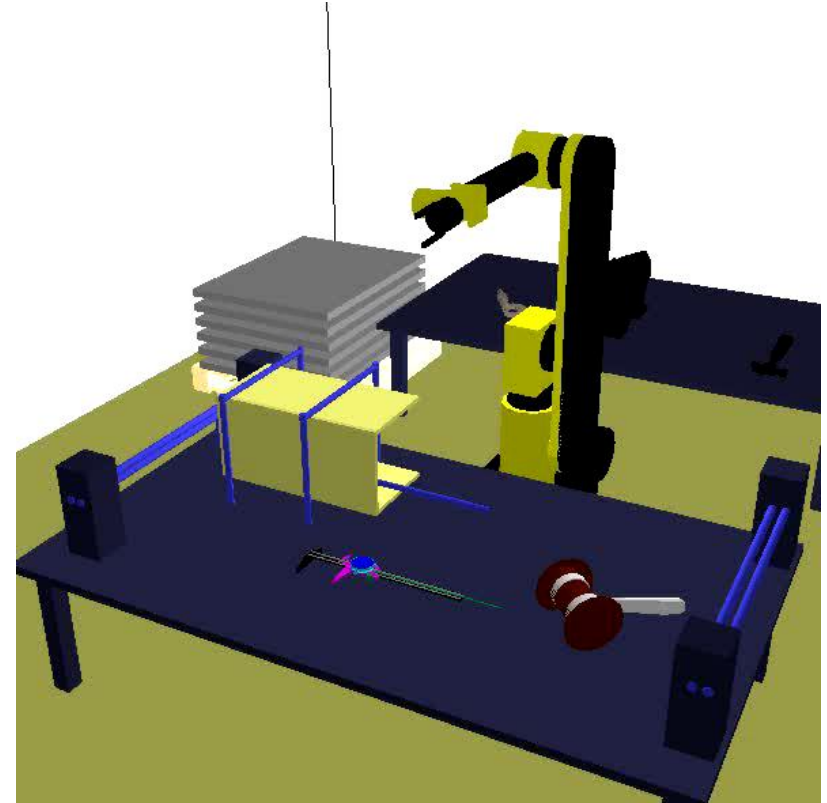
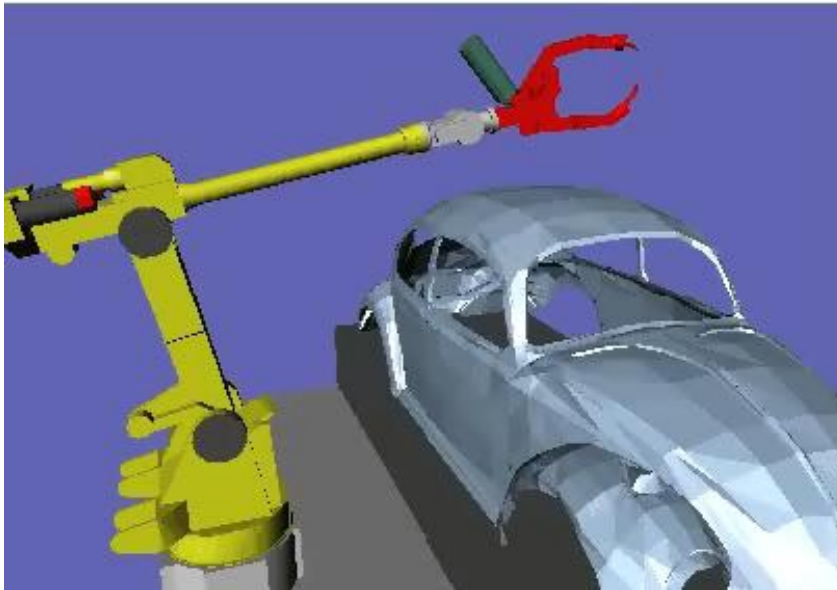


XIII. Bahnplanung

Prof. Dr.-Ing. Rüdiger Dillmann

- Motivation
- Grundlagen der Bahnplanung
- Planungsverfahren

Bahnplanung zur Erzeugung von kollisionsfreien Bewegungen



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

- **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

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 Manipulatoren
 - Bahnplanung für mobile Roboter
 - Bahnplanung für Laufmaschinen und anthropomorphe Systeme
 - Greif- und Montageplanung

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

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

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

$$\begin{aligned}\tau: [0,1] &\rightarrow \mathcal{C} \\ \text{mit } \tau(0) &= q_{\text{Start}} \\ \tau(1) &= q_{\text{Ziel}}\end{aligned}$$

Definitionen (2)

- Arbeitsraumhindernis**

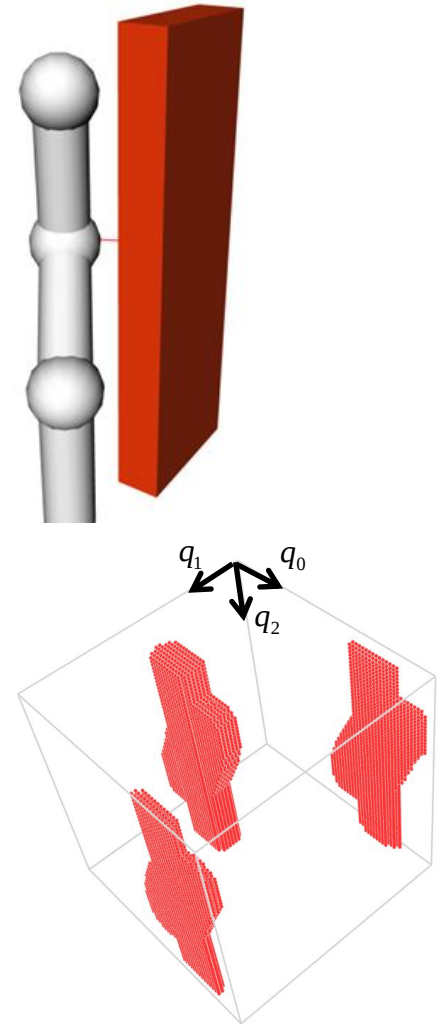
Ein **Arbeitsraumhindernis** H ist der Raum, welcher von einem Objekt im Arbeitsraum eingenommen wird.

- Konfigurationsraumhindernis**

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

- Hindernisraum**

Der **Hindernisraum** ist die Menge aller Konfigurationsraumhindernisse: $\mathcal{C}_{\text{obst}} = \bigcup \mathcal{C}_{H_i}$



Definitionen (3)

- **Freiraum**

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

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

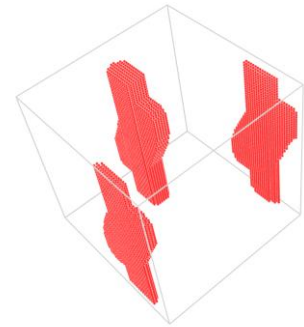
Aufwand für die Berechnung des Freiraums:

$$O(m^n)$$

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

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

-> Einsatz von approximativen Verfahren zur vereinfachten Repräsentation des Freiraumes.



- Motivation
- Grundlagen der Bahnplanung
- Planungsverfahren
 - Voronoi-Diagramme
 - Sichtgraphen
 - Zellzerlegung
 - Potentialfelder
 - Probabilistic Roadmaps (PRM)
 - Rapidly-exploring Random Trees (RRT)

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

Beispiel: Straßenkarte

Gegeben:

- 2-dim. Weltmodell
- Start und Ziel

Aufgabe:

Suche günstigste Verbindung von Start zu Ziel

Lösung:

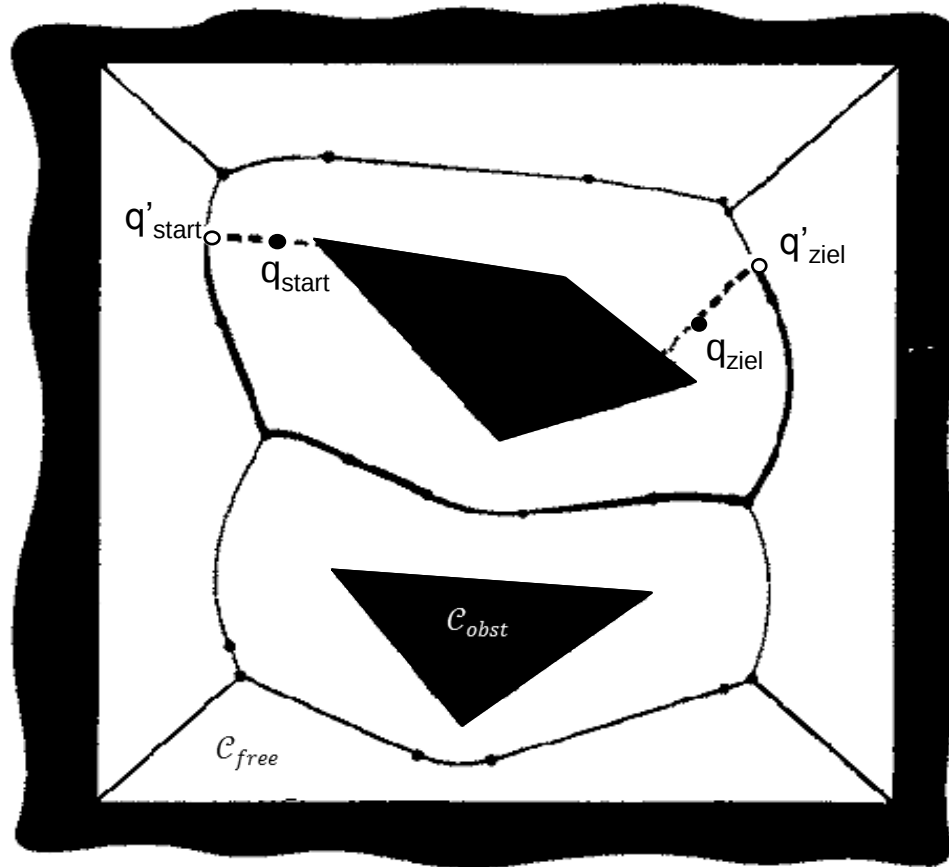
- 1) Konstruiere ein Netz von Wegen $\mathbb{W}$ in $\mathcal{C}_{\text{free}}$
- 2) Bilde q_{Start} und q_{Ziel} auf $\mathbb{W}$ ab
 $\rightarrow q'_{\text{Start}} = \mathbb{W}(q_{\text{Start}}), q'_{\text{Ziel}} = \mathbb{W}(q_{\text{Ziel}})$
- 3) Suche einen Weg, der q'_{Start} mit q'_{Ziel} verbindet

Teilaufgaben

1. Wegkonstruktion mit
 - Retraktionsverfahren, z.B. Voronoi-Diagramm
 - Sichtgraphen
2. Suche im Netz, z.B. mit
 - A*-Algorithmus (Baumsuche)
 - euklidischer Abstand
 - Potentialfeld

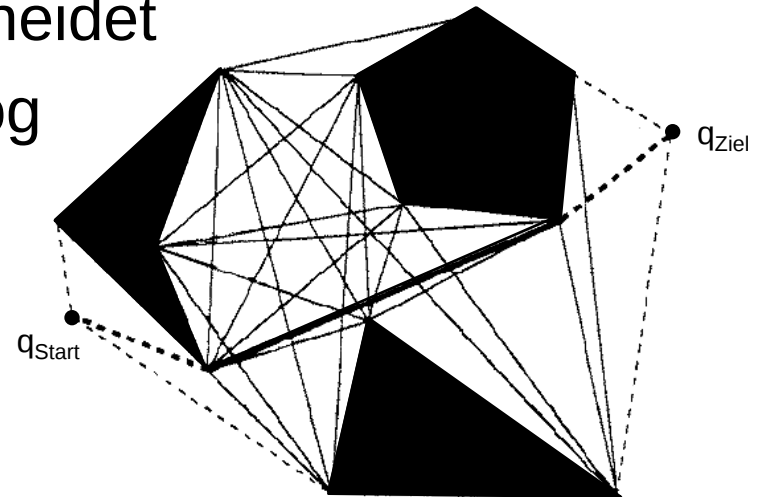
Beispiel: Voronoi-Diagramm

p: mittlerer Abstand

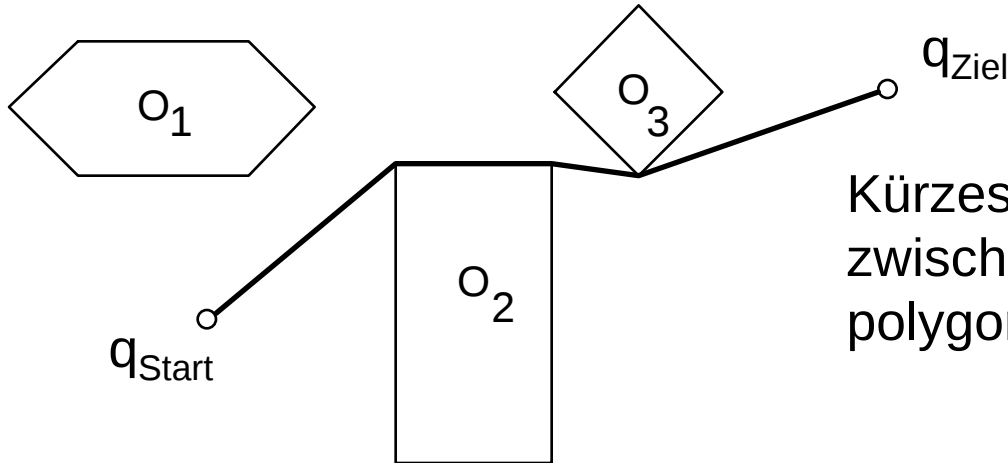


Sichtgraphen

- 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

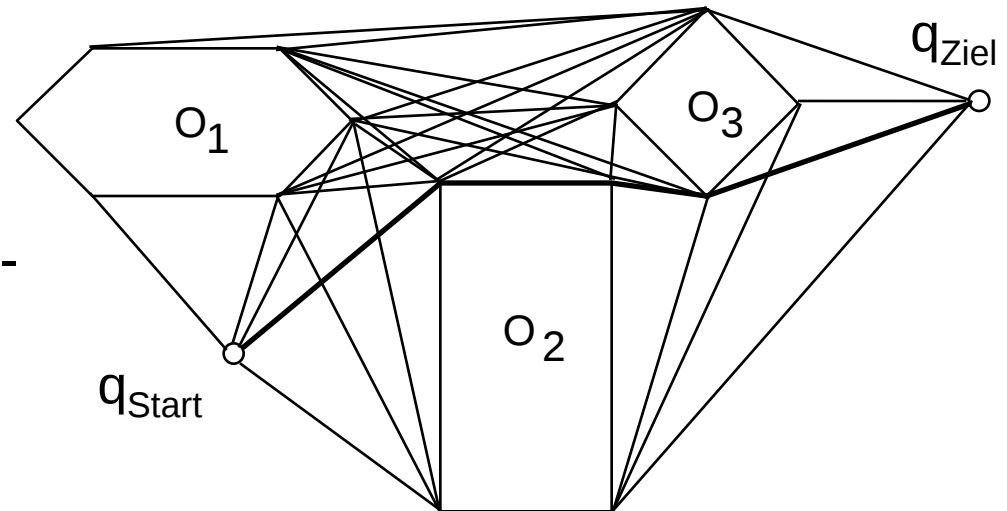


Beispiele für Sichtgraphen



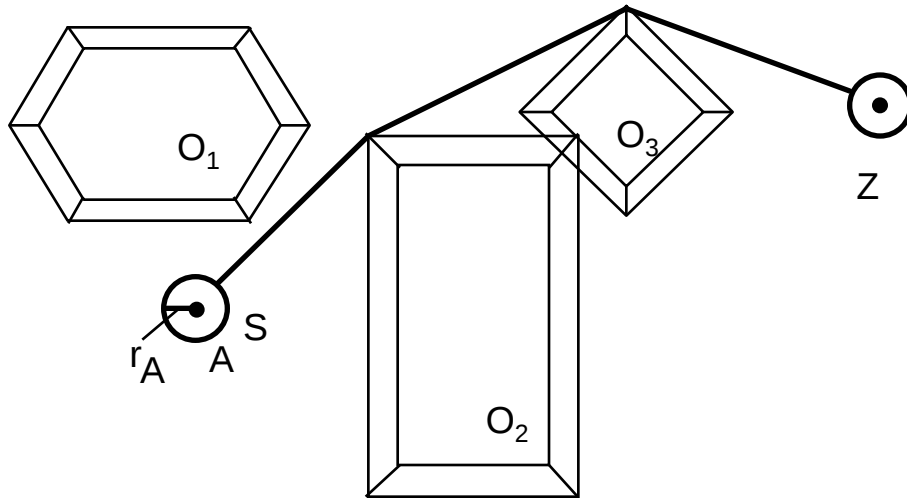
Kürzester kollisionsfreier Weg
zwischen konvexen
polygonalen Hindernissen

Methode des Sichtgraph-
Algorithmus



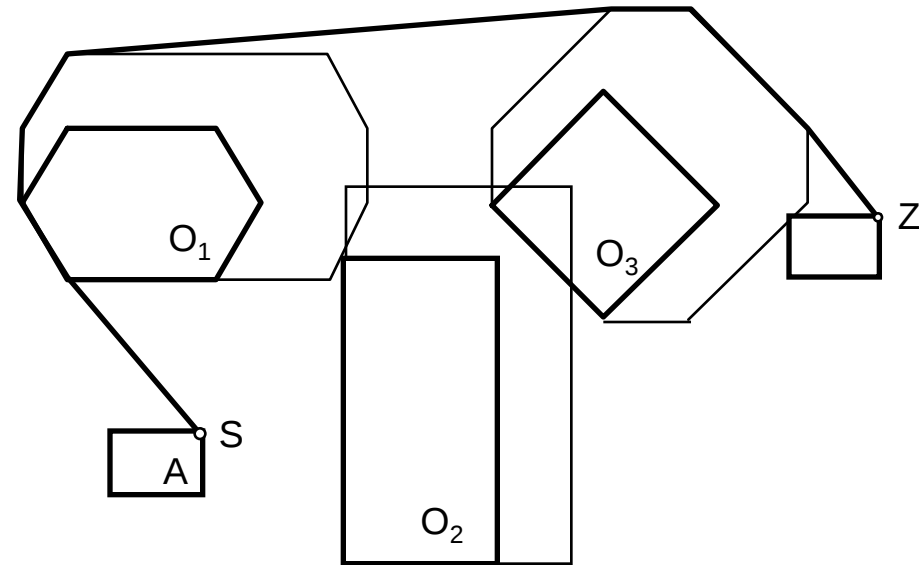
- Wege sind nur “halbfrei” (nicht kollisionsfrei), da Hinderniskanten auch Wegsegmente sein können. Abhilfe durch Erweiterung der Hindernisse.
- Wenn ein Weg gefunden ist, ist es auch der kürzeste Weg
- Methode ist exakt, wenn Roboter nur 2 translatorische Freiheitsgrade hat und sowohl Roboter als auch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

Erweiterung der Hindernisse



Vergrößern von Hindernissen
durch einen Kreis

Vergrößern von Hindernissen
durch ein Rechteck



Zellenzerlegung

Vorgehen:

1. Zerlege C_{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:

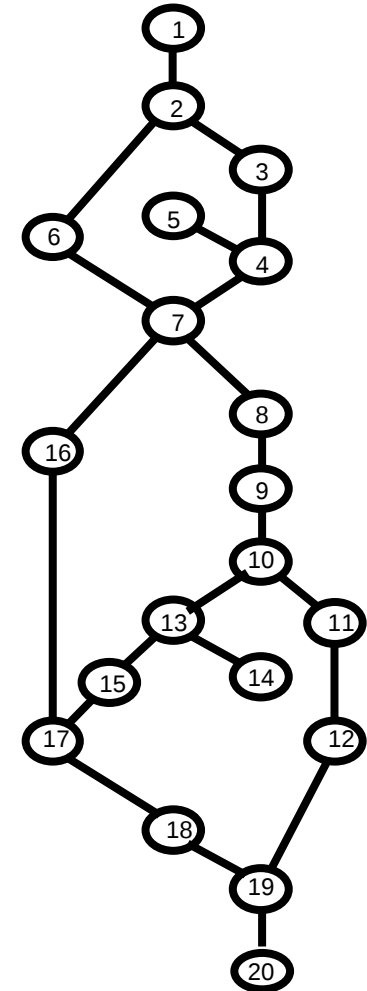
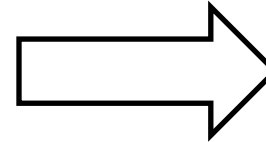
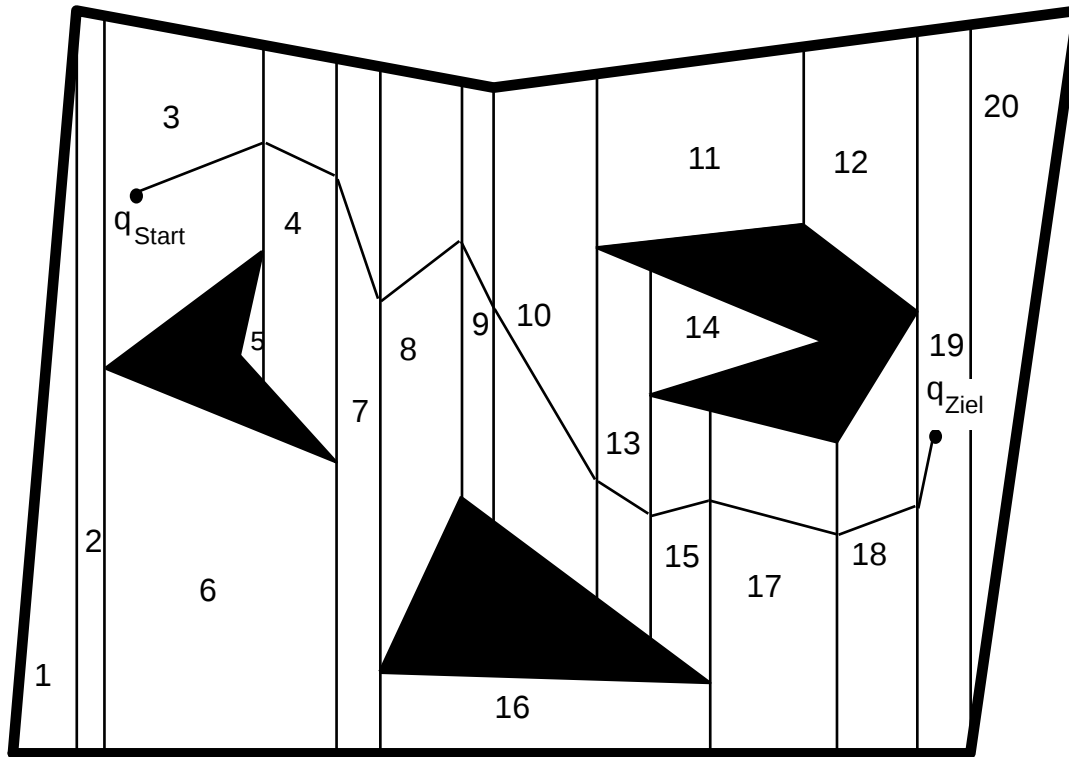
- die Zellen sich nicht überlappen

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

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

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

Beispiel



Exakte Zellenzerlegung mit Line-Sweep

Verbindungsgraph

Approximative Zellenzerlegung

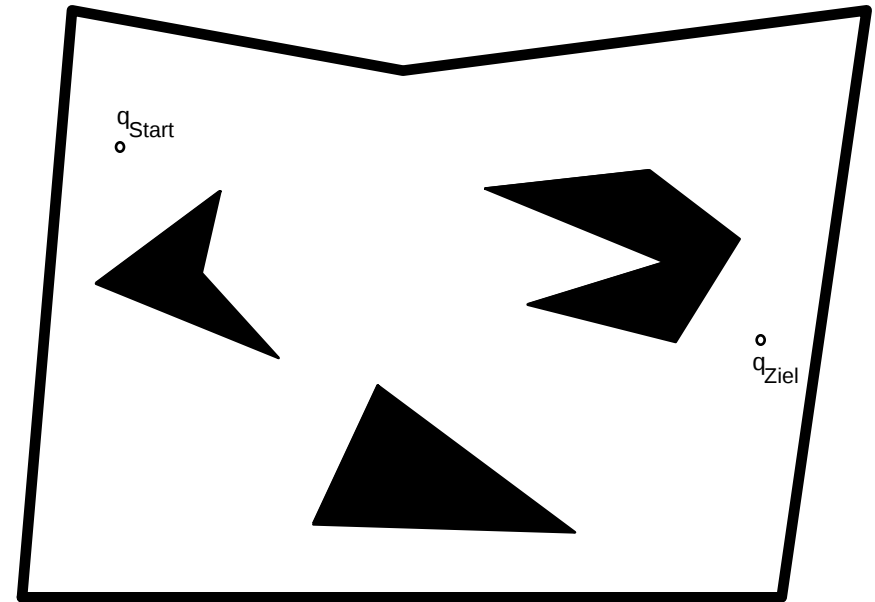
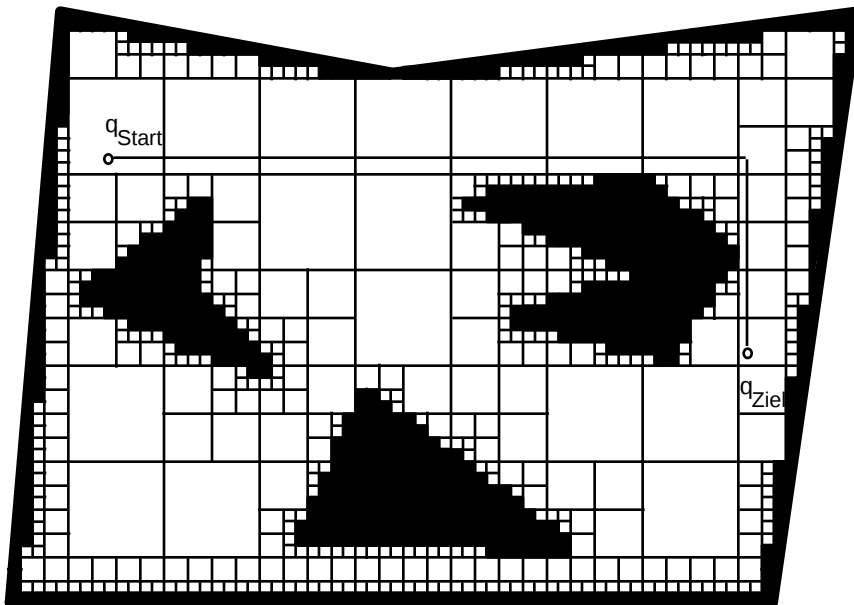
Vorgehen:

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

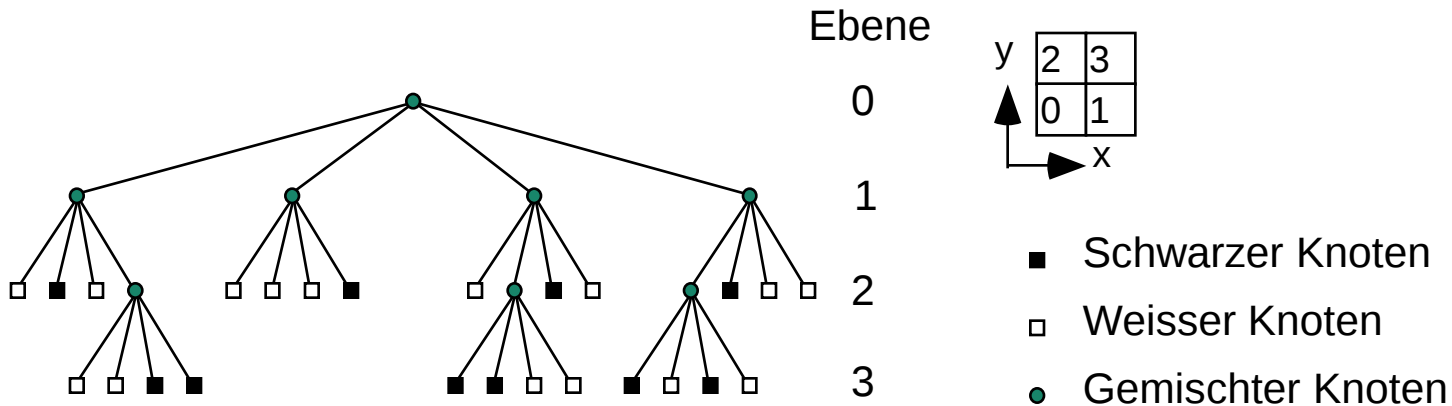
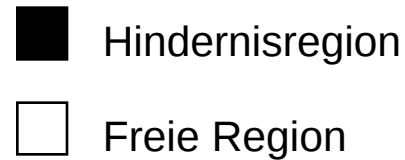
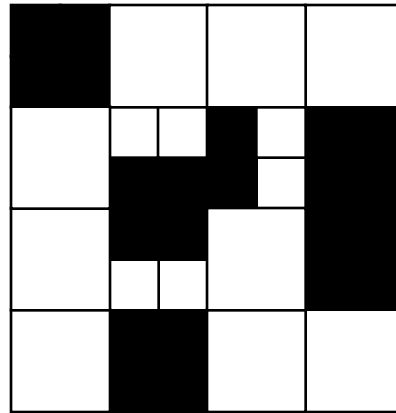
Vorteil: einfache Zerlegung → einfachere Wegsuche

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

Approximative Zellenzerlegung

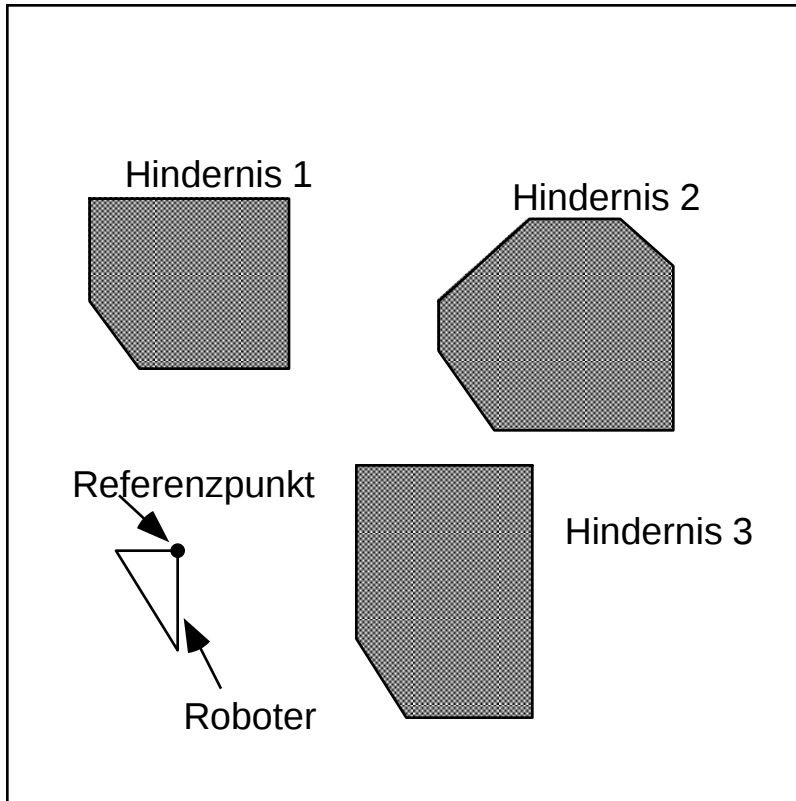


Beispiel: Quadtree mit Baumsuche (1)



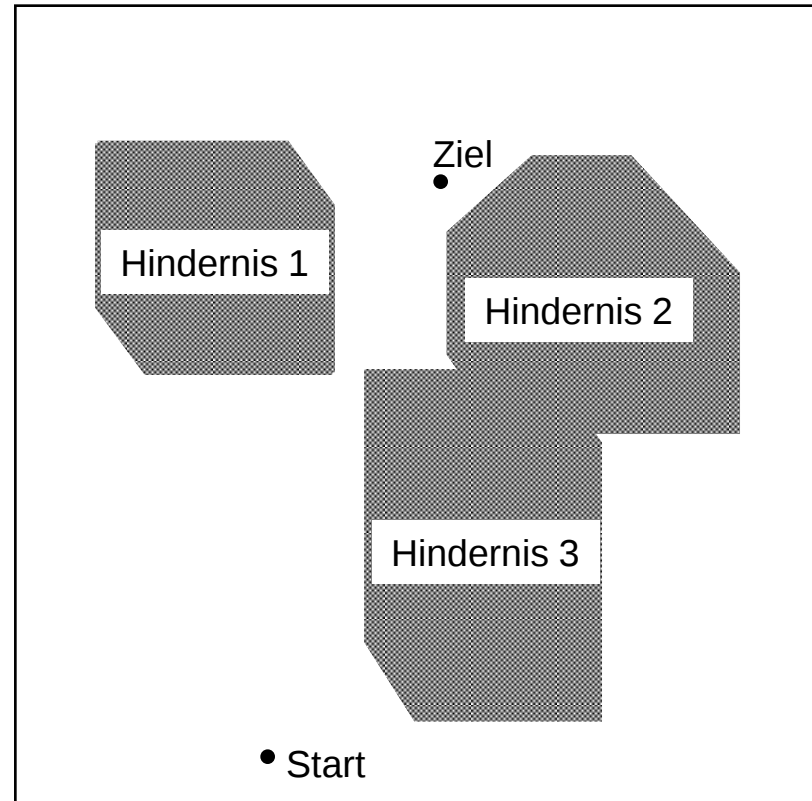
Beispiel: Quadtree mit Baumsuche (2)

Arbeitsraum



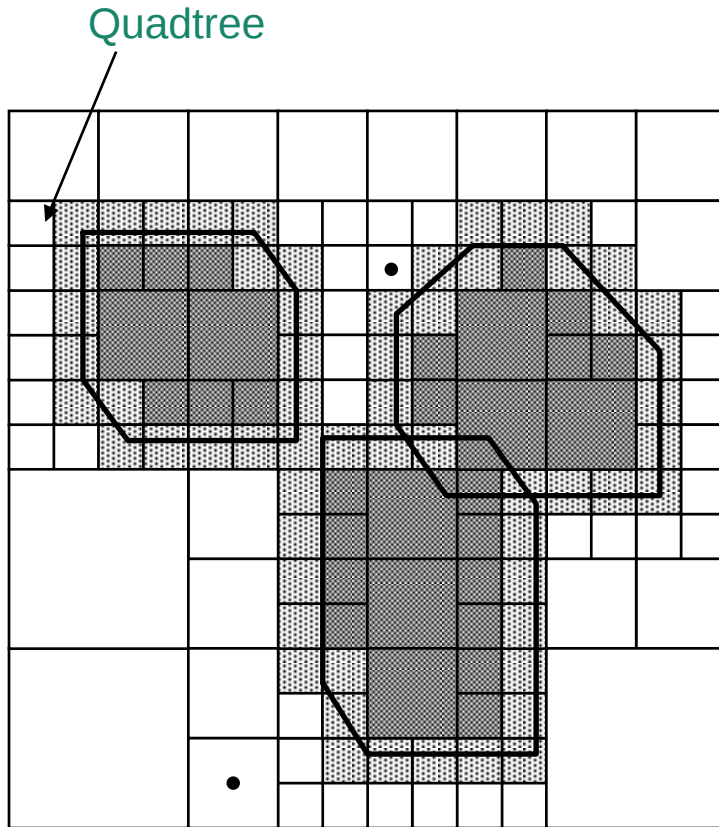
Arbeitsraum eines Roboters mit Hindernissen

Konfigurationsraum

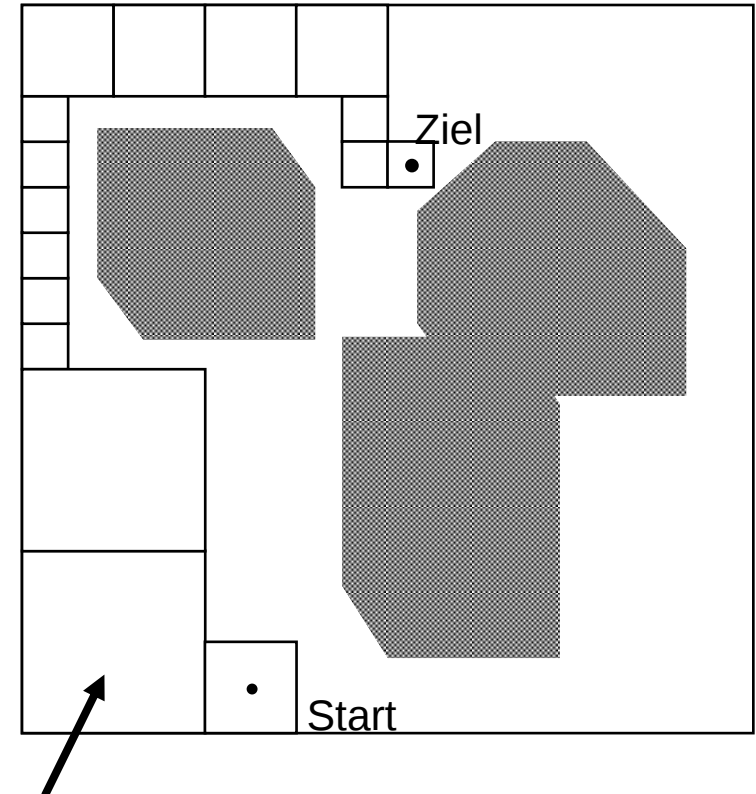


Konfigurationsraum für das vorliegende Robotersystem

Beispiel: Quadtree mit Baumsuche (3)



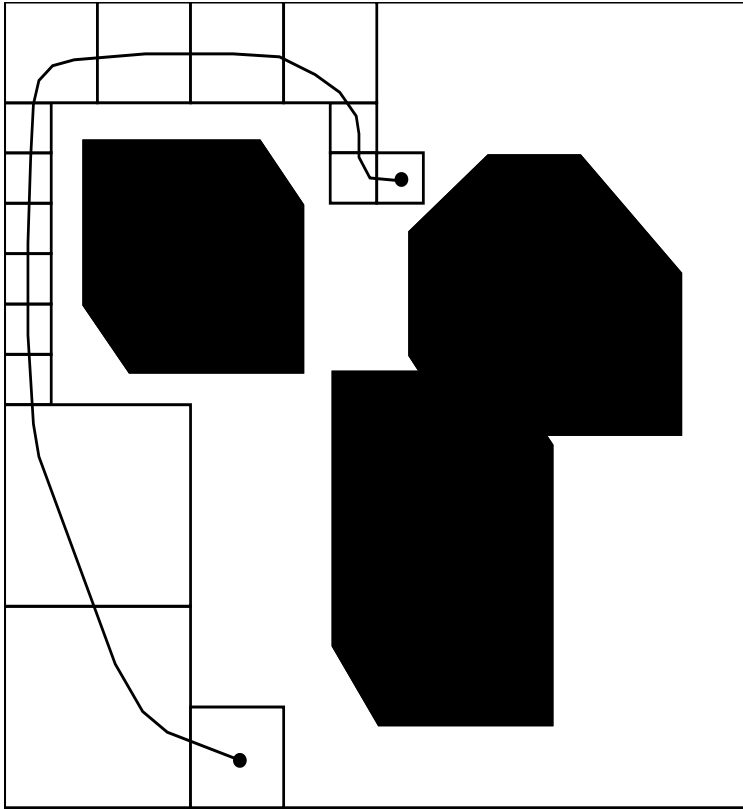
Zerlegung des Konfigurationsraums
in Kachelzonen



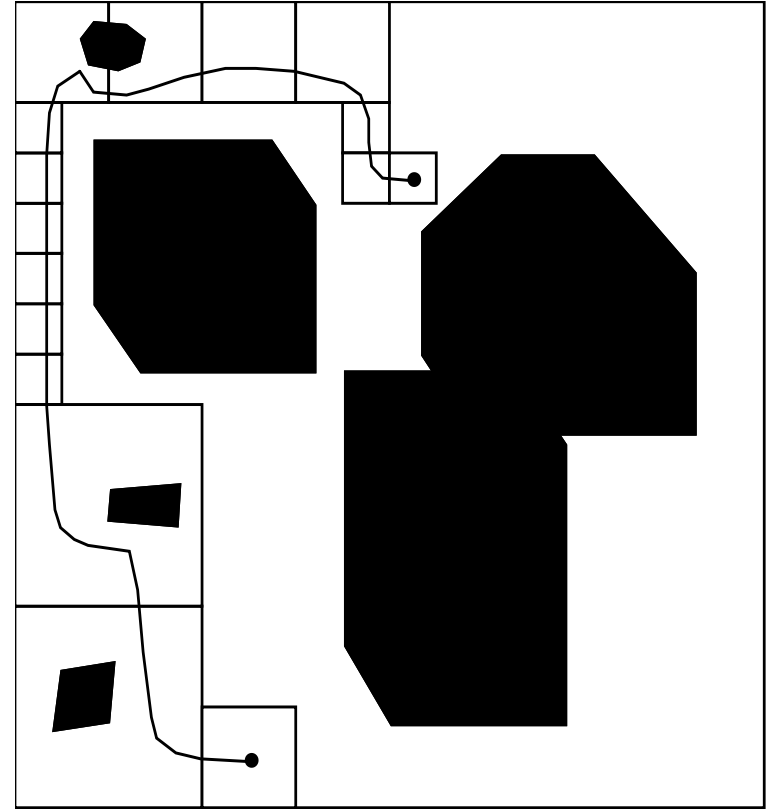
Freie Kachel

Folge von freien Kacheln vom
Start- zum Zielpunkt

Beispiel: Quadtree mit Baumsuche (4)



Hindernisfreie Verfahrroute



Ausweichmanöver um lokale Hindernisse

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

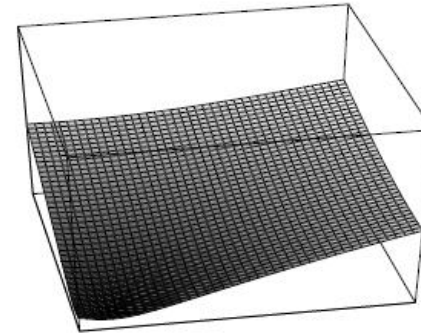
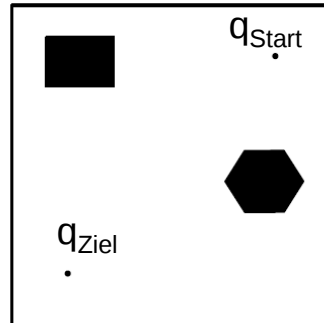
$$\vec{F}(q) = -\vec{\nabla}U(q)$$

Potentialfelder (2)

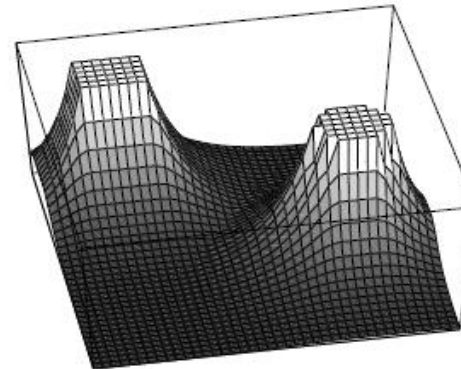
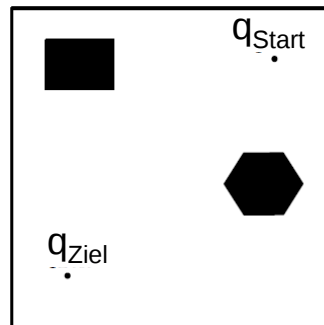
- Abstoßungspotenzial
 - Hindernisse erzeugen ein abstoßendes Potential
 - In großem Abstand zu Hindernissen soll der Roboter nicht beeinflusst werden
- Anziehungspotential
 - Es soll nur ein Minimum in q_{Ziel} geben

Beispiel: Potentialfelder (1)

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



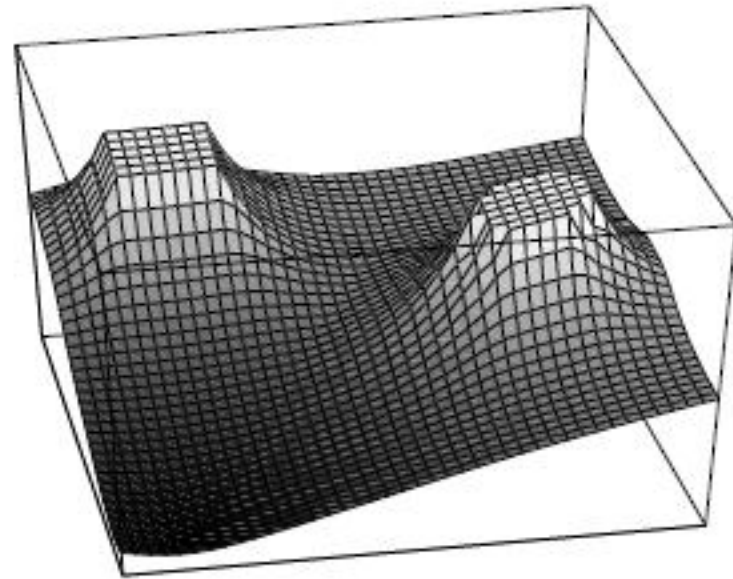
- Die Hindernisräume $\mathcal{C}_{H_i}$ haben ein Abstoßungspotential U_{ab}



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:

$$\vec{F}(q) = \vec{F}_{An}(q) + \vec{F}_{Ab}(q)$$



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:

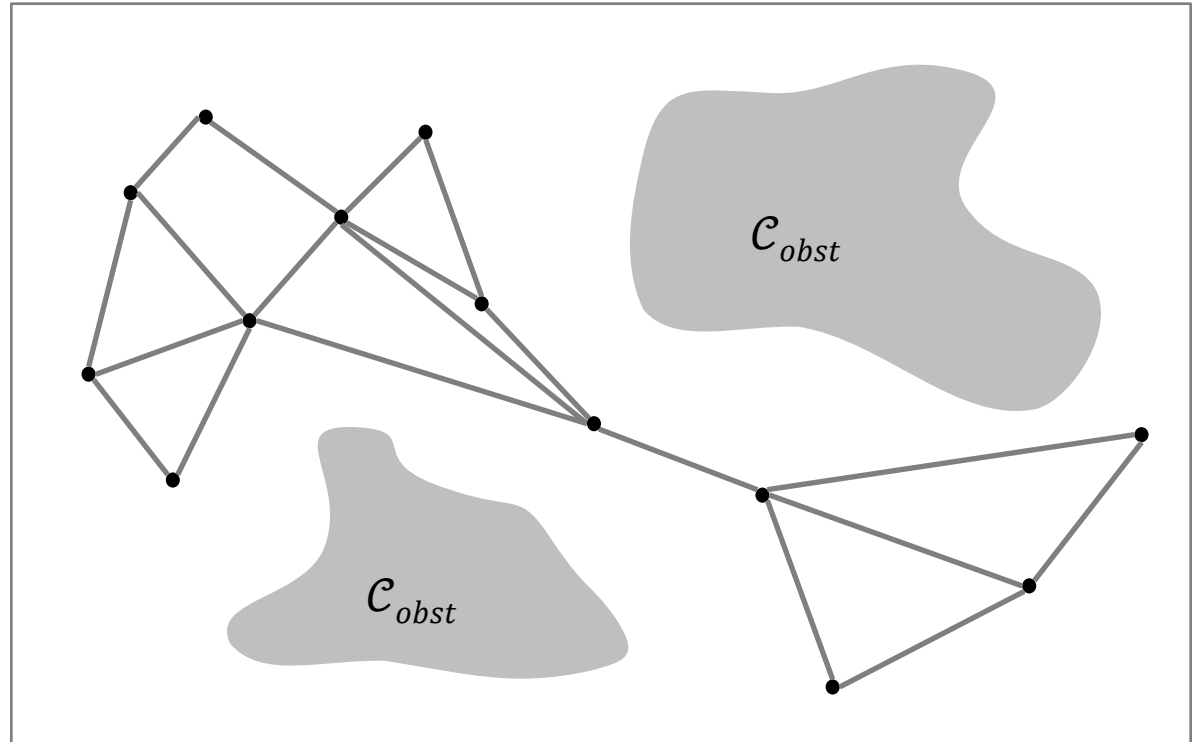
- 1) U_{An} und U_{Ab} so definieren, dass U kein lokales Minimum außer $q = q_{Ziel}$ hat
- 2) Im Suchalgorithmus Techniken zur “Flucht” aus dem lokalen Minimum anwenden

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

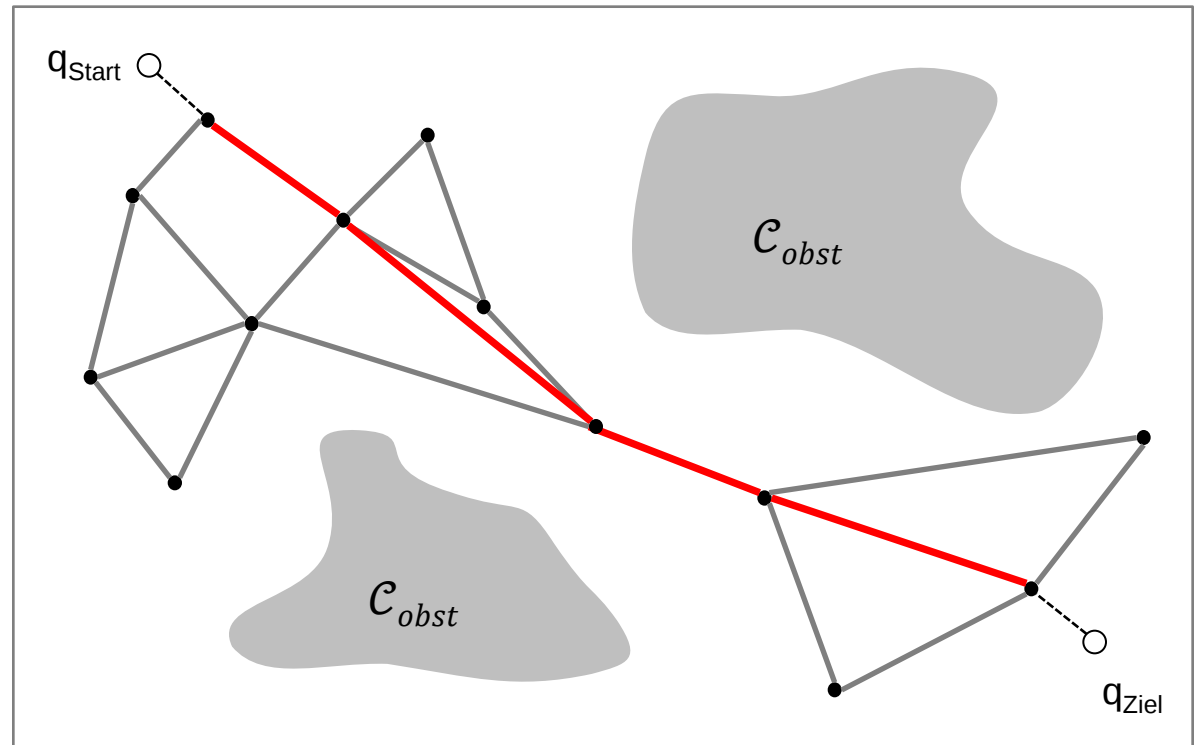
Probabilistic Roadmaps (PRM)

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



Probabilistic Roadmaps (PRM)

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



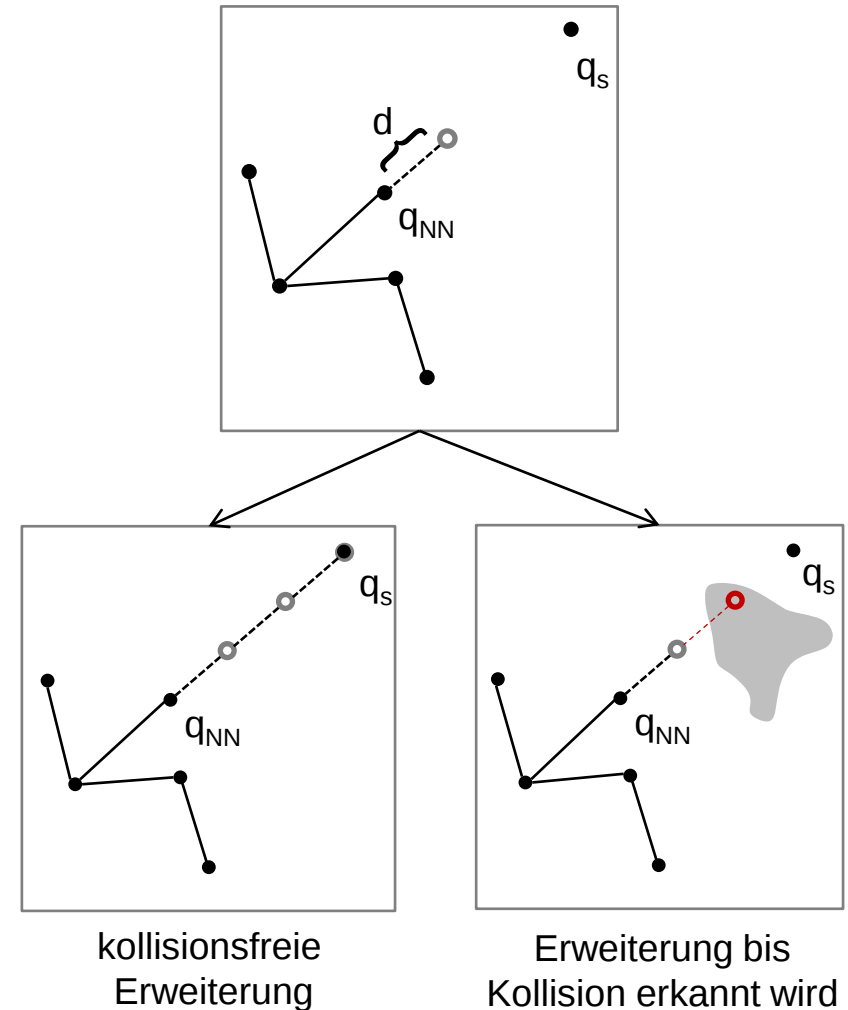
RRT: Rapidly-exploring Random Trees (1)

- 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

RRT: Rapidly-exploring Random Trees (2)

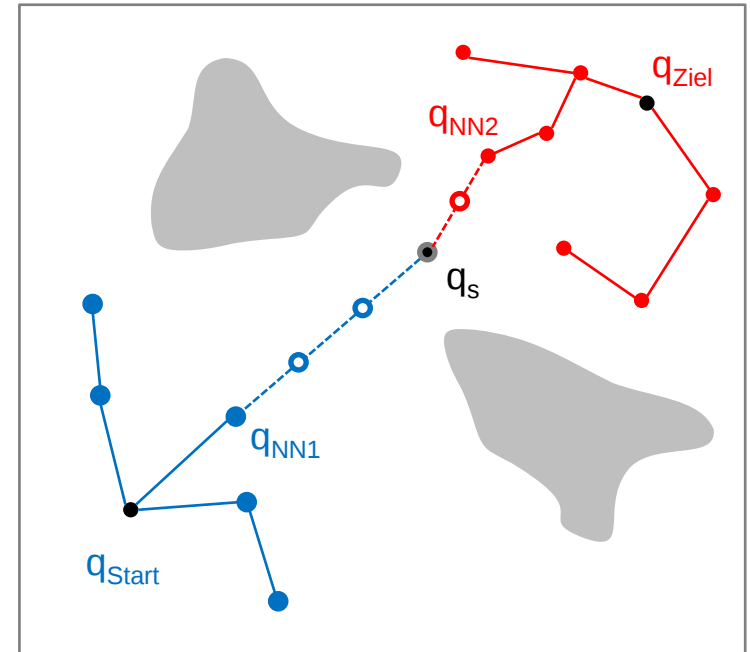
Aufbau der RRT-Bäume:

- 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



RRT: Rapidly-exploring Random Trees (3)

- Aufbau von zwei Bäumen
 - Wurzelknoten: q_{Start} bzw. q_{Ziel}
- Stichprobe q_s wird jeweils zur Erweiterung beider Bäume verwendet
- Wenn beide Bäume mit q_s verbunden werden können ist eine Lösung gefunden

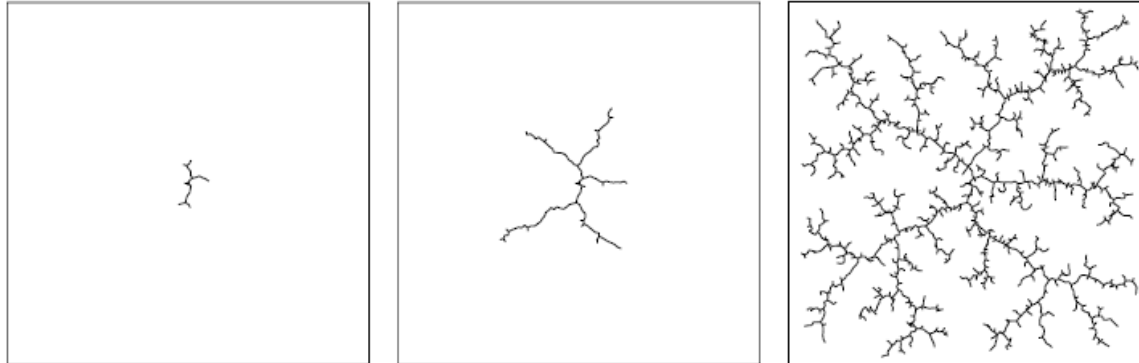


blau: Baum1, rot: Baum2

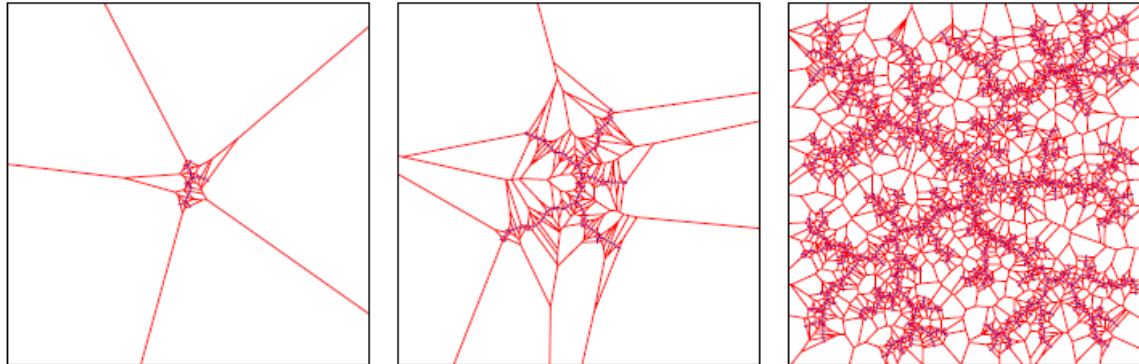
RRT: Rapidly-exploring Random Trees (4)

- Erweiterung eines Baumes

RRT-Baum



Voronoi-Regionen

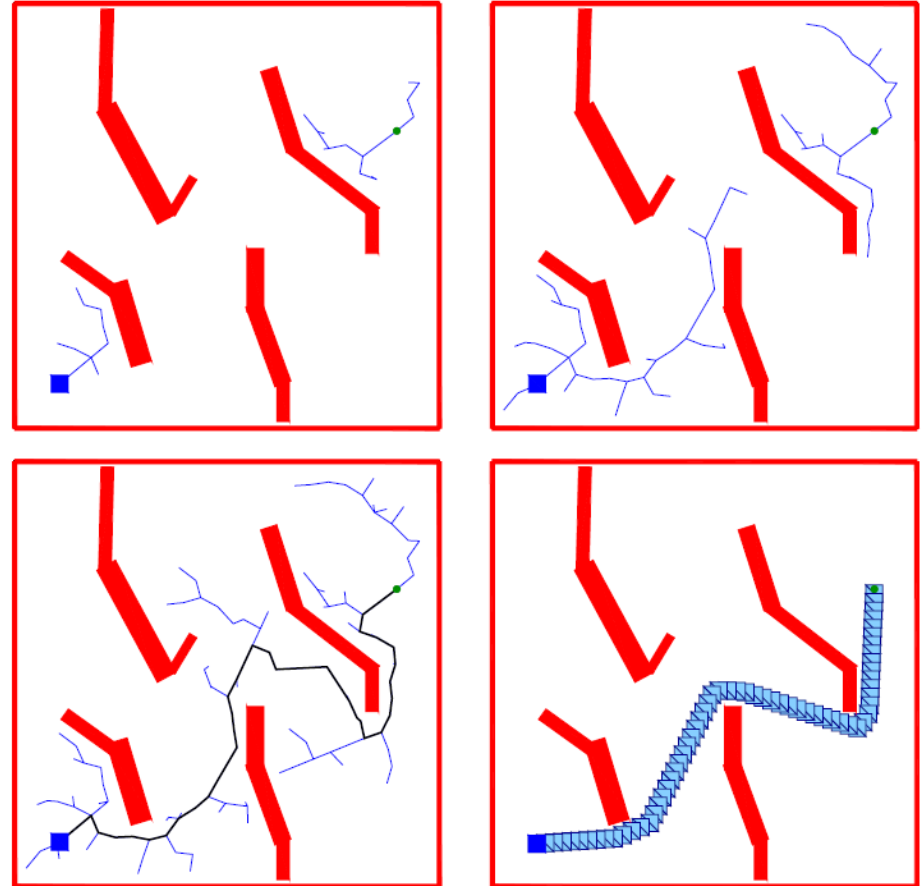


Die Wahrscheinlichkeit, dass ein Knoten erweitert wird ist proportional zur Größe der entsprechenden Voronoi-Region

RRT: Rapidly-exploring Random Trees (5)

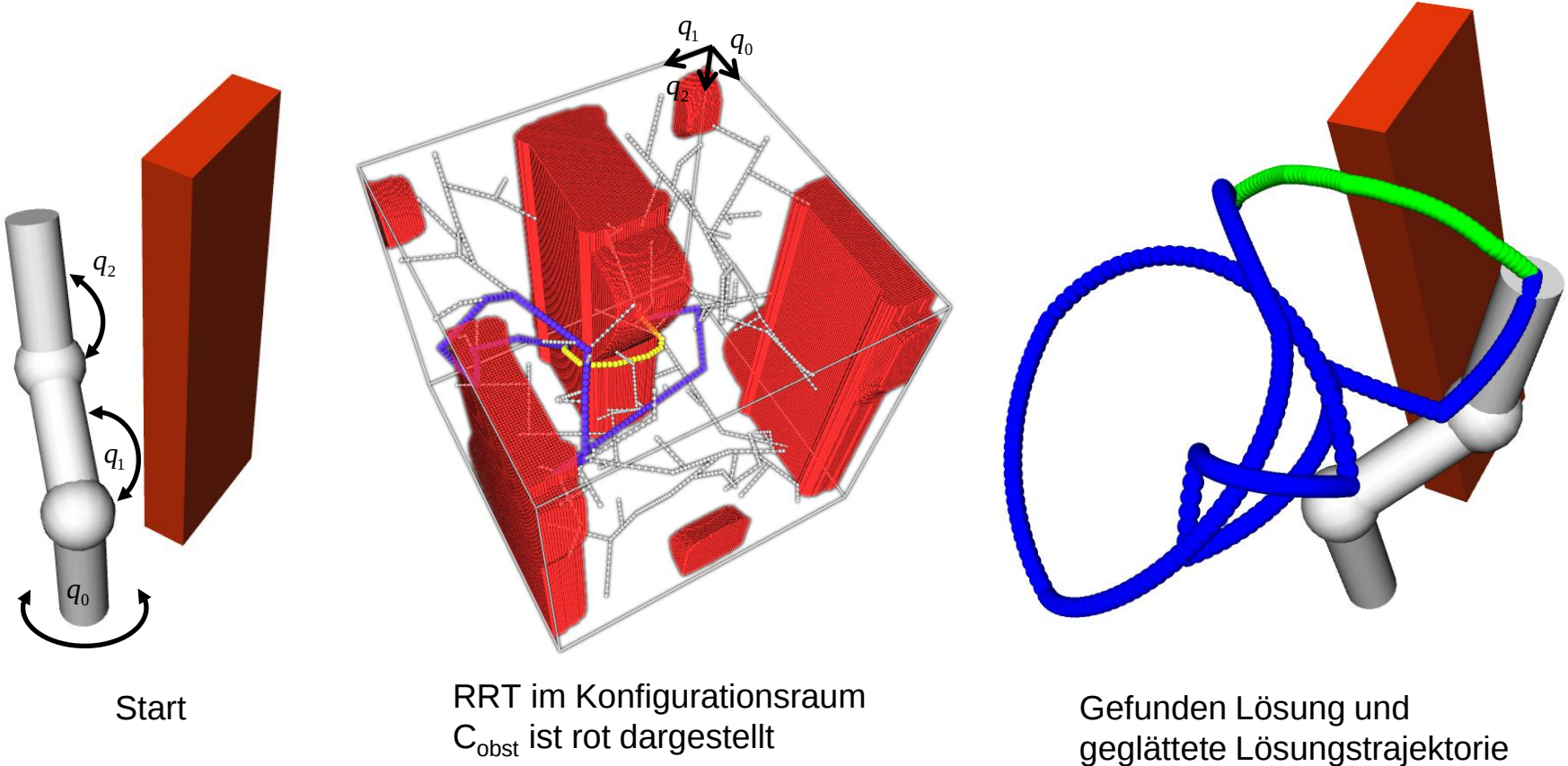
Beispiel (2D)

Glättung des
resultierenden Pfades



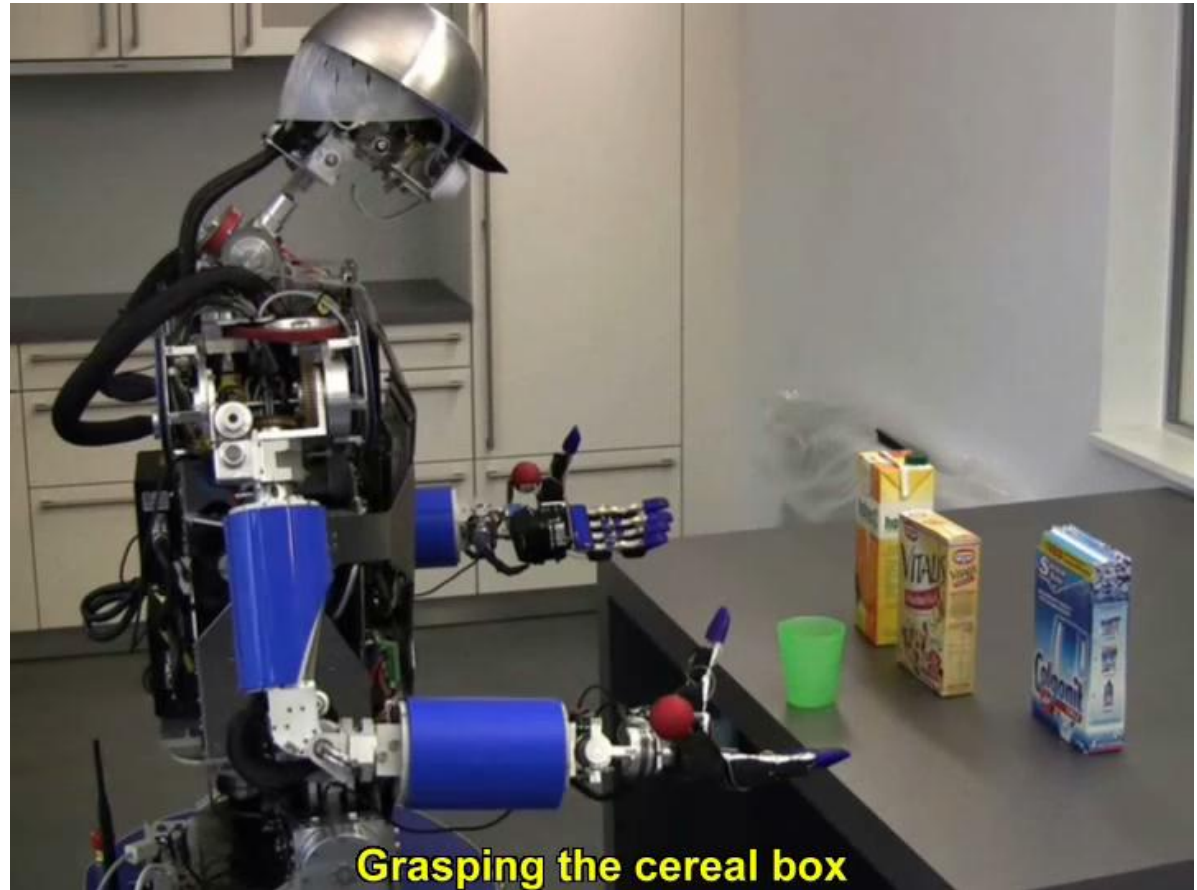
RRT: Rapidly-exploring Random Trees (6)

- Beispiel (dreidimensionaler Konfigurationsraum)



VIDEO: Anwendung RRT beim Greifen mit ARMAR-III

- Objekterkennung zur Erstellung einer virtuellen Repräsentation der Umgebung
- IK-RRT zur Bahnplanung
- Visual Servoing zur bildgestützten Ausführung der Bewegung



Vorlesungsende