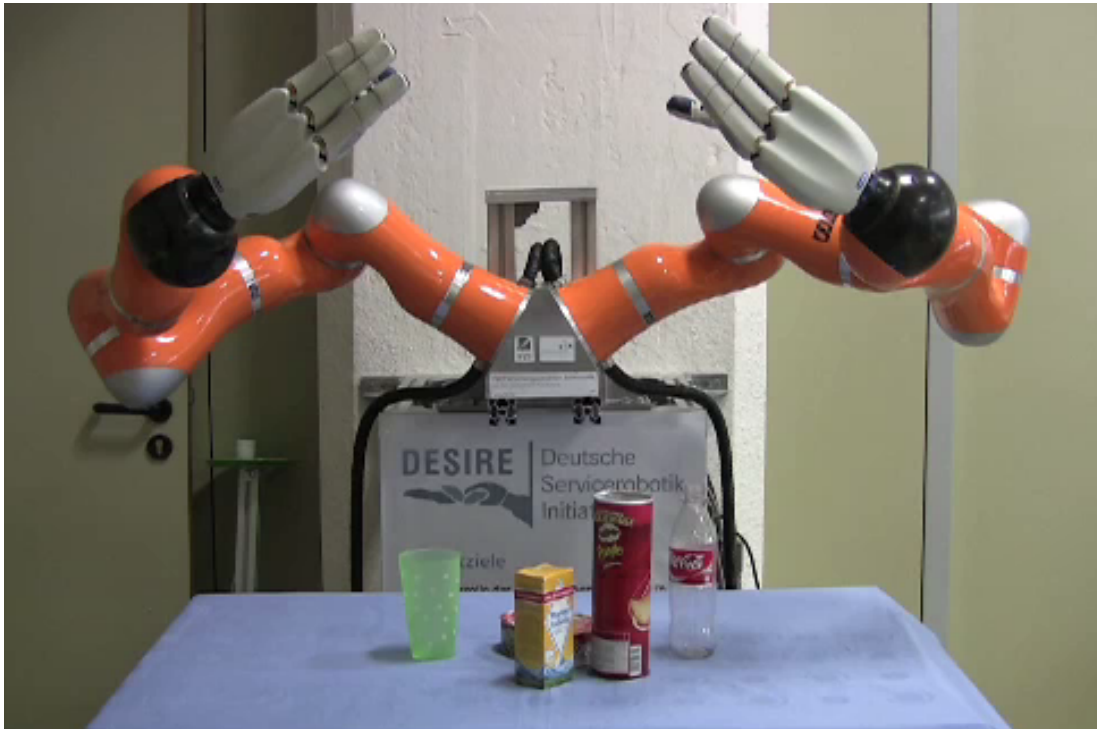


Motivation

Bahnplanung mit

verschiedenen Zielen und Einschränkungen





Inhalt

- Motivation
- Grundlagen der Bahnplanung
  - Problemstellung
  - Klassifizierung der Verfahren
    - Kartesischer Raum – Gelenkwinkelraum
    - Problemklassen
  - Begriffsbildung
  - Vorgehensweise
- Planungsverfahren
  - Mobile Roboter
  - Manipulatoren
  - Robotergreifer

Grundlagen der Bahnplanung

- **Problemstellung**
  - Bewegungen eines Roboters werden als Zustandsänderungen über der Zeit (Trajektorie) relativ zu einem stationären Koordinatensystem (kartesischer Raum, Gelenkwinkelraum) aufgefasst.
- Gegeben
  - S_{Start} : Zustand zum Startzeitpunkt
  - S_{Ziel} : Zustand zum Zielzeitpunkt
- Gesucht
  - S_i : Zwischenzustände (Stützpunkte), damit die Trajektorie “glatt” und stetig wird
- Bedingungen
  - Gütekriterien, Neben- und Randbedingungen sowie Zwangsbedingungen

Grundlagen der Bahnplanung

Klassifizierung der Verfahren

- Bahnplanungsverfahren können in unterschiedlichen **Zustandsräumen** erfolgen
  - Gelenkwinkelzustandsraum (Konfigurationsraum)
  - euklidischer Raum (Arbeitsraum, 3D / 6D)
  - Sensorzustandsraum
  - Objektzustandsraum
- Unterschieden werden noch Bahnplanungsverfahren nach **Art des Roboters**
  - Bahnplanung für mobile Roboter
  - Bahnplanung für Laufmaschinen und anthropomorphe Systeme
  - Bahnplanung für Manipulatoren
  - Greif- und Montageplanung

Grundlagen der Bahnplanung

Problemklassen

- **Klasse a)**

bekannt: vollständiges Umweltmodell, sowie
vollständige Neben-, Rand- und Zwangsbedingungen
gesucht: Kollisionsfreie Bahn vom Start- zum Zielzustand

- **Klasse b)**

bekannt: unvollständiges Umweltmodell, sowie
unvollständige Neben-, Rand- und Zwangsbedingungen
gesucht: Kollisionsfreie Bahn vom Start- zum Zielzustand
Problem: Kollision mit unbekannten Objekten

Grundlagen der Bahnplanung

Problemklassen

- **Klasse c)**

  bekannt: zeitvariantes Umweltmodell (bewegliche Hindernisse)
  gesucht: Kollisionsfreie Bahn vom Start- zum Zielzustand
  Problem: Hindernisse in Ort und Zeit variant

- **Klasse d)**

  bekannt: kein Umweltmodell
  gesucht: Kollisionsfreie Bahn vom Start- zum Zielzustand
  Problem: Kartografieren

- **Klasse e)**

  bekannt: zeitvariantes Umweltmodell
  gesucht: Bahn zu einem bewegliche Ziel (Rendezvous-Problem)
  Problem: Zielzustand in Ort und Zeit beweglich

Grundlagen der Bahnplanung

Definitionen (1)

- **Konfiguration**

  Eine **Konfiguration q** beschreibt den Zustand eines Roboters A
  - im euklidischen Raum durch seine Lage und Orientierung
  - im Gelenkwinkelzustandsraum durch die Werte der Gelenke

- **Konfigurationsraum**

  Der **Konfigurationsraum** (*configuration space*) $\mathcal{C}$ des Roboters A ist der Raum aller möglichen Konfigurationen von A

- **Weg**

  Ein **Weg** für den Roboter A von der Konfiguration q_{Start} zur Konfiguration q_{Ziel} ist eine stetige Abbildung

$$\tau: [0,1] \rightarrow \mathcal{C}$$

mit $\tau(0) = q_{Start}$
$\tau(1) = q_{Ziel}$

Grundlagen der Bahnplanung

Definitionen (2)

- **Arbeitsraumhindernis**

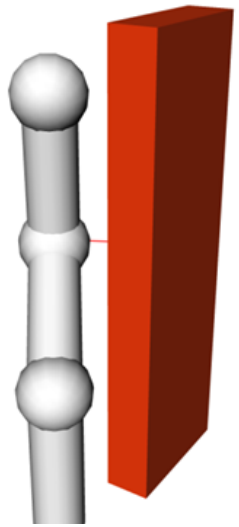
  Ein **Arbeitsraumhindernis H** ist der Raum, welcher von einem Objekt im Arbeitsraum eingenommen wird.

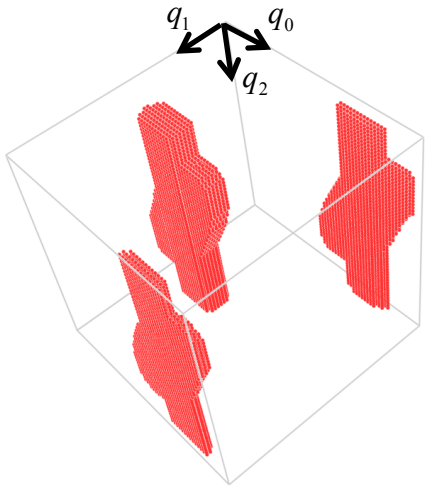
- **Konfigurationsraumhindernis**

  Ein **Konfigurationsraumhindernis** $\mathcal{C}_{\mathbf{H}}$ ist die Menge aller Punkte des Konfigurationsraumes, welche innerhalb eines Arbeitsraumhindernisses H_i liegen.

- **Hindernisraum**

  Der **Hindernisraum** ist die Menge aller Konfigurationsraumhindernisse: $\mathcal{C}_{obst} = \cup\ \mathcal{C}_{Hi}$





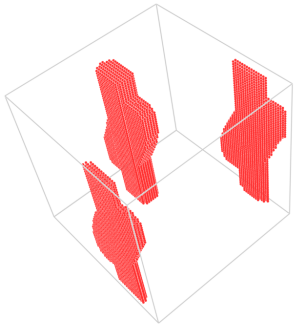
Grundlagen der Bahnplanung

Definitionen (3)

- **Freiraum**

Der **Freiraum** ist die Menge aller Punkte aus $\mathcal{C}$, welche nicht im Hindernisraum liegen

$$\mathcal{C}_{\text{free}} = \{ q \in \mathcal{C} \mid q \notin \mathcal{C}_{\text{obst}} \} = \mathcal{C} \setminus \mathcal{C}_{\text{obst}}$$



Aufwand für die Berechnung des Freiraums:

$O(m^n)$

mit n: Anzahl der Bewegungsfreiheitsgrade des Roboters
m: Anzahl der Hindernisse

→ sehr hoch, für komplexe Kinematiken (n>3) kann $\mathcal{C}_{free}$ nicht effizient berechnet werden

→ Einsatz von approximativen Verfahren zur vereinfachten Repräsentation des Freiraumes.

Inhalt

- Motivation
- Grundlagen der Bahnplanung
- Planungsverfahren für
  - Mobile Roboter
    - Voronoi-Diagramme
    - Sichtgraphen
    - Zellzerlegung
    - Potentialfelder
  - Manipulatoren
  - Robotergreifer

Planungsverfahren

Definitionen

- Umweltmodellierung:
  - **Exakt**: Beispielsweise über CSG (constructed-solid-geometry), in Form einer algebraischen Beschreibung
  - **Approximiert**: Die Umwelt wird durch Näherungen beschrieben (Kuben, verallgemeinerte Zylinder, Polyeder,....)

- Planungsalgorithmen:
  - **Vollständige** Verfahren: Die Algorithmen liefern immer eine korrekte Lösung und können ermitteln, ob keine Lösung existiert.
  - **Probabilistisch vollständige** Verfahren:
    Falls eine Lösung existiert konvergiert die Wahrscheinlichkeit, dass eine Lösung gefunden wird, bei fortschreitender Zeit gegen 1.
    Existiert keine Lösung, kann dies nicht ermittelt werden.
    -> Randomisierte Algorithmen

Planungsverfahren: Mobile Roboter

Beispiel: Straßenkarte

Gegeben:

- 2-dim. Weltmodell
- Start und Ziel

Aufgabe:

Suche günstigste Verbindung von Start zu Ziel

Lösung:

1) Konstruiere ein Netz von Wegen in $\mathcal{C}_{\text{free}}$
2) Bilde q_{Start} und q_{Ziel} auf ab
   $\rightarrow q'_{\text{Start}} = \mathbb{W}(q_{\text{Start}}),\ q'_{\text{Ziel}} = \mathbb{W}(q_{\text{Ziel}})$
3) Suche einen Weg, der q'_{Start} mit q'_{Ziel} verbindet

Planungsverfahren: Mobile Roboter

Teilaufgaben

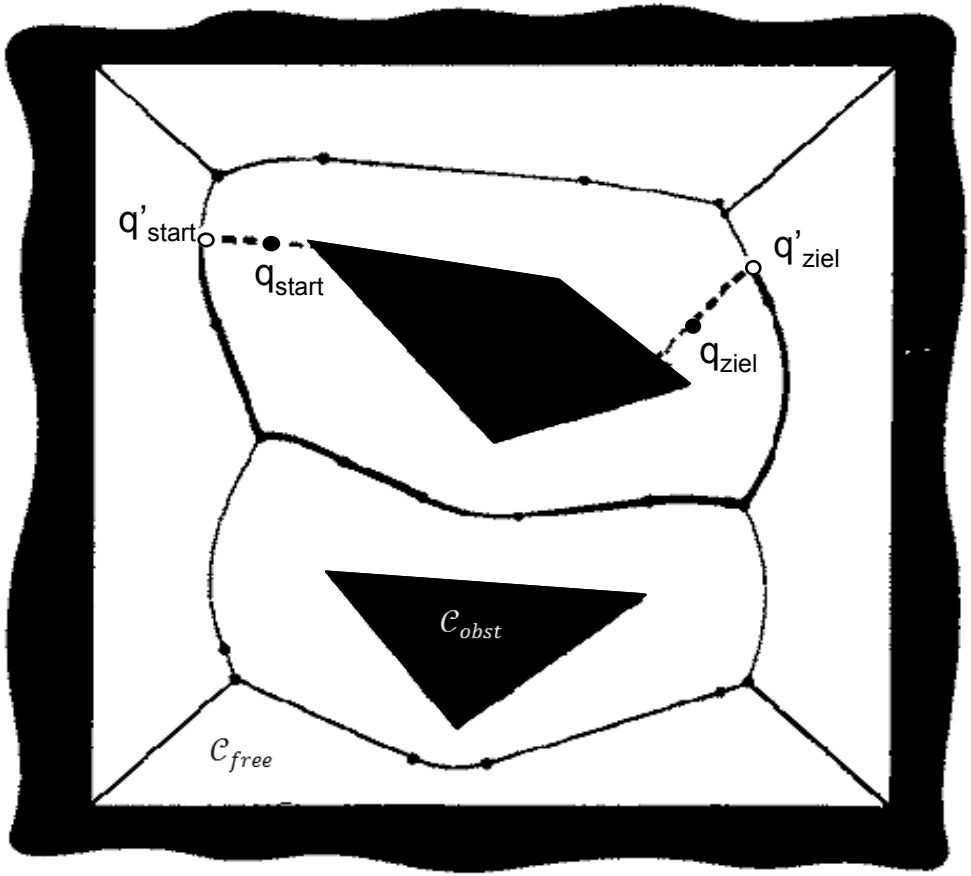
1. Wegkonstruktion mit
   - Retraktionsverfahren, z.B. Voronoi-Diagramm
   - Sichtgraphen

2. Suche im Netz, z.B. mit
   - A*-Algorithmus (Baumsuche)
   - euklidischer Abstand
   - Potentialfeld

Planungsverfahren: Mobile Roboter

Beispiel: Voronoi-Diagramm

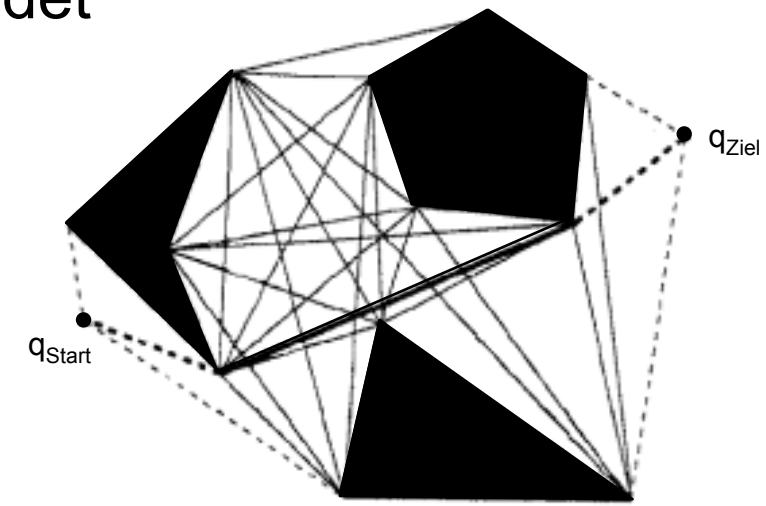
p: mittlerer Abstand



Planungsverfahren: Mobile Roboter

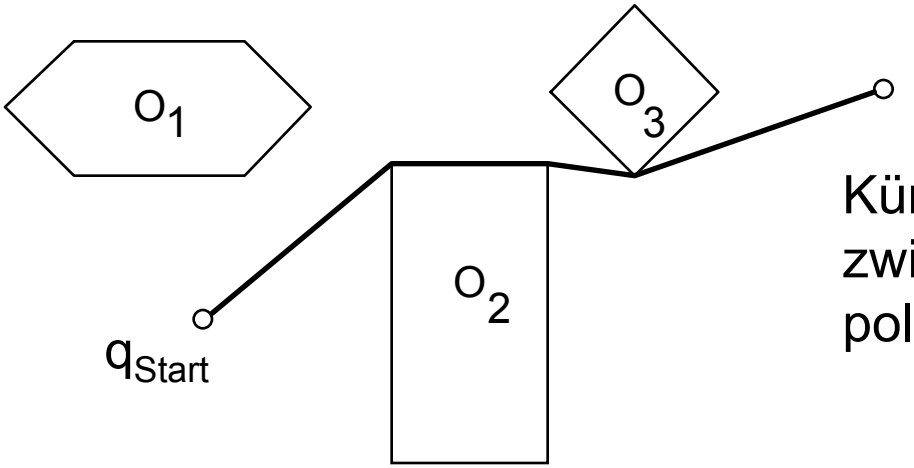
Sichtgraphen

- Konstruktion:
  - Verbinde jedes Paar von Eckpunkten auf dem Rand von $\mathcal{C}_{\text{free}}$ durch ein gerades Liniensegment, wenn das Segment kein Hindernis schneidet
  - Verbinde q_{Start} und q_{Ziel} analog

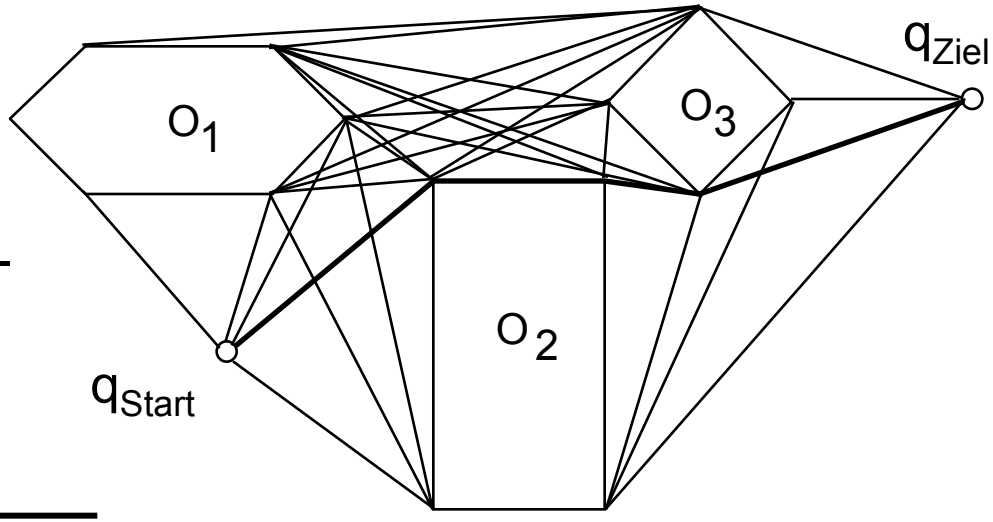


Planungsverfahren: Mobile Roboter

Beispiele für Sichtgraphen



Kürzester kollisionsfreier Weg zwischen konvexen polygonalen Hindernissen



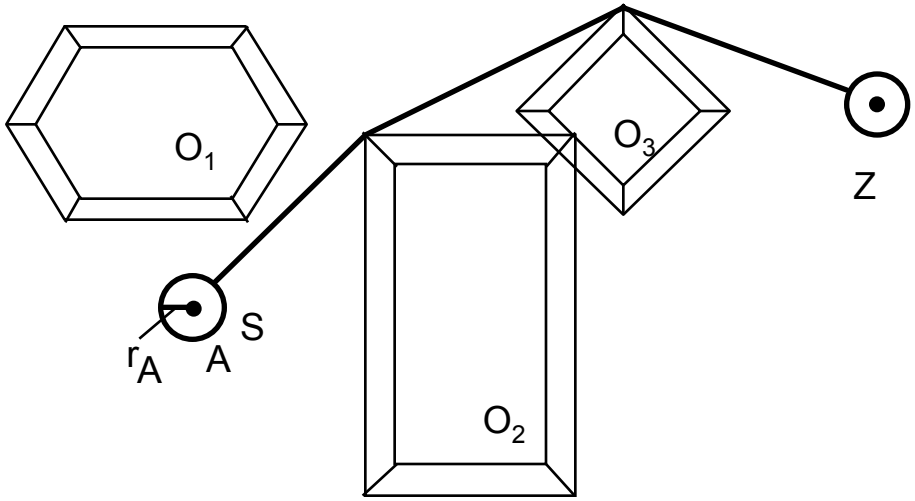
Methode des Sichtgraph-Algorithmus

Planungsverfahren: Mobile Roboter

- Wege sind nur “halbfrei” (nicht kollisionsfrei), da Hindernis-kanten auch Wegsegmente sein können. Abhilfe durch Erweiterung der Hindernisse.
- Wenn ein Weg gefunden ist, ist es auch der kürzeste Weg
- Methode ist exakt, wenn Roboter nur 2 translatorische Freiheitsgrade hat und sowohl Roboter als auch Hindernisse durch konvexe Polygone dargestellt werden können
- Methode auch im ^{3} anwendbar, jedoch sind die gefundenen Wege i.A. keine kürzesten Wege mehr

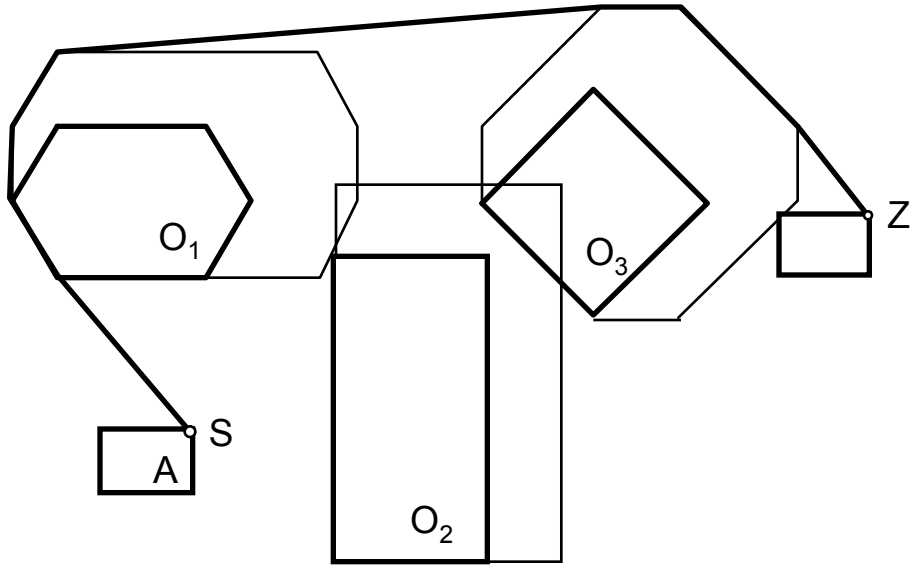
Planungsverfahren: Mobile Roboter

Erweiterung der Hindernisse



Vergrößern von Hindernissen durch einen Kreis

Vergrößern von Hindernissen durch ein Rechteck



Planungsverfahren: Mobile Roboter

Zellenzerlegung

Vorgehen:

1. Zerlege $\mathcal{C}_{\text{free}}$ in solche Zellen, so dass ein Weg zwischen 2 Konfigurationen innerhalb einer Zelle leicht zu finden ist
2. Stelle die Nachbarschaft (Adjadenz) Relation in einem Graphen dar
3. Suche den optimalen Weg von q_{Start} nach q_{Ziel} in dem Graphen

Es gibt 2 Zerlegungsarten:

- exakte Zerlegungsart
- approximative Zerlegungsart

Exakte Zellenzerlegung

Zerlegung des Freiraumes $\mathcal{C}_{free}$ in Zellen Z_i, so dass:

- die Zellen sich nicht überlappen

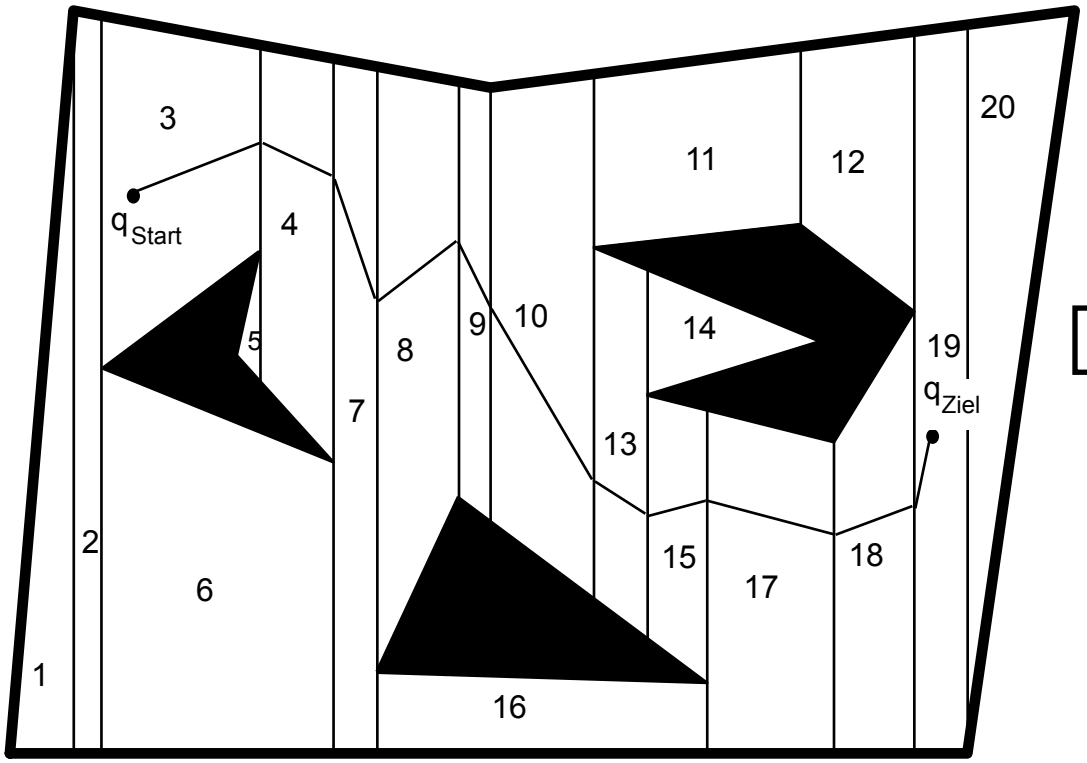
$$\forall i,k \quad i \neq k : Z_i \cap Z_k = \emptyset$$

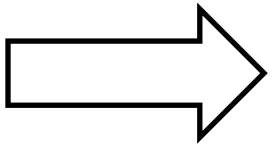
- die Vereinigungsmenge aller Z_i ist $\mathcal{C}_{free}$

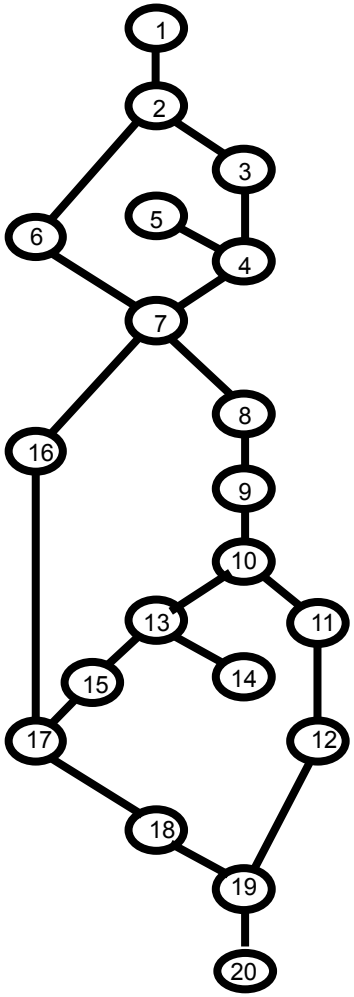
$$\bigcup_{i=1}^{n} Z_i = \mathcal{C}_{free}$$

Planungsverfahren: Mobile Roboter

Beispiel







Exakte Zellenzerlegung mit Line-Sweep

Verbindungsgraph

Planungsverfahren: Mobile Roboter

Approximative Zellenzerlegung

Vorgehen:

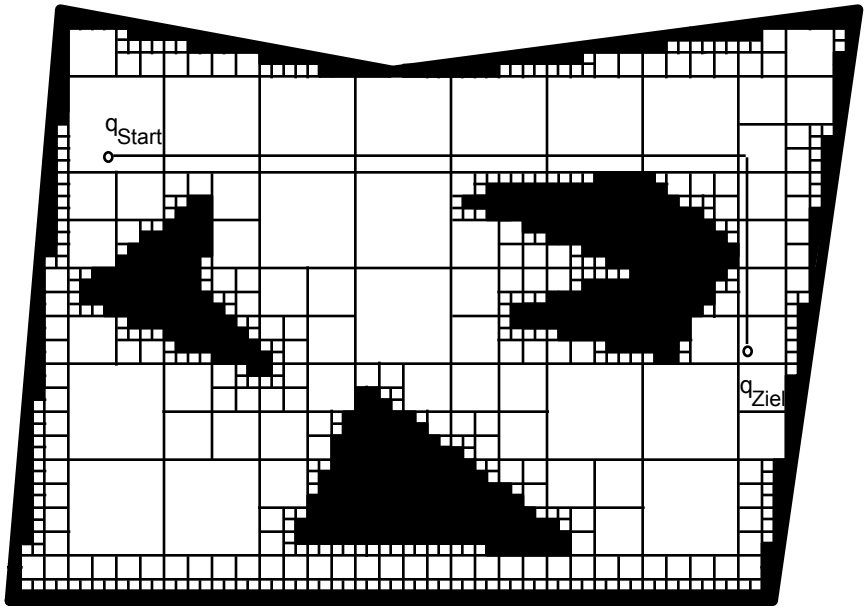
1. Zerlege den Freiraum $\mathcal{C}_{free}$ in Zellen mit vordefinierter Form (z.B. rechteckig)
2. Wenn eine Zelle nicht vollständig in $\mathcal{C}_{free}$ liegt, verringere die Größe und zerlege die Zelle weiter (z.B. Quadtree)
3. Wende diesen Schritt nur bis zu einer Minimalgröße der Zellen an

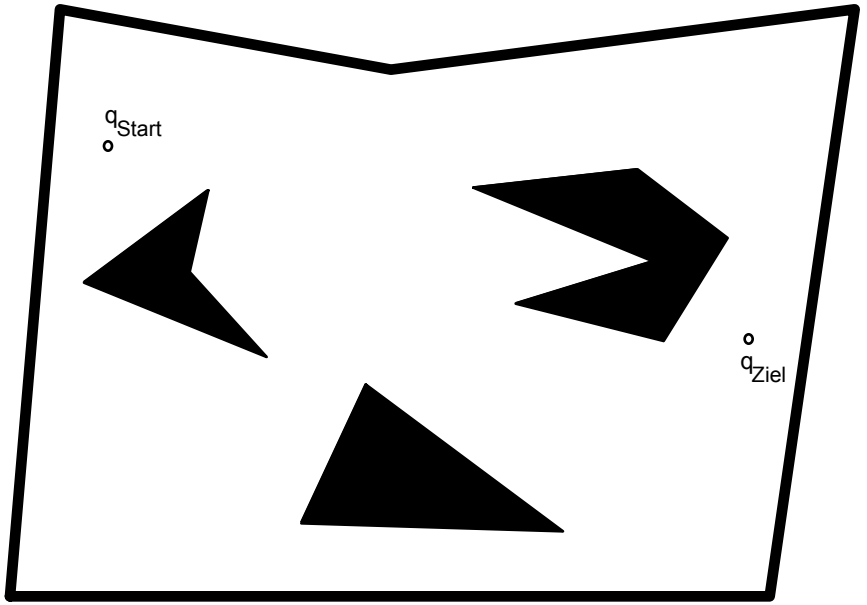
Vorteil: einfache Zerlegung ➔ einfachere Wegsuche

Nachteil: der Freiraum kann i.A. nur annähernd beschrieben werden

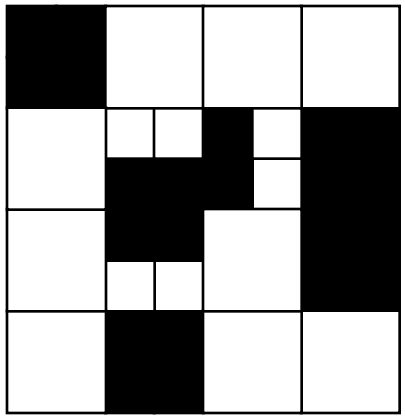
Planungsverfahren: Mobile Roboter

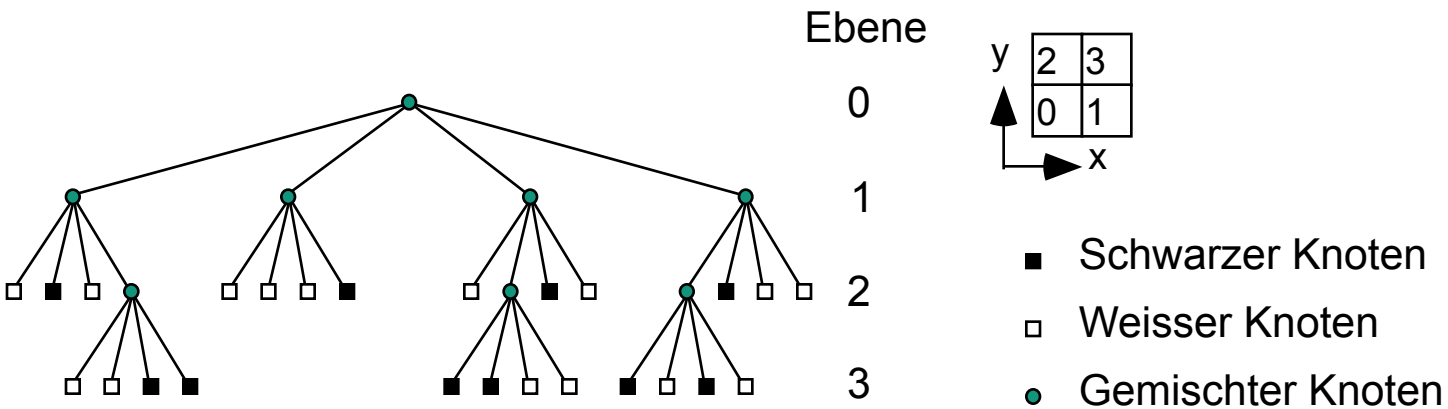
Approximative Zellenzerlegung





Beispiel: Quadtree mit Baumsuche (1)

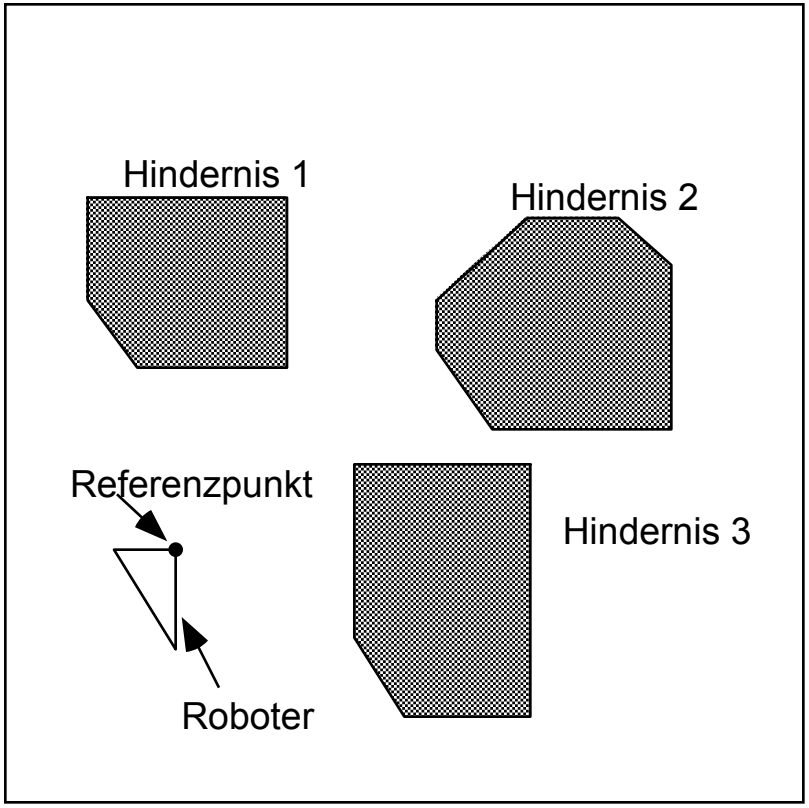




Planungsverfahren: Mobile Roboter

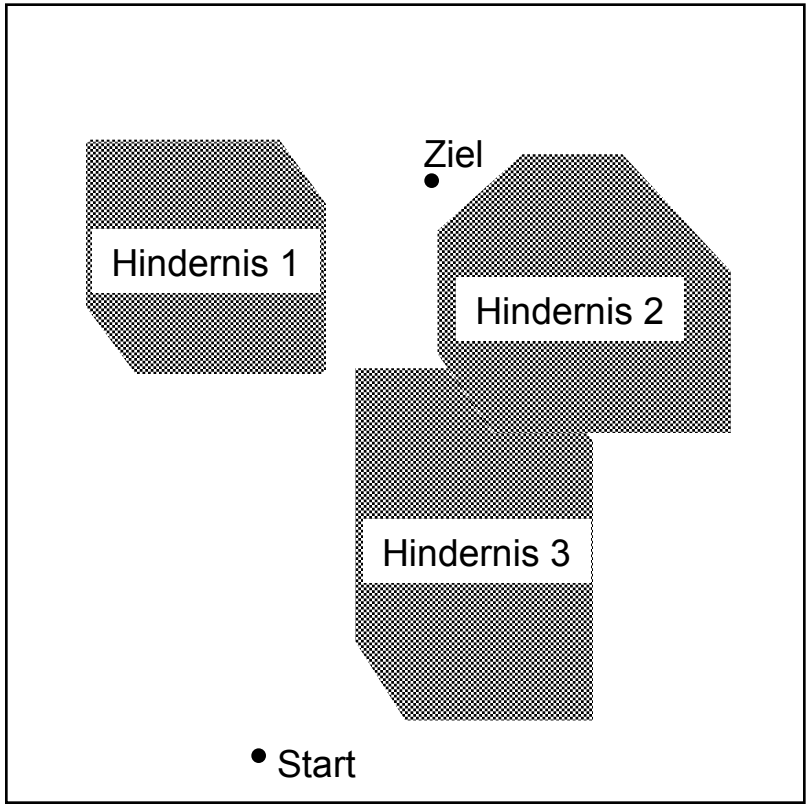
Beispiel: Quadtree mit Baumsuche (2)

Arbeitsraum



Arbeitsraum eines Roboters mit Hindernissen

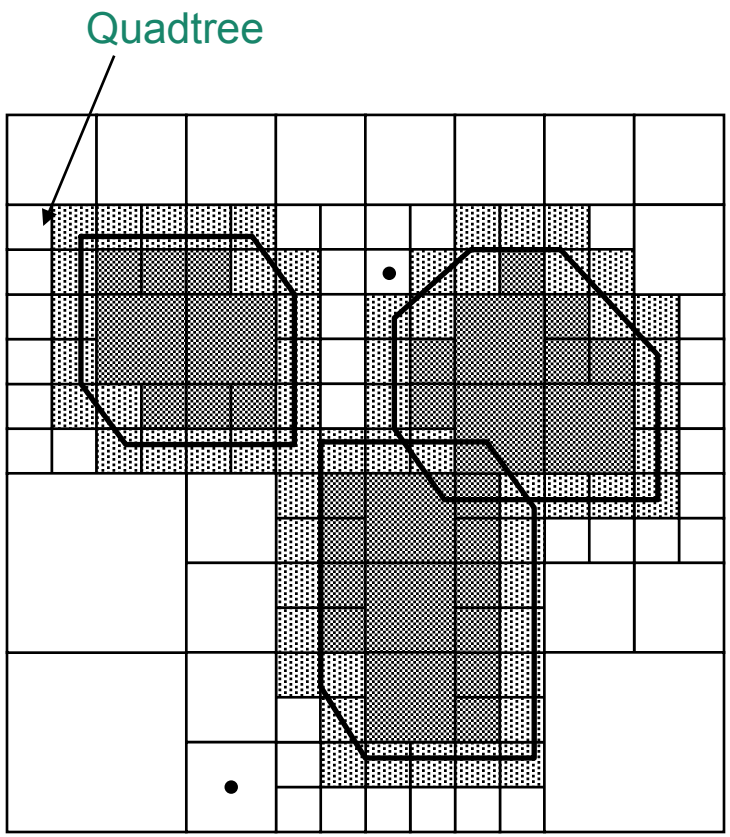
Konfigurationsraum



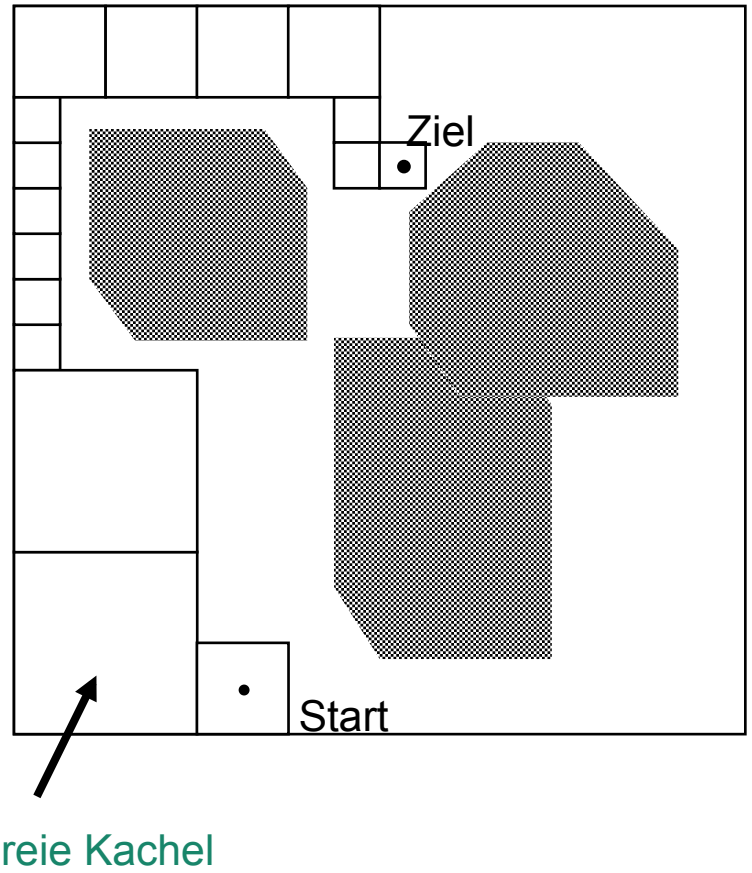
Konfigurationsraum für das vorliegende Robotersystem

Planungsverfahren: Mobile Roboter

Beispiel: Quadtree mit Baumsuche (3)



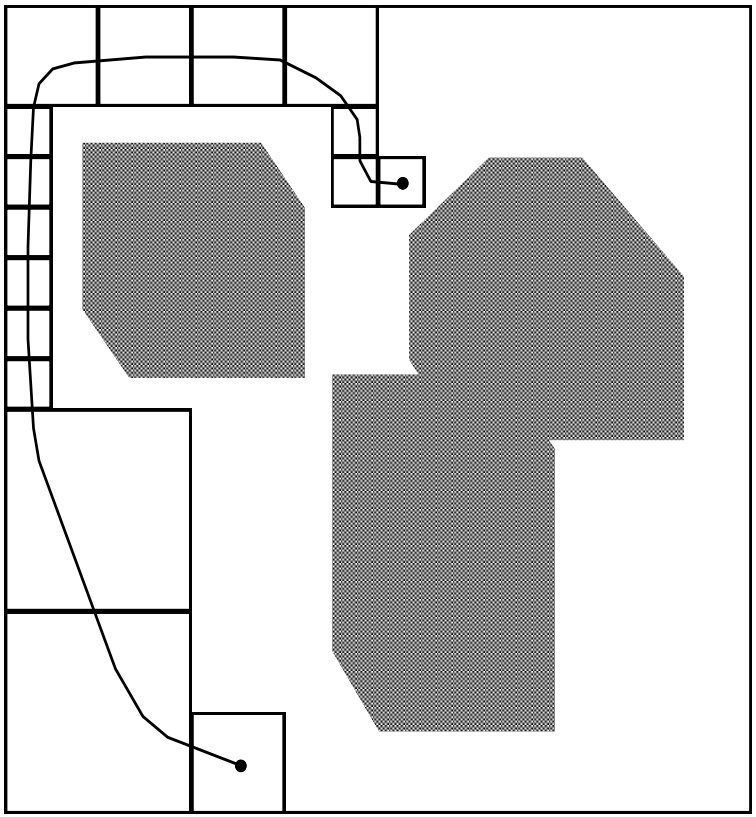
Zerlegung des Konfigurationsraums in Kachelzonen



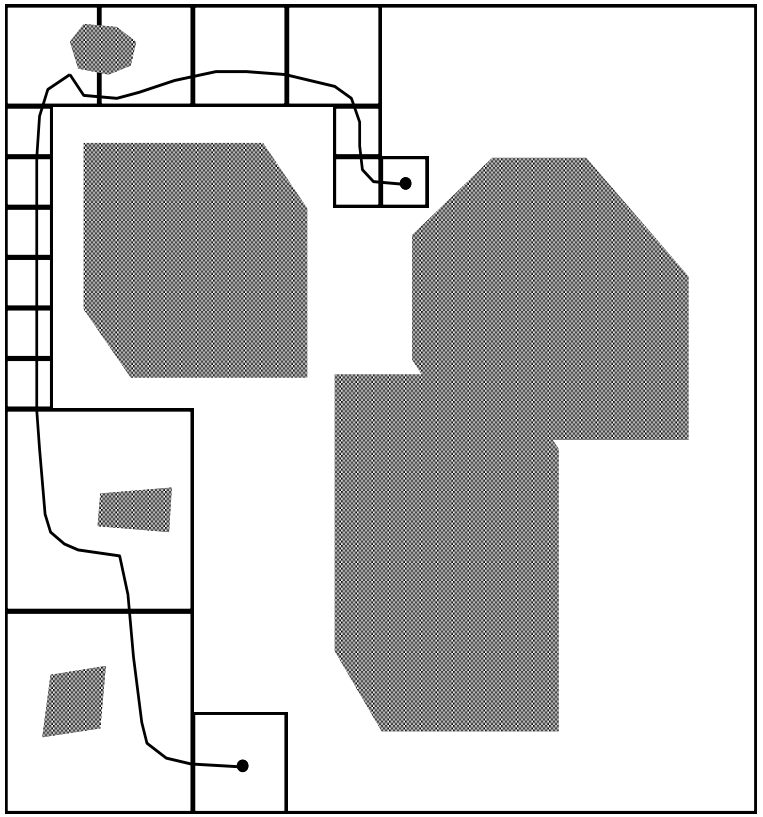
Folge von freien Kacheln vom Start- zum Zielpunkt

Planungsverfahren: Mobile Roboter

Beispiel: Quadtree mit Baumsuche (4)



Hindernisfreie Verfahrroute



Ausweichmanöver um lokale Hindernisse

Planungsverfahren: Mobile Roboter

Potentialfelder (1)

- Der Roboter bewegt sich unter dem Einfluss von Kräften, welche ein Potentialfeld auf ihn ausübt
- **Definition:**
  - Ein Potentialfeld U ist eine Skalarfunktion über den Freiraum

$$U: \mathcal{C}_{free} \rightarrow \mathbb{R}$$

  - Die Kraft in einem Punkt q des Potentialfeldes ist der negative Gradient in diesem Punkt

Planungsverfahren: Mobile Roboter

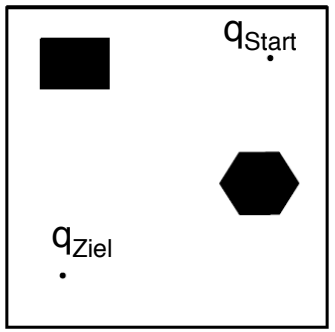
Potentialfelder (2)

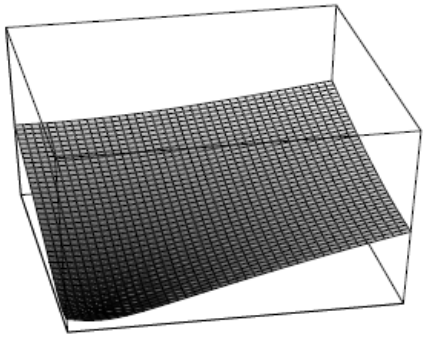
- Abstoßungspotenzial
  - Hindernisse erzeugen ein abstoßendes Potential
  - In großem Abstand zu Hindernissen soll der Roboter nicht beeinflusst werden
- Anziehungspotential
  - Es soll nur ein Minimum in q_{Ziel} geben

Planungsverfahren: Mobile Roboter

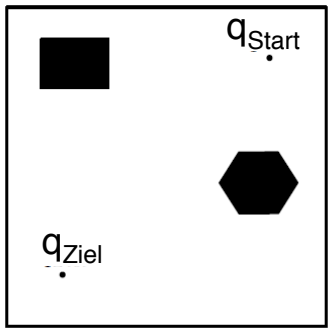
Beispiel: Potentialfelder (1)

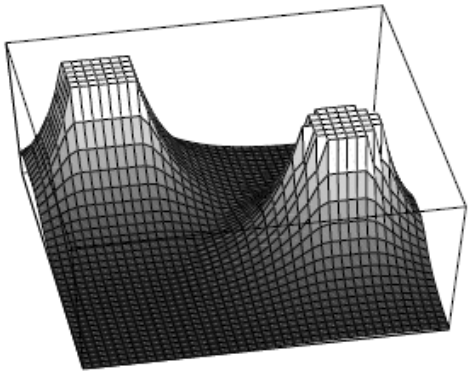
- Die Zielstellung q_{Ziel} hat ein Anziehungspotential U_{an}





- Die Hindernisräume $\mathcal{C}_{Hi}$ haben ein Abstoßungspotential U_{ab}





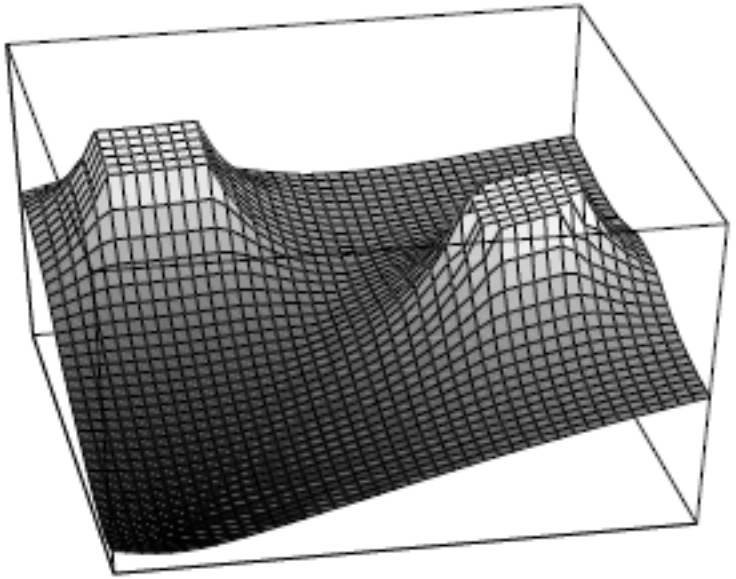
Planungsverfahren: Mobile Roboter

Beispiel: Potentialfelder (2)

- Die Summe der einwirkenden Kräfte bestimmt die Richtung der Bewegung.
- Für das Potentialfeld gilt: $U(q) = U_{An}(q) + U_{Ab}(q)$

- Für das Kräftefeld gilt:

$$F(q) = F_{an}(q) + F_{ab}(q)$$



Planungsverfahren: Mobile Roboter

Problem: Potentialfelder

Lokale Minima

Durch Summation von U_{An} und U_{Ab} kann U lokale Minima besitzen. Wenn der Roboter sich in Richtung des negativen Gradienten des Potentialfeldes bewegt, kann er in einem solchen lokalen Minimum "steckenbleiben".

Maßnahmen:

1) U_{An} und U_{Ab} so definieren, dass U kein lokales Minimum außer $q = q_{Ziel}$ hat
2) Im Suchalgorithmus Techniken zur "Flucht" aus dem lokalen Minimum anwenden

Inhalt

- Motivation
- Grundlagen der Bahnplanung
- Planungsverfahren für
  - Mobile Roboter
  - Manipulatoren
    - Probabilistic Roadmaps (PRM)
    - Rapidly-exploring Random Trees (RRT)
  - Robotergreifer

Planungsverfahren: Manipulatoren

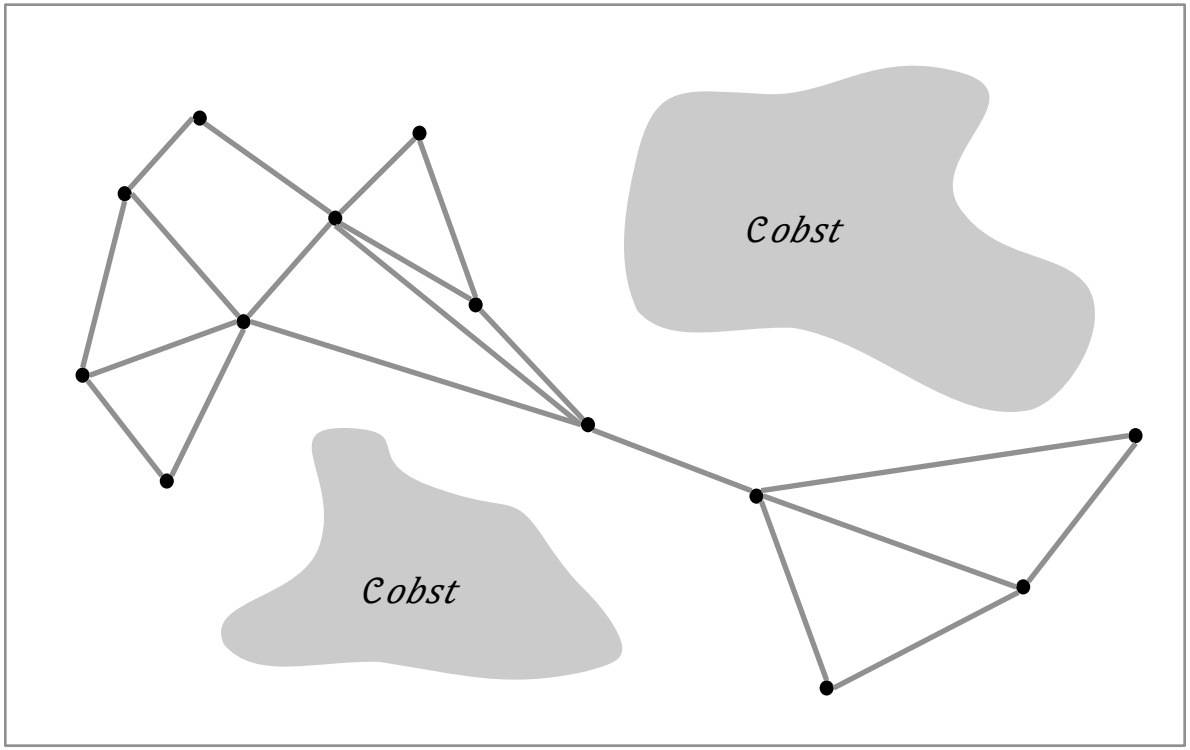
Probabilistic Roadmaps (PRM)

- **Schritt 1**: Vorverarbeitung
  - Erzeugung einer kollisionsfreien Straßenkarte
    → Graph
- **Schritt 2**: Anfrage
  - Verbinde q_{Start} und q_{Ziel} mit der Karte
  - Suche einen Weg durch den Graphen
- Motivation: Approximation des Freiraumes
  -> Effizienter als die Erzeugung einer expliziten Repräsentation des Freiraumes

Planungsverfahren: Manipulatoren

Probabilistic Roadmaps (PRM)

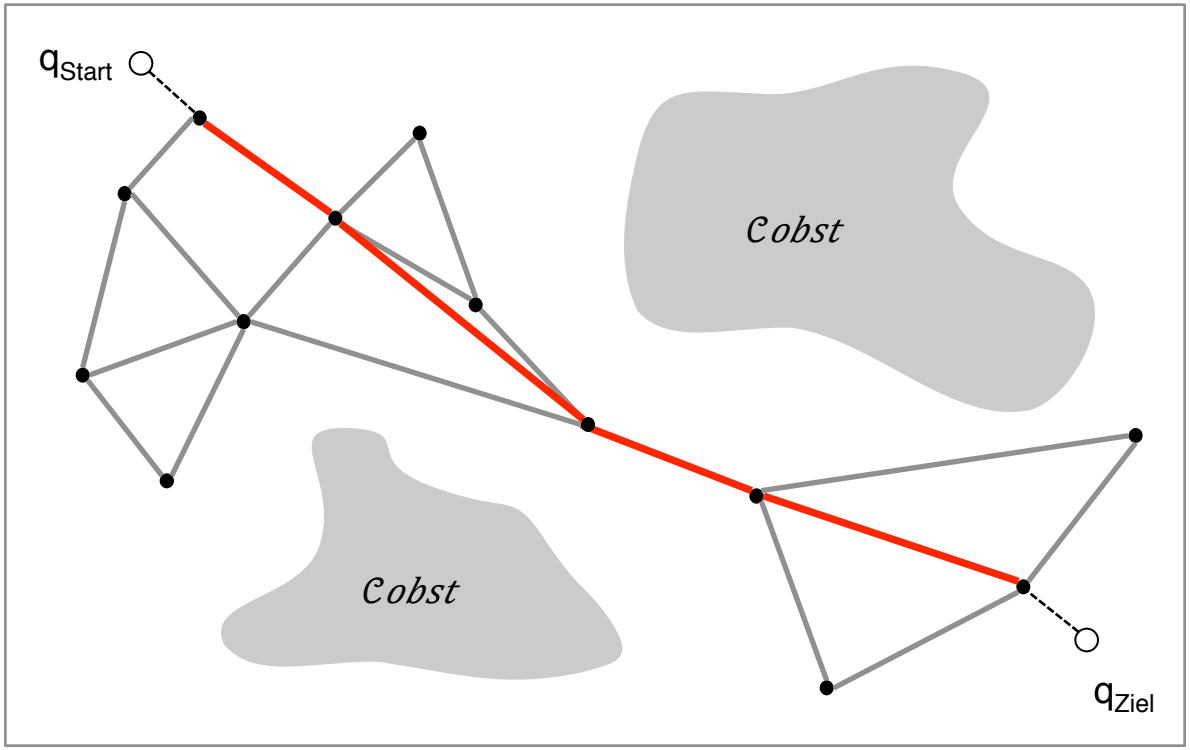
- Vorverarbeitung:
  - Zufällige Erzeugung von kollisionsfreien Stichproben (Sampling)
  - Stichproben werden über kollisionsfreie Pfade miteinander verbunden



Planungsverfahren: Manipulatoren

Probabilistic Roadmaps (PRM)

- Anfrage:
  - Verbinde Start und Ziel mit dem Wegenetz
  - Suche im Graphen



Planungsverfahren: Manipulatoren

RRT: Rapidly-exploring Random Trees (1)

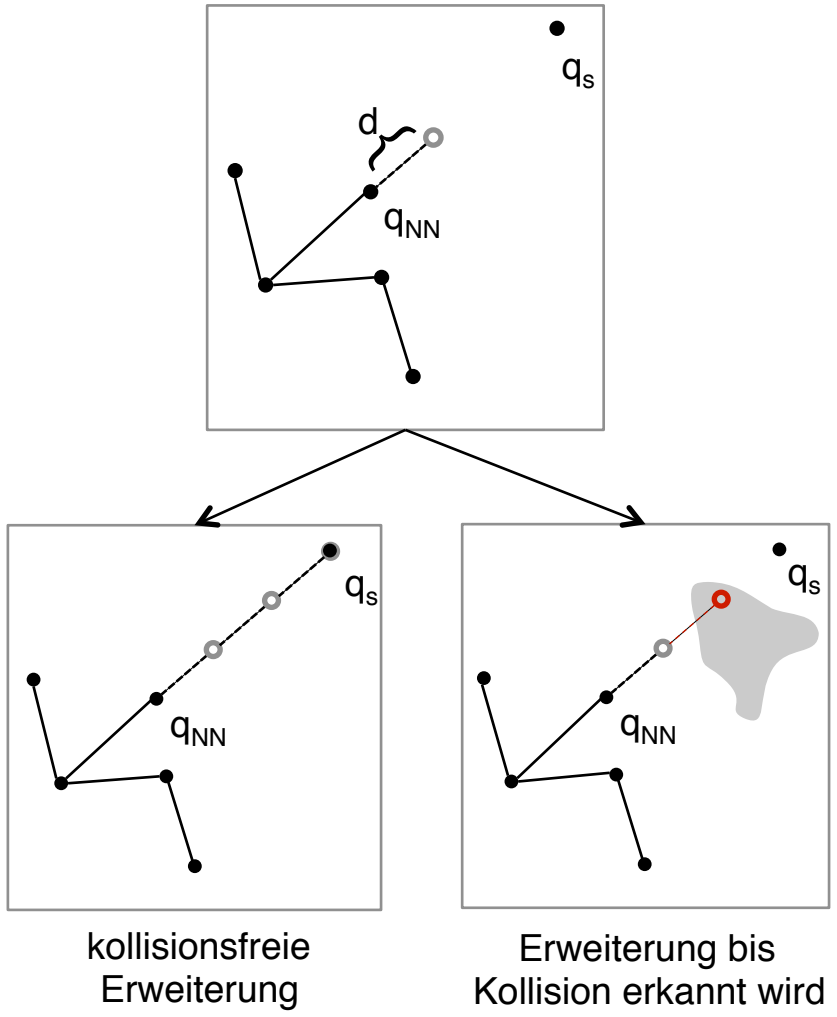
- Algorithmus zur Einmalanfrage (keine Vorverarbeitung nötig)
- Aufbau zweier Bäume zur Approximation des Freiraumes
- Zufallsgesteuerter Algorithmus
- Probabilistisch vollständig
- Der RRT-Ansatz lässt sich effizient für hochdimensionale Problemstellungen einsetzen

Planungsverfahren

RRT: Rapidly-exploring Random Trees (2)

Aufbau der RRT-Bäume:

- Wurzelknoten: q_{Start} bzw. q_{Ziel}
- Zufällige Erzeugung kollisionsfreier Stichproben q_s
- Suche des nächsten Nachbarn q_{NN}
- Erweiterungsschritte in Richtung q_s
  - Mit Schrittweite d
  - bis Kollision erkannt oder q_s erreicht wird



Planungsverfahren: Manipulatoren

RRT: Rapidly-exploring Random Trees (2)

BUILD_RRT(K_{Start},n,ε)

1. T.init(K_{Start})
2. for k = 1 to n
3. K_{Zuf} ← RAND_CONF()
4. K_{Nahe} ← NEAREST_VERTEX(K_{Zuf}, T)
5. K_{Neu} ← EXTEND(K_{Nahe}, K_{Zuf}, ε)
6. if VALID(K_{neu})
7. T.add_vertex(K_{Neu})
8. T.add_edge(K_{Nahe}, K_{Neu})
9. Return T

Kommentare:

Neuer Baum mit Startkonfiguration in der Wurzel

Gleichverteilt zufällige Erzeugung einer Konfiguration

Bestimmung des nächsten Knotens

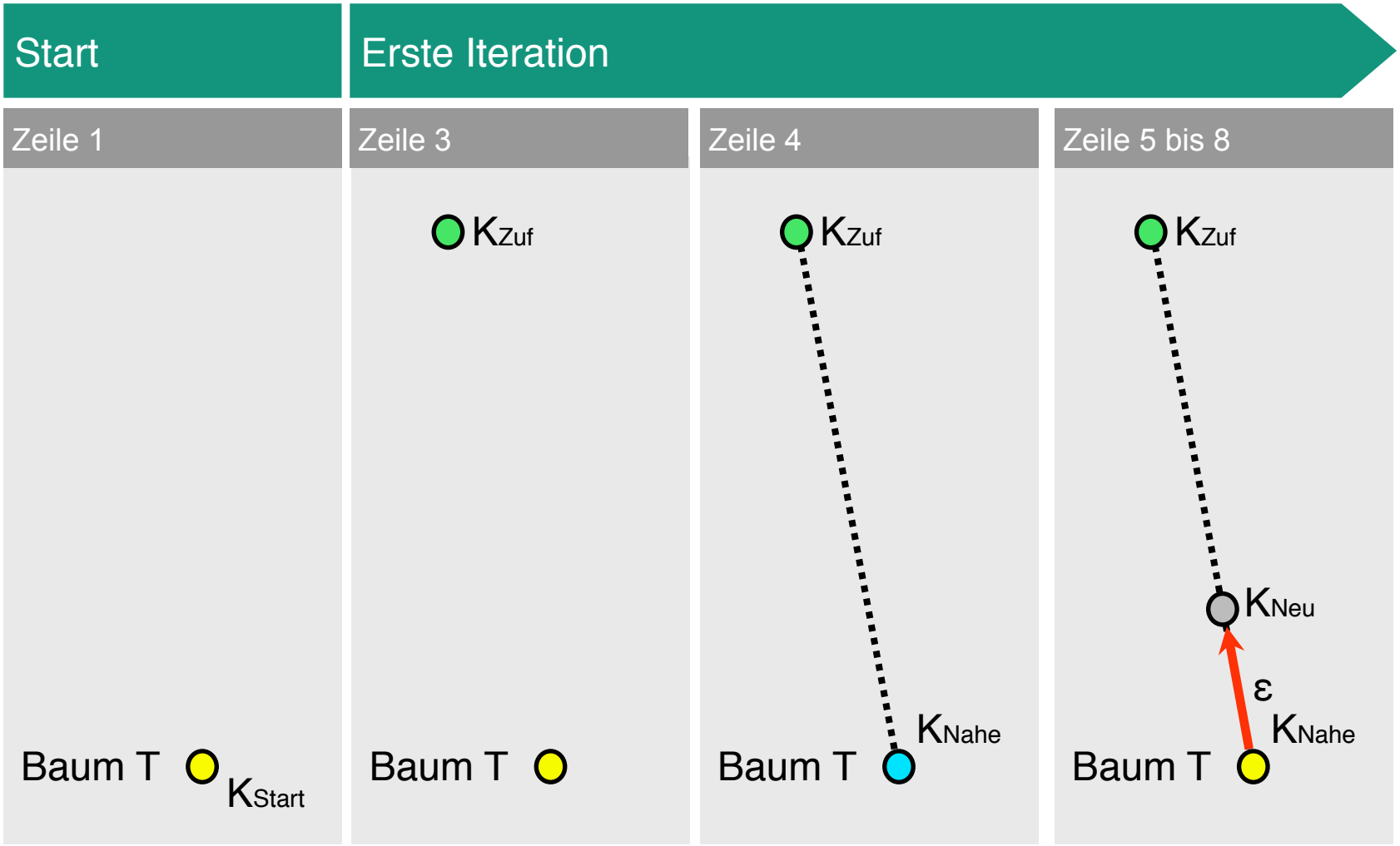
Erzeugung einer neuen Konfiguration

Falls gültige Konfiguration

Hinzufügen eines Knotens

und einer Kante

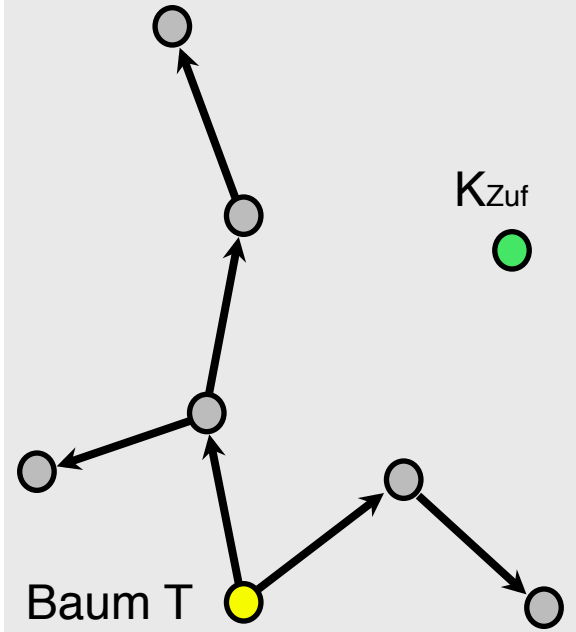
Planungsverfahren: Manipulatoren



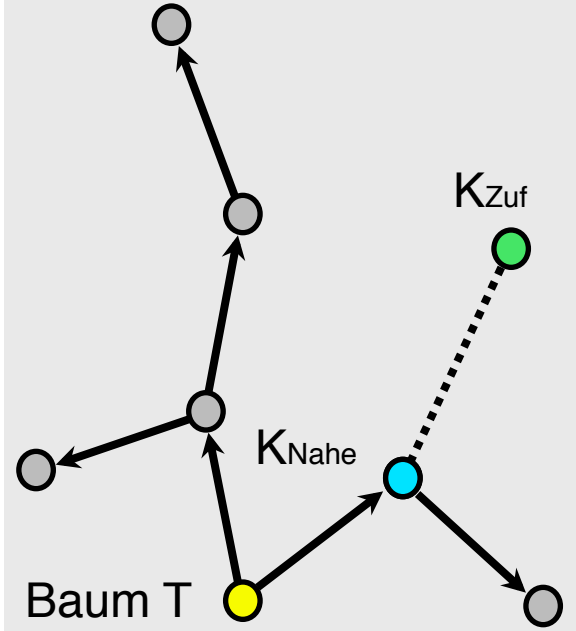
Planungsverfahren: Manipulatoren

7te Iteration

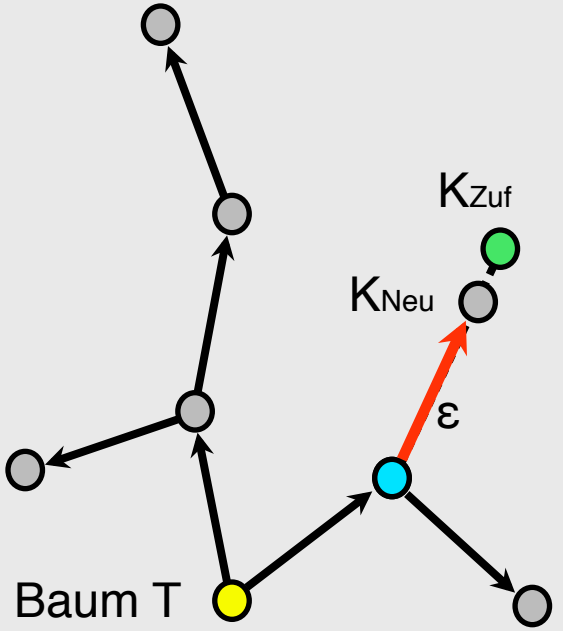
Zeile 3



Zeile 4



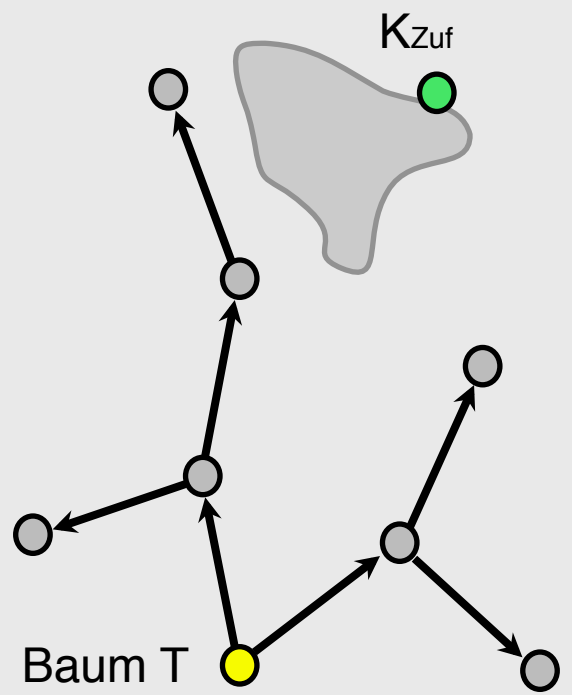
Zeile 5 bis 8



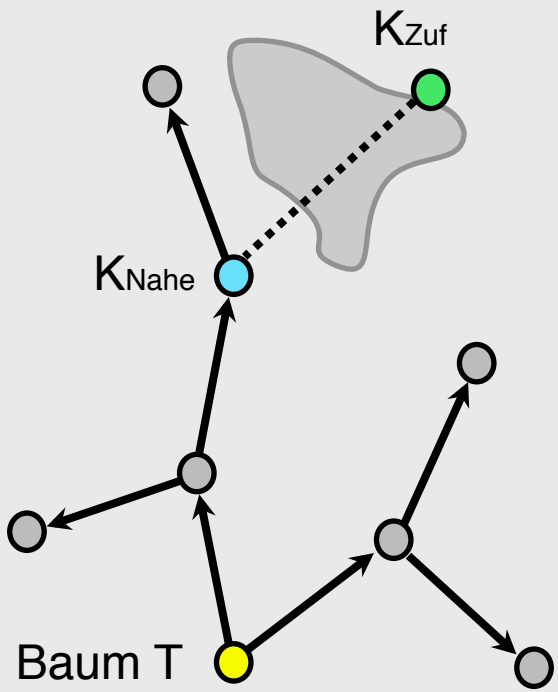
Planungsverfahren: Manipulatoren

8te Iteration: Hindernis?

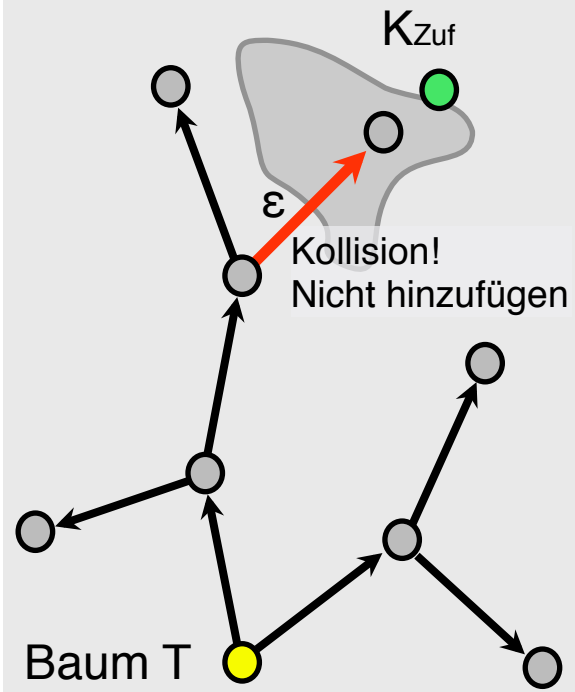
Zeile 3



Zeile 4



Zeile 5 bis 8

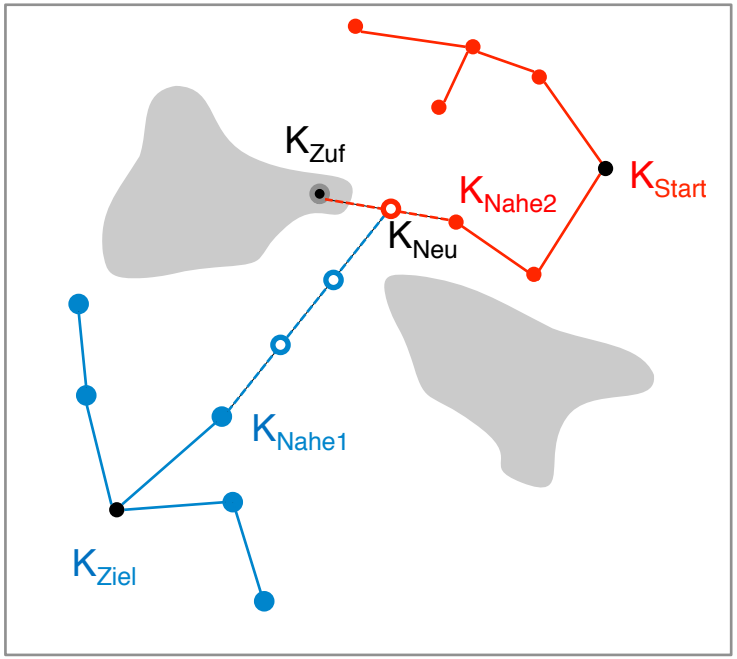


Planungsverfahren: Manipulatoren

RRT: Rapidly-exploring Random Trees (3)

Zielgerichtetes Planen:

- Aufbau von zwei Bäumen T_1 und T_2
  - Wurzelknoten: K_{Start} bzw. K_{Ziel}
- Stichprobe K_{Zuf} wird zur Erweiterung von T_1 verwendet mit Ergebnis K_{Neu}
- Erweiterung von T_2 mit K_{Neu} bis K_{Neu} erreicht oder !VALID(K_{Neu})
- Falls K_{Neu} von T_2 erreicht wird, ist ein gültiger Pfad gefunden
- Sonst: Vertausche Rolle von T_1 und T_2



rot: T_1, blau: T_2

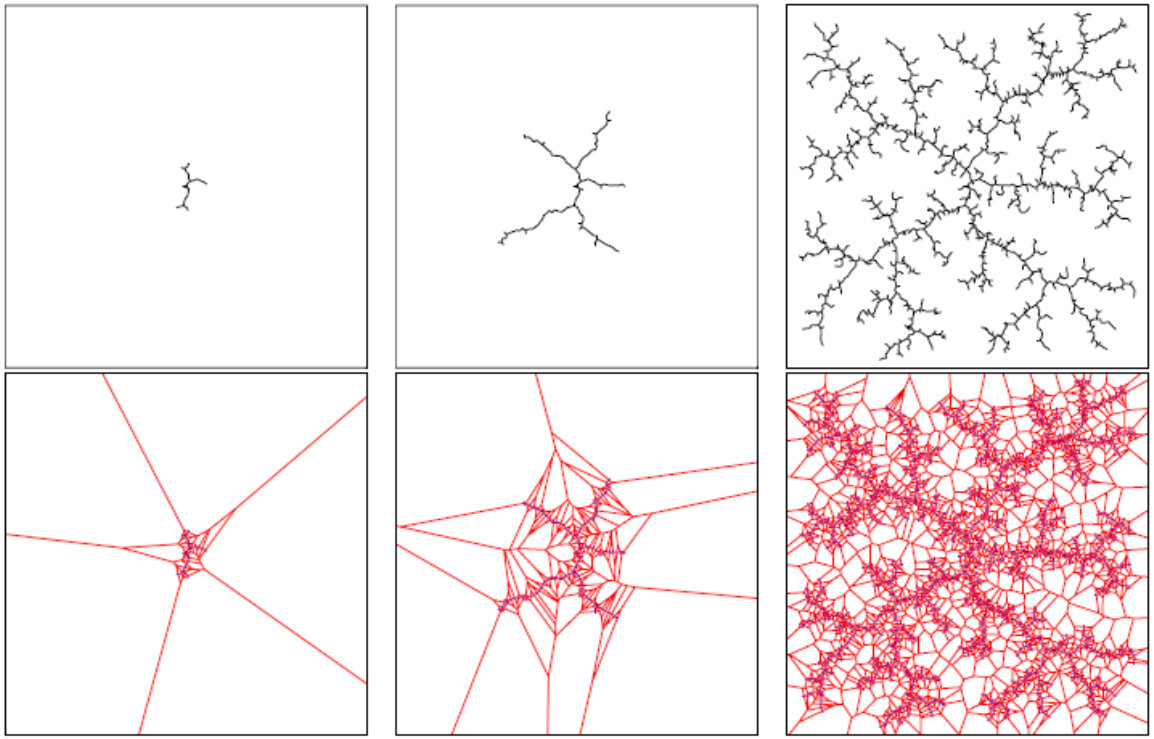
Planungsverfahren: Manipulatoren

RRT: Rapidly-exploring Random Trees (4)

- Erweiterung eines Baumes

RRT-Baum

Voronoi-Regionen

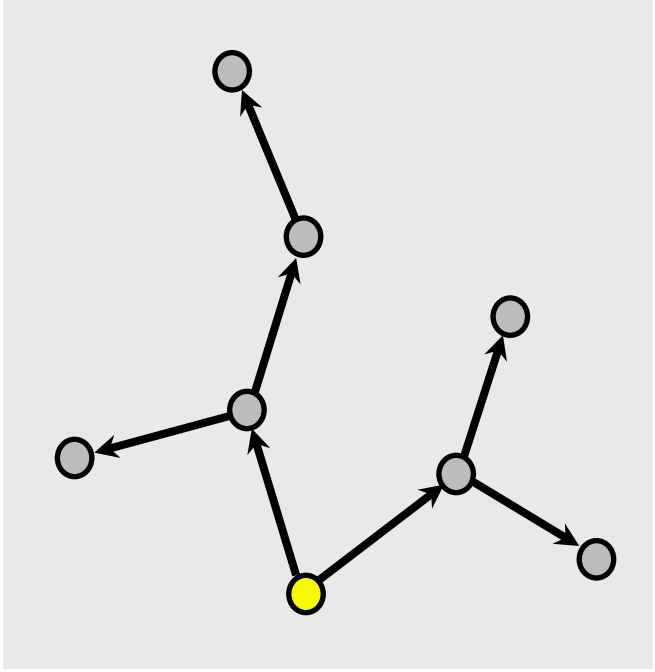


Planungsverfahren: Manipulatoren

RRT: Rapidly-exploring Random Trees (5)

- Welcher Knoten wird am wahrscheinlichsten erweitert?

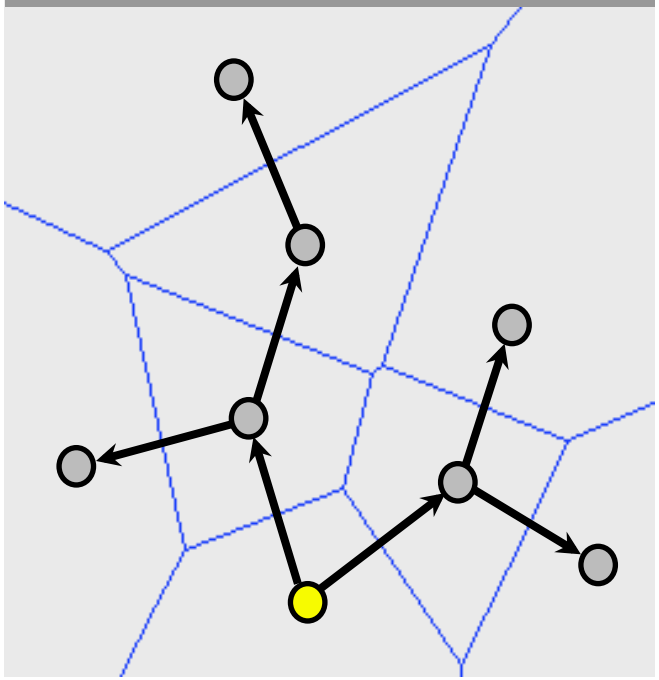
Baum nach 8. Iteration



Knoten mit größter Voronoi-Region (Voronoi-Bias)

Voronoi-Gebiete am Anfang im Randbereich groß
→ Rasche Exploration
→ Verfeinerung

Baum nach 8. Iteration

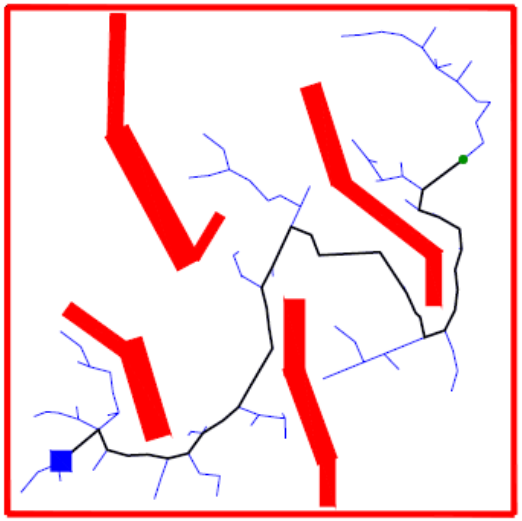


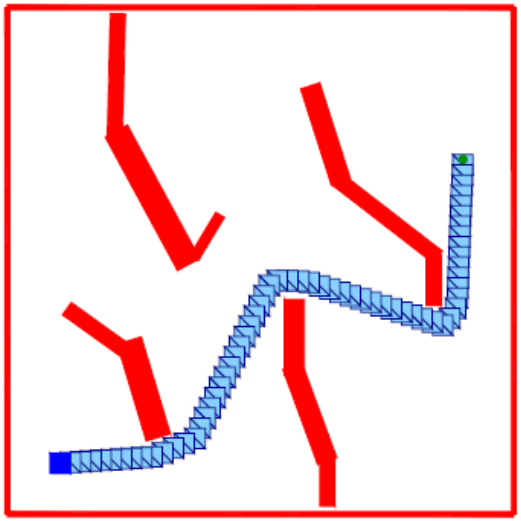
Die Wahrscheinlichkeit, dass ein Knoten erweitert wird, ist proportional zur Größe der entsprechenden Voronoi-Region

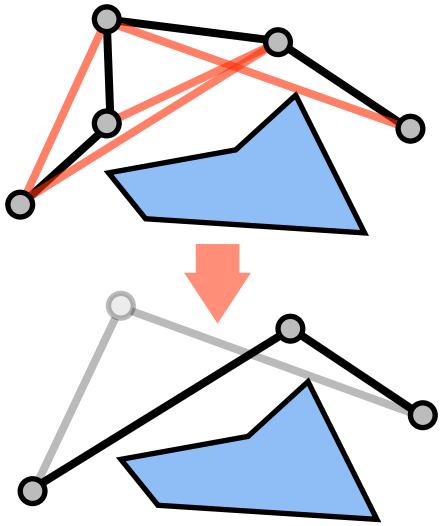
Planungsverfahren: Manipulatoren

Glättungs-Algorithmus

- Zufällige Wahl zweier Knoten im Lösungsweg
- Falls die Verbindung kollisionsfrei ist, Verbinden der beiden Knoten und Löschen der dazwischenliegenden Knoten aus dem Lösungspfad



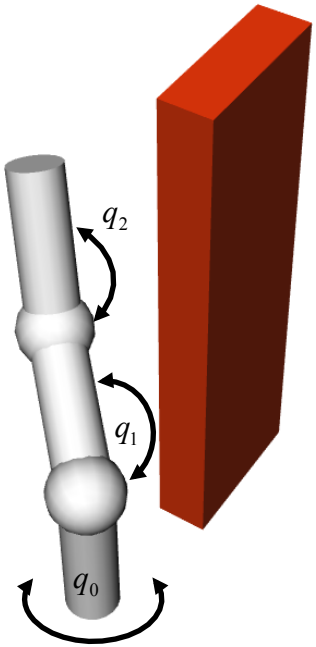




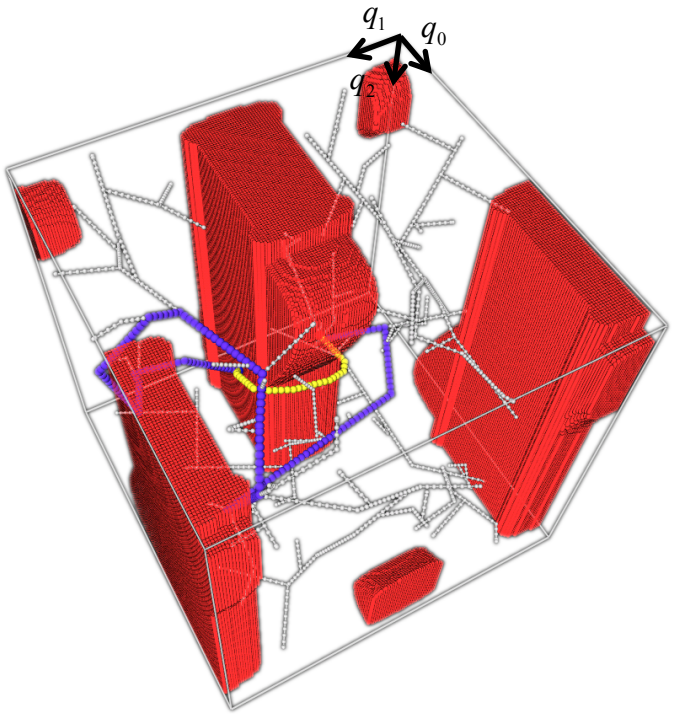
Planungsverfahren: Manipulatoren

RRT: Rapidly-exploring Random Trees (6)

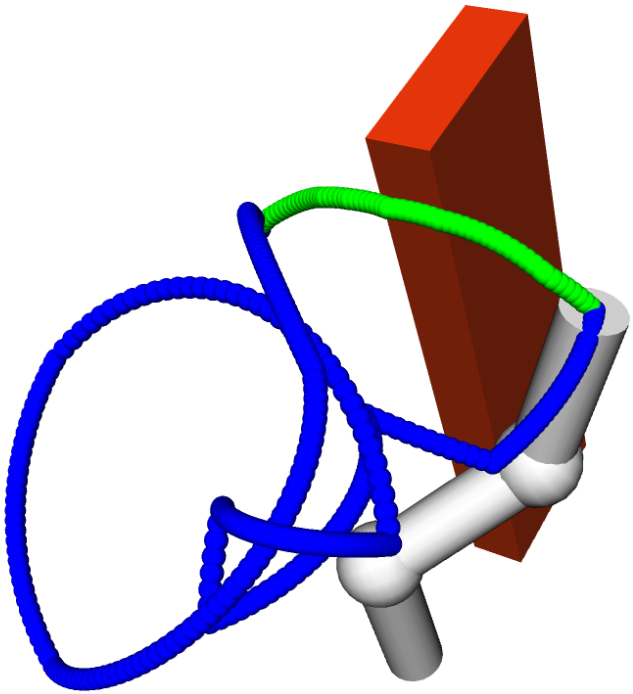
- Beispiel (dreidimensionaler Konfigurationsraum)



Start



RRT im Konfigurationsraum
C_{obst} ist rot dargestellt



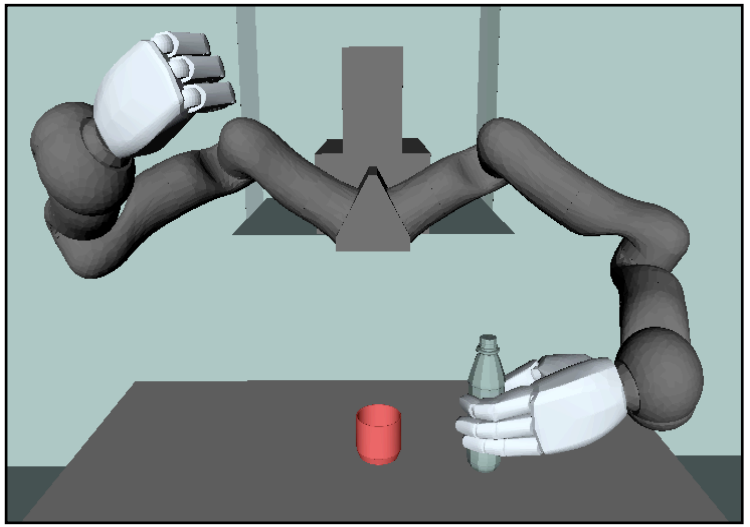
Gefunden Lösung und
geglättete Lösungstrajektorie

Planungsverfahren: Manipulatoren

RRT: Rapidly-exploring Random Trees (7)

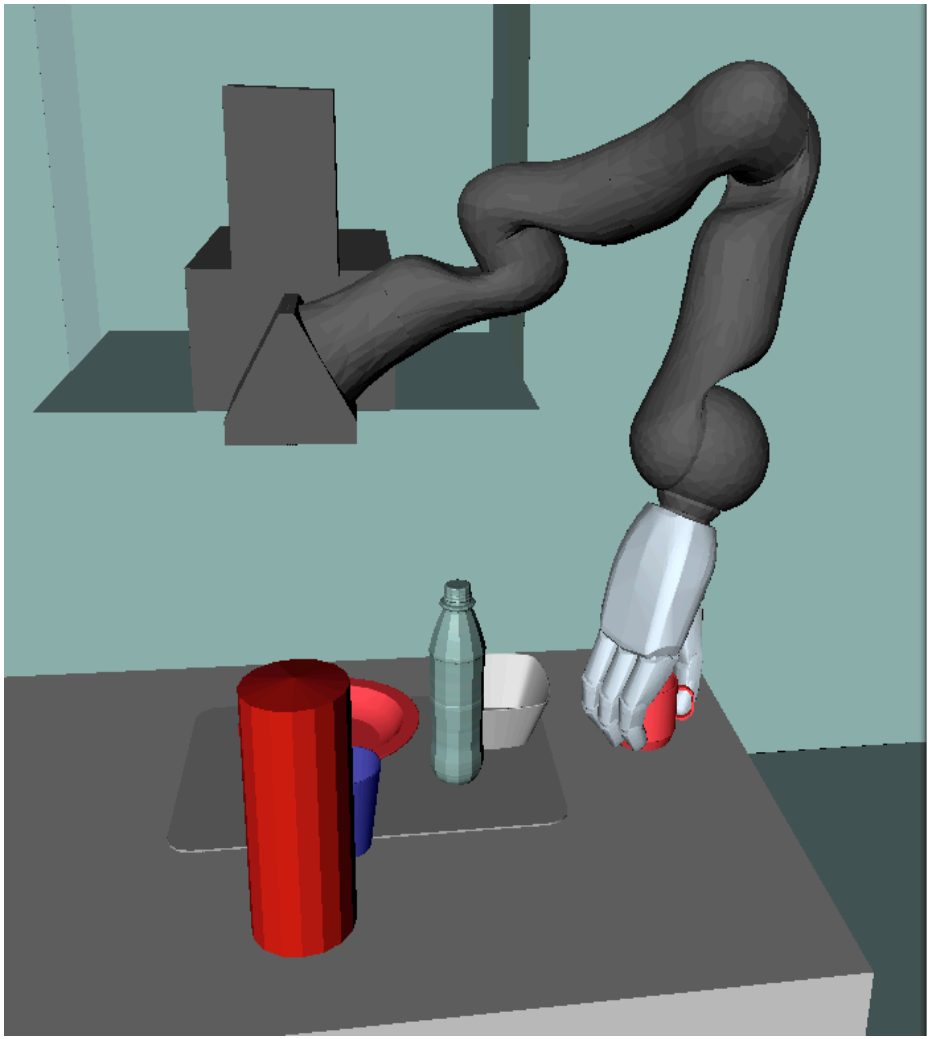
Beispiel: Holonome Planung für einen Roboterarm mit 7 Freiheitsgraden

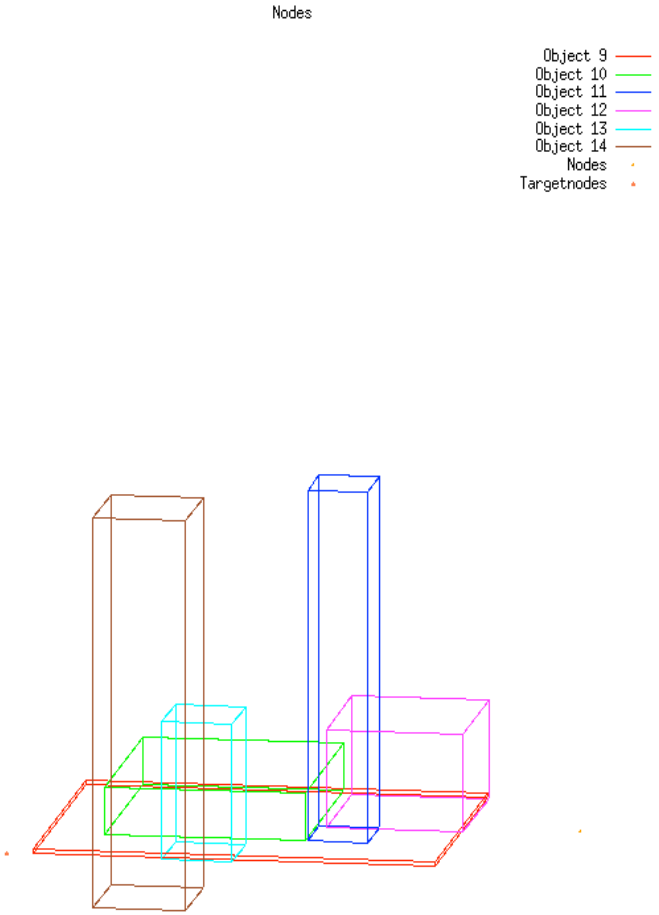
- Konfigurationsraum IK = $[0,1]^7$
- Distanzfunktion auf IK: $d(k,s) = \sum w_i (k_i - s_i)$
- Gewichtsvektor: (1.5,1.5,1.5,1.25,1,1,1)
- $\varepsilon = 0.001$, $\delta = 0.01$
- $f_{Ziel}(K) = 1 \Leftrightarrow K = K_{Ziel}$
- $f_{Weg}(K) = 1 \Leftrightarrow$ K ist kollisionsfrei



Planung für linken Arm ohne Hand

Planungsverfahren: Manipulatoren



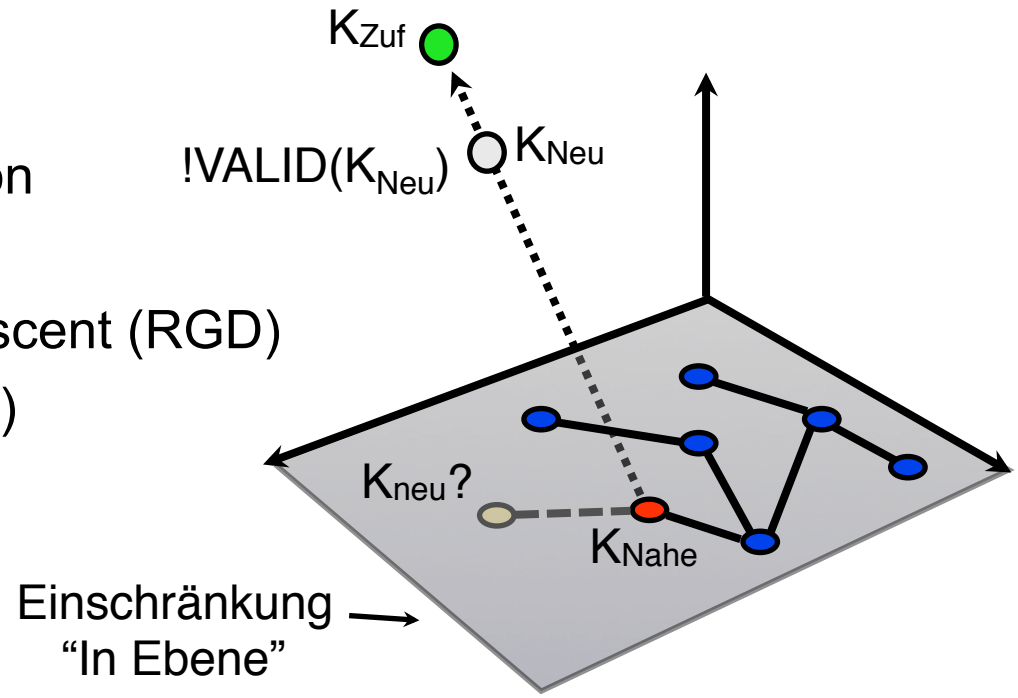


Planungsverfahren: Manipulatoren

RRT-Erweiterung

- VALID(K) unzureichend für Einschränkungen mit P(VALID(K)) = 0
- Erweiterung: Berechnung von Distanz und/oder Richtung zu Konfiguration im Freiraum

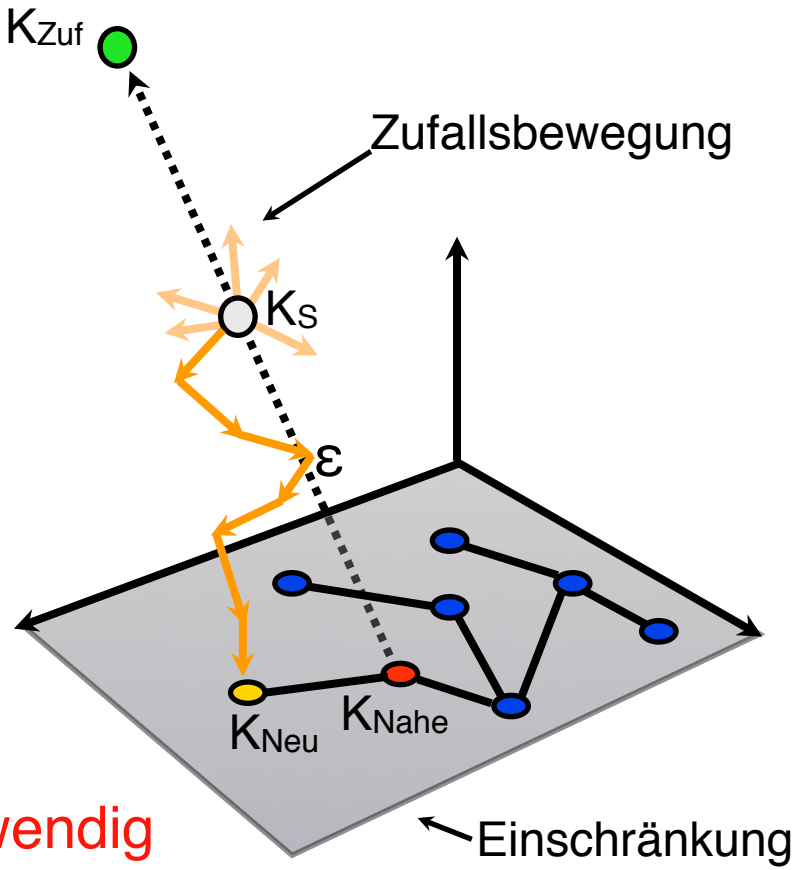
- Ersetzung der Extend-Funktion
- Beispiele:
  - Randomized Gradient Descent (RGD)
  - First Order Retraction (FR)



Planungsverfahren: Manipulatoren

Randomized Gradient Descent (RGD)

- Toleranzwert für Einschränkungen: α
- Zufällige Bestimmung von n Nachbarn von K_S (in Hyperkugel mit Radius d_{max})
- Falls minimale Distanz der Nachbarn kleiner als die Distanz von K_S, ersetze K_S mit dem Nachbarn mit kleinster Distanz
- Wiederholen bis maximale Iterationszahl erreicht oder die Distanz von K_S < α ist

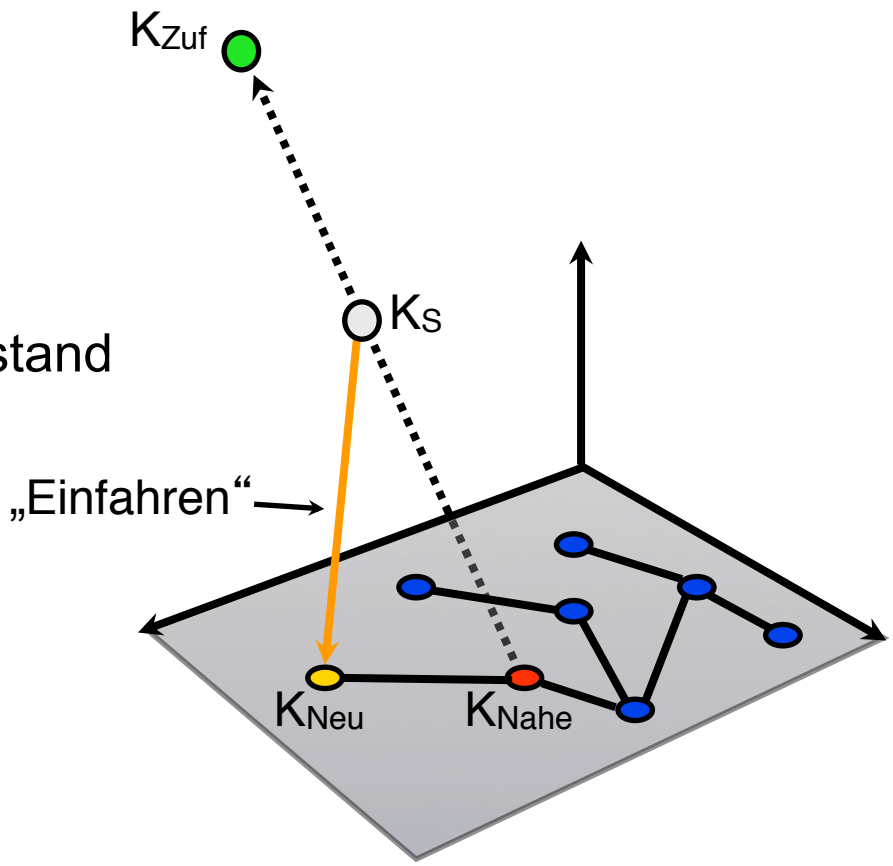


→ Keine Richtungsinformation notwendig

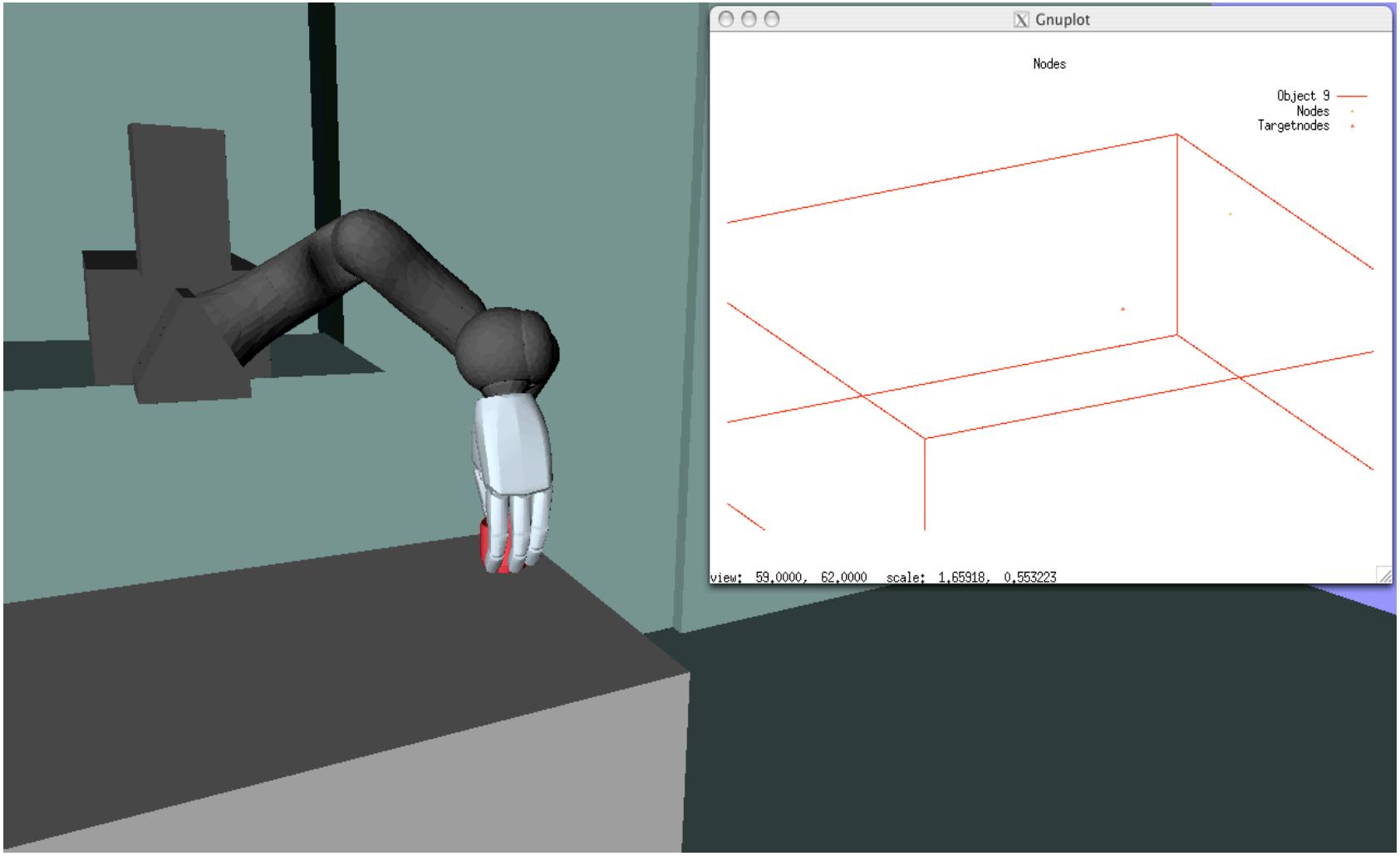
Planungsverfahren: Manipulatoren

First Order Retraction (FR)

- Toleranzwert für Einschränkungen: α
- Speichern von K_S in K_O
- Berechnen der Distanz Δx von K_S
- „Einfahren“ von K_S:
  $K_S = K_S - J(K_S)^{\dagger} \Delta x$
- Iterieren bis die Distanz < α oder Abstand
  $|K_O - K_S|$ größer als $|K_O - K_{Nahe}|$



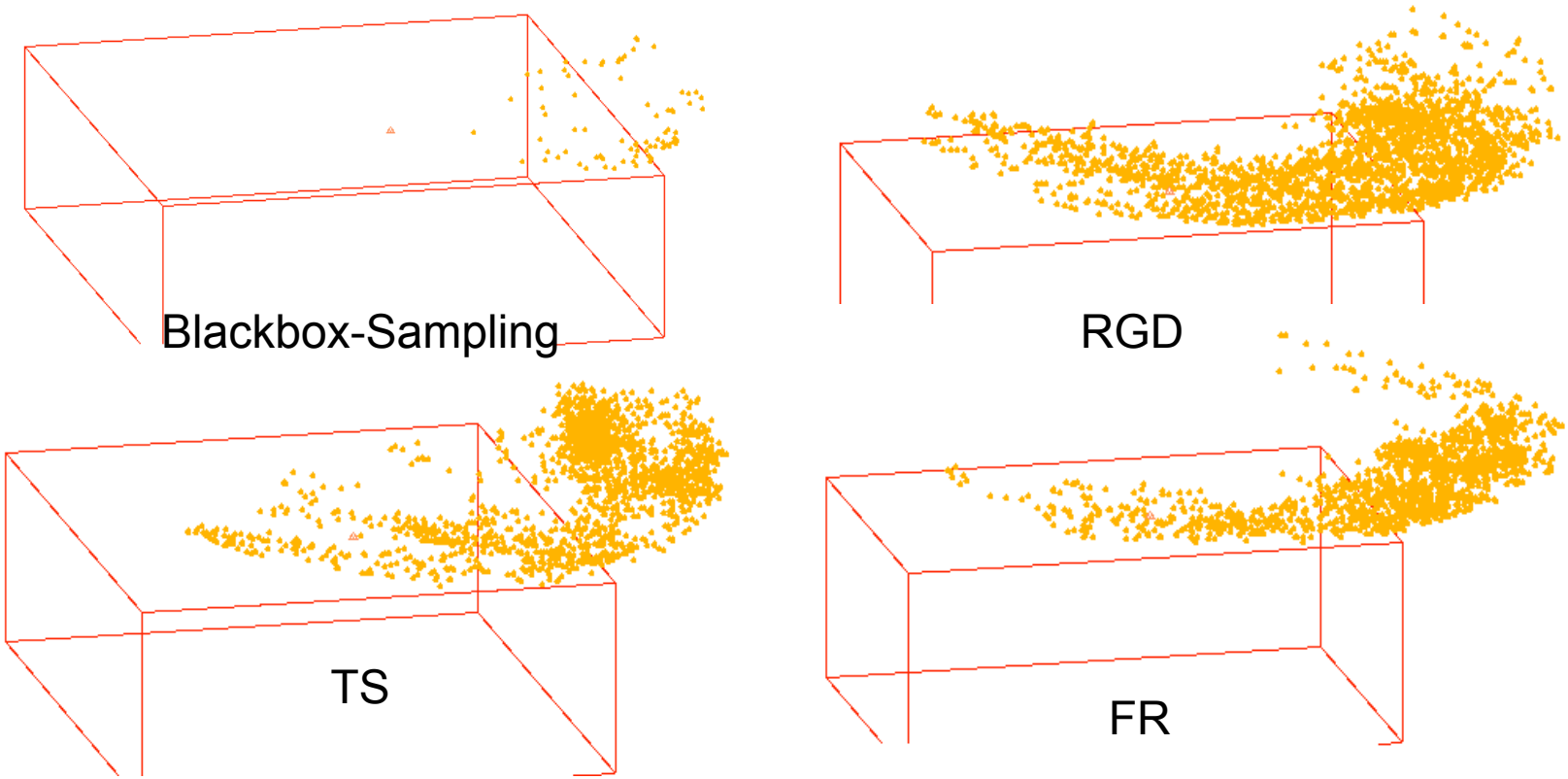
Planungsverfahren: Manipulatoren



Planungsverfahren: Manipulatoren

Beispiel: Stichprobenverteilung

- Jacobimatrix-basierte Ansätze bei komplexeren Einschränkungen im Vorteil



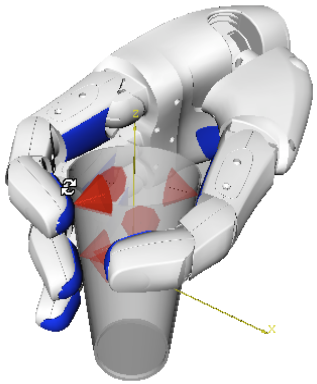
Inhalt

- Motivation
- Grundlagen der Bahnplanung
- Planungsverfahren für
  - Mobile Roboter
  - Manipulatoren
  - Robotergreifer
    - Grundlagen
    - Vorwärtsgreifplanung

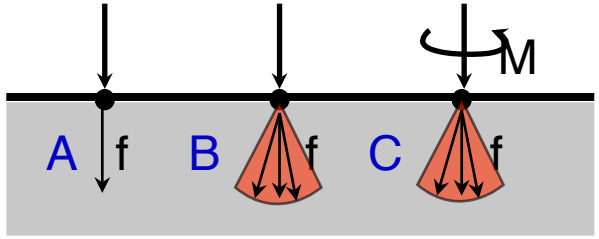
Planungsverfahren: Robotergreifer

Übersicht

- Ziel: Berechnung eines Griffs, d.h. Menge von Kontaktstellen zwischen Roboter und Objekt

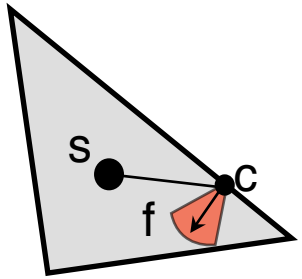


- Kontaktstellen:
  - Punktkontakt ohne Reibung (A)
  - Starrer Punktkontakt mit Reibung (B)
  - Nicht-starrer Punktkontakt mit Reibung („soft finger contact“) (C)
  - Flächenkontakte



- Wirkung auf Objekt
  - Wrenchvektor: Kraft + Moment auf Objekt:

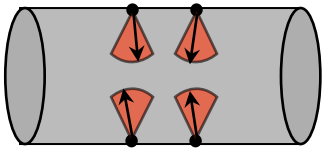
$$\begin{pmatrix} f \\ (c\text{-}s) \times f \end{pmatrix}$$



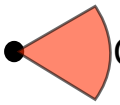
Planungsverfahren: Robotergreifer

Definitionen

- Beschreibung der Griffqualität:
  - Cone Wrench Space
  - Grasp Wrench Space
  - Task Wrench Space



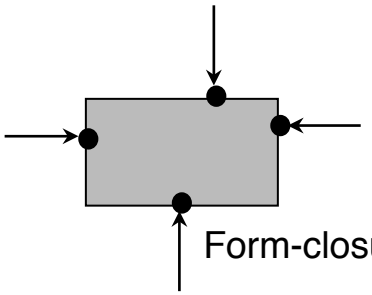
GWS: alle möglichen Summen aus jeweils einem Wrenchvektor jedes einzelnen Kegels



CWS: Kegel

TWS: Knopf drücken

- Eigenschaften eines Griffs:
  - Widerstand gegen Stöße (beliebiger externer Wrench w):
    - Force-closure: -w liegt im GWS
    - Form-closure: geometrische Einschränkung



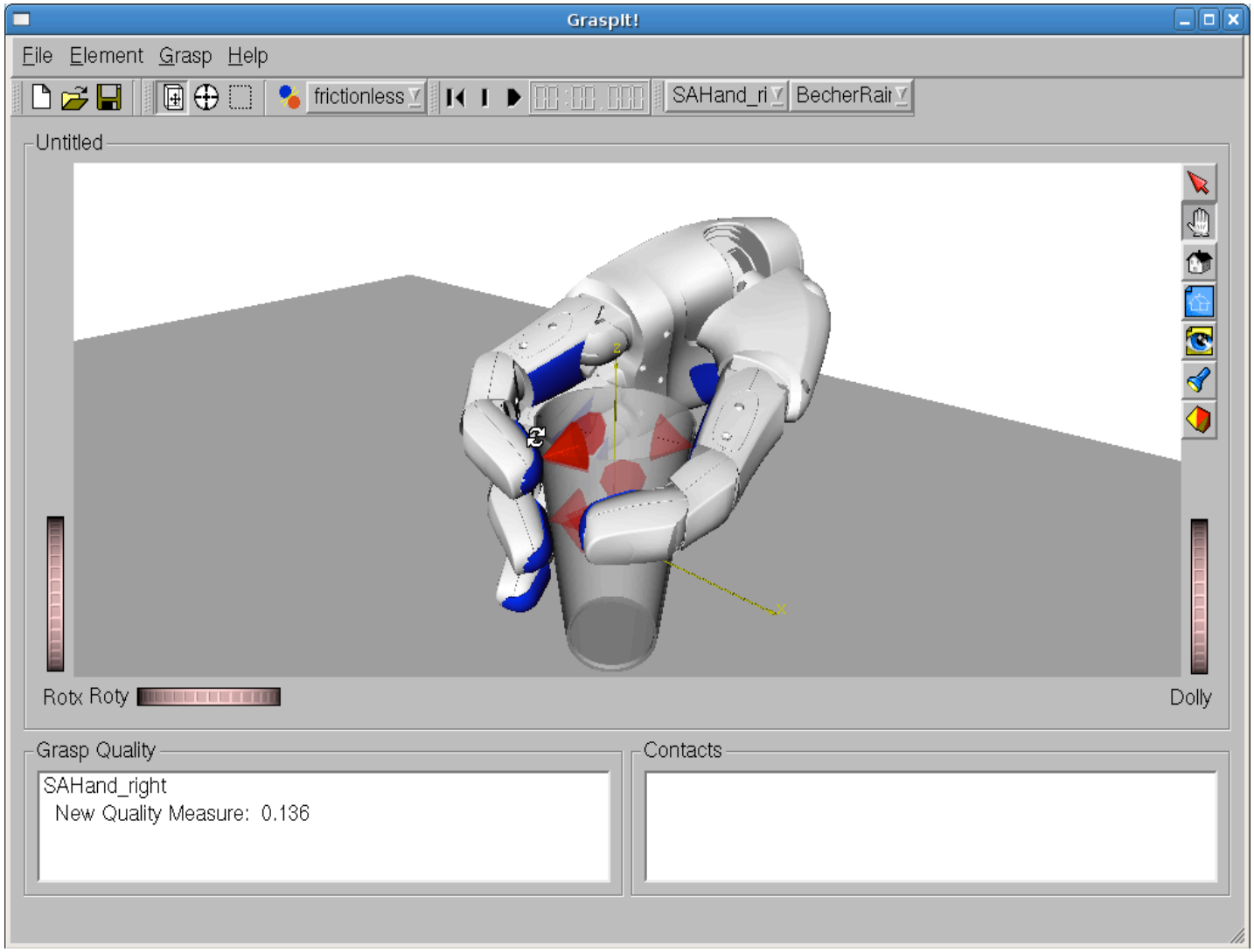
- Qualitätsmaße für Griffe (Grasp quality measure)
  - Beispiel: größte Hyperkugel um 0, die im GWS liegt

Planungsverfahren: Robotergreifer

Vorwärtsgreifplanung

1. Setze Gelenkwinkel des Handmodells vor dem Zugreifen, z.B. prismatischer Kraftgriff
2. Setze 3d-Handmodell relativ zu Objekt vor dem Anrücken
3. Bewege Hand auf das Objekt zu
4. Schließe jeden einzelnen Finger bis auf Kontakt
5. Bestimme Kraftkegel in allen Kontaktpunkten
6. Berechne Griffqualität
7. * Iteriere bis Griff mit hoher Griffqualität gefunden

GraspIt! Simulator

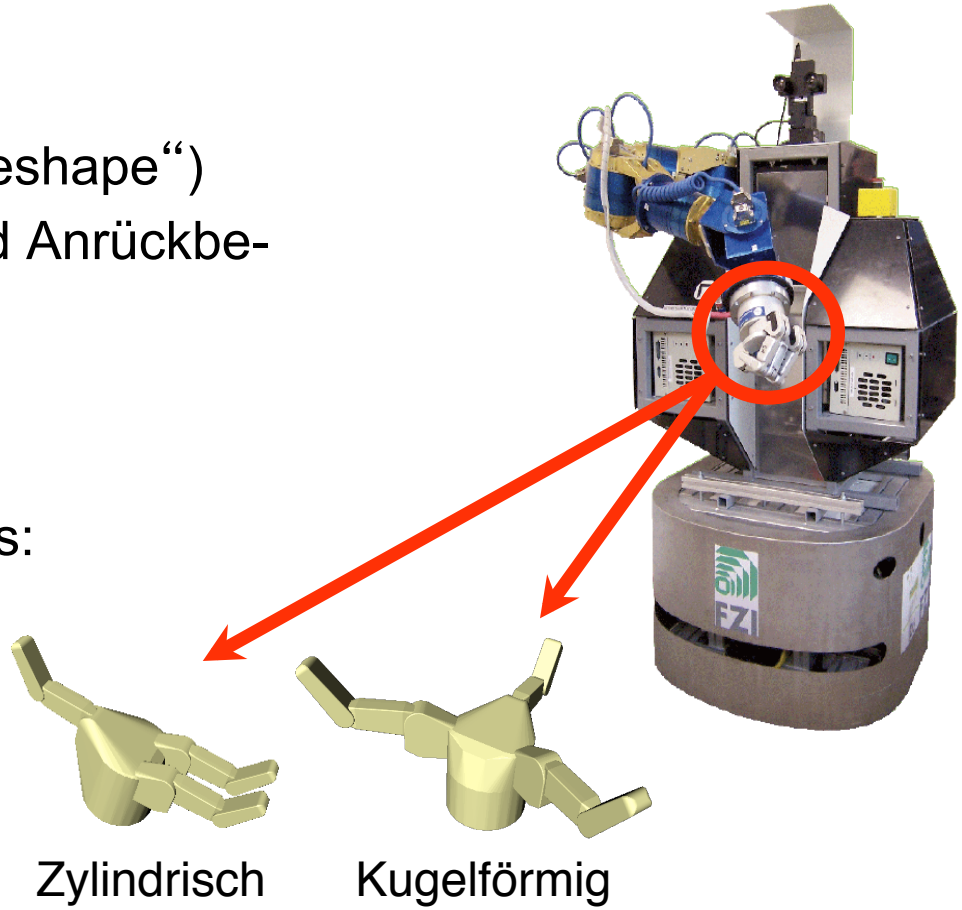


Planungsverfahren: Robotergreifer

GraspIt!: Vorwärtsgreifplanung

Miller03: Automatic Grasp Planning Using Shape Primitives

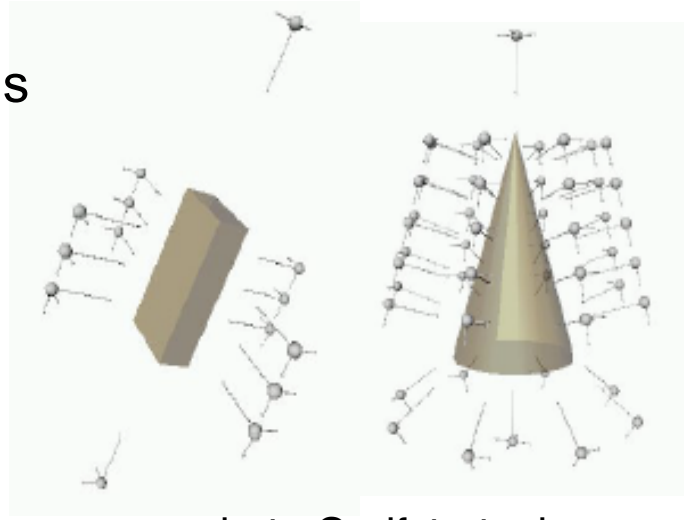
- Übersicht:
  - Vorgabe einer Griffform („preshape“)
  - Vorgabe einer Startpose und Anrückbe-wegung
  - Durchführung des Griffs
  - Evaluation mit Qualitätsmaß
  - Zwei vordefinierte Preshapes:



Planungsverfahren: Robotergreifer

GraspIt!: Vorwärtsgreifplanung

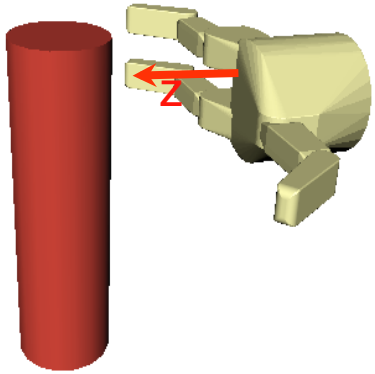
- Greifstrategie (Preshape, Startlage) auf Basis einer Zerlegung des Objekts in Primitive:
  - Kugeln, Zylinder, Kegel, Quader



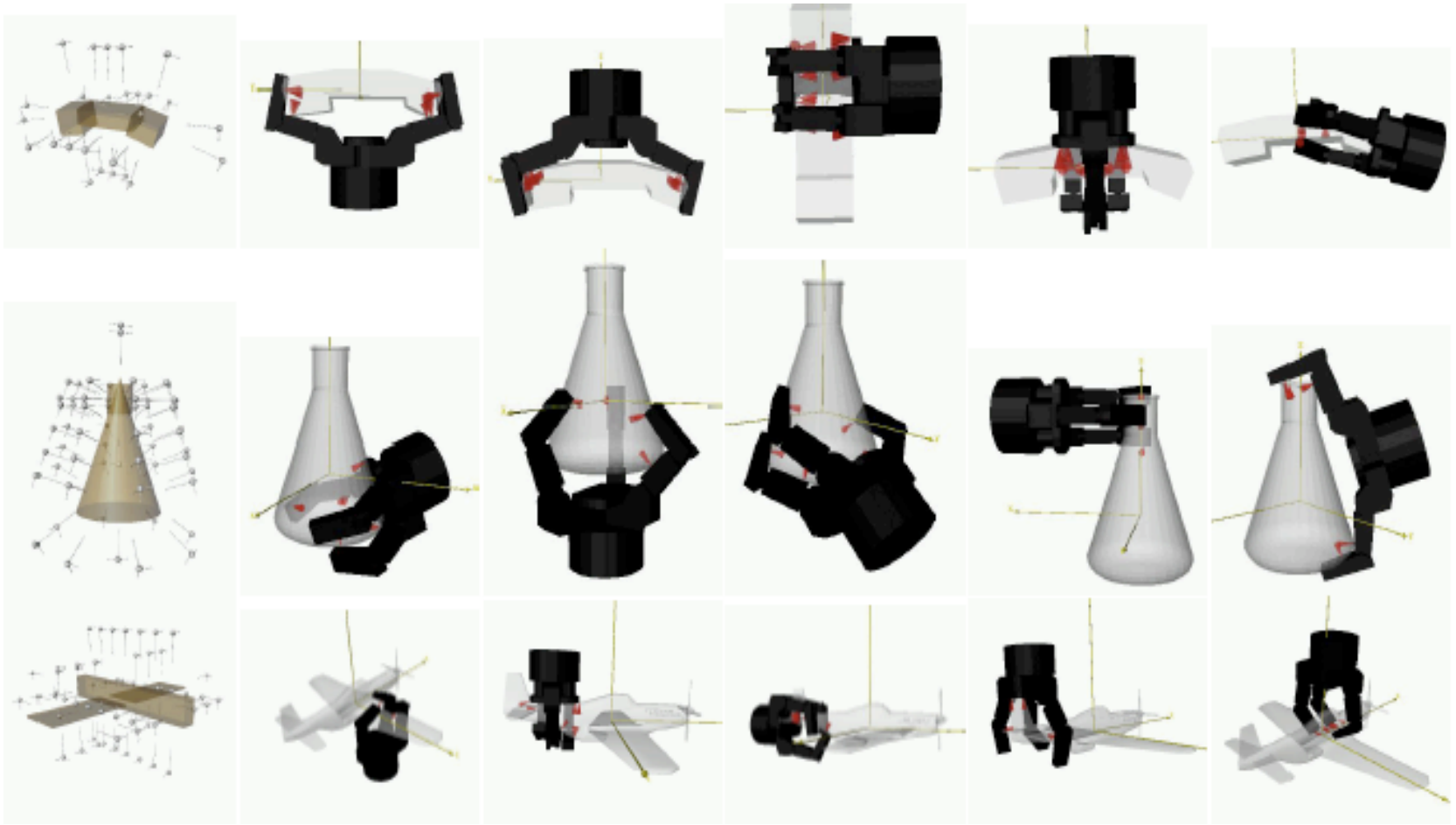
generierte Greifstrategien

- Vorgabe einer Menge von Greifstrategien für jedes Objektprimitive

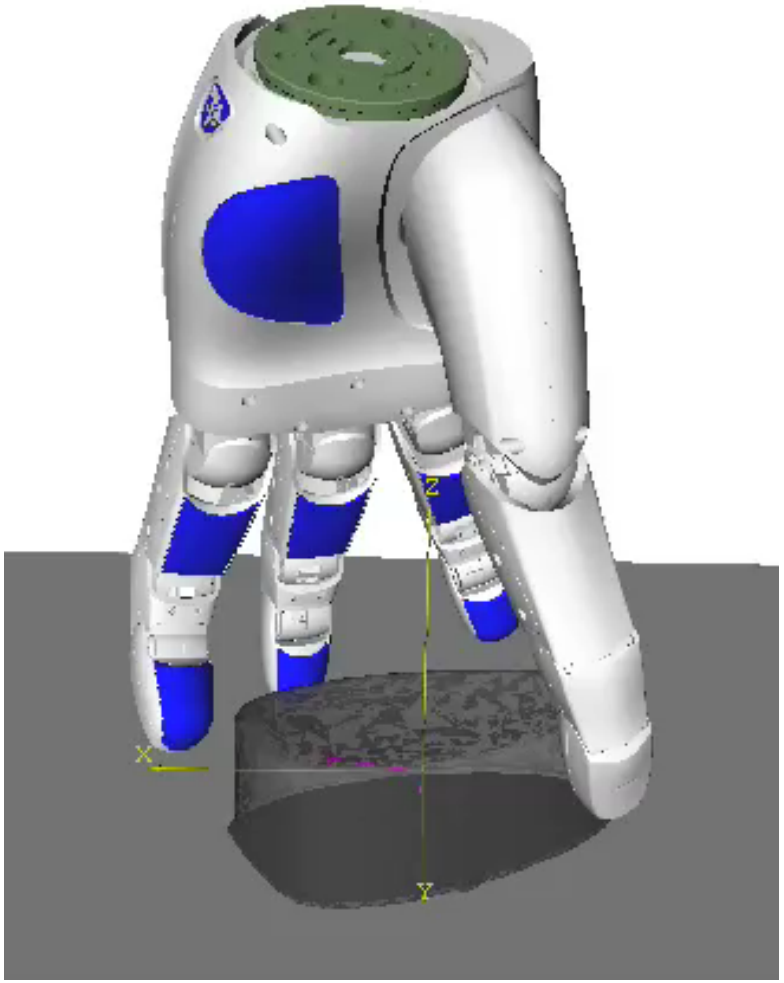
- Anrückbewegung: linear in z-Richtung bis zu einem Kontakt und danach Backtracking

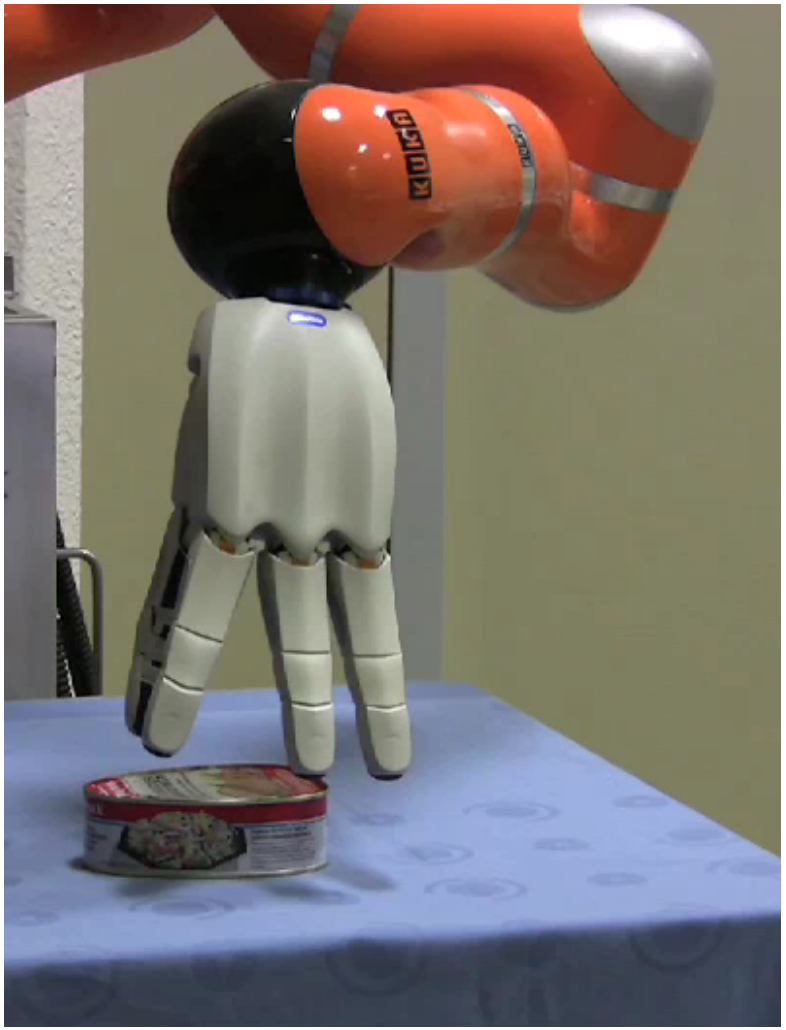


Planungsverfahren: Robotergreifer



Planungsverfahren: Robotergreifer





Nächste Woche

Programmieren durch Vormachen