

Robotik I: Einführung in die Robotik

Bewegungsplanung

Tamim Asfour, Rüdiger Dillmann

KIT-Fakultät für Informatik, Institut für Anthropomatik und Robotik (IAR)
Hochperformante Humanoide Technologien (H²T)

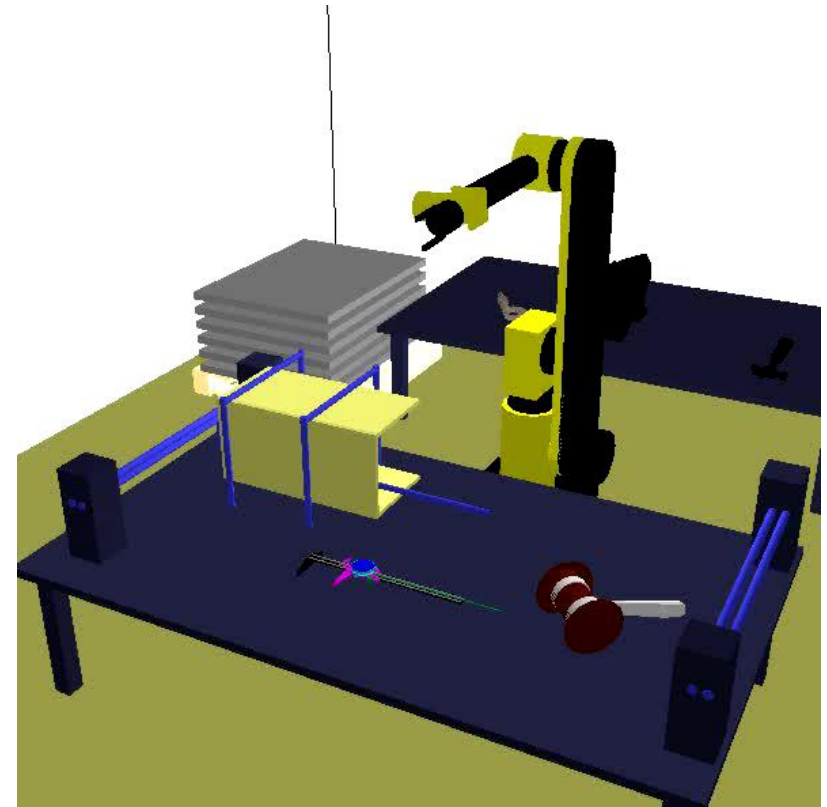
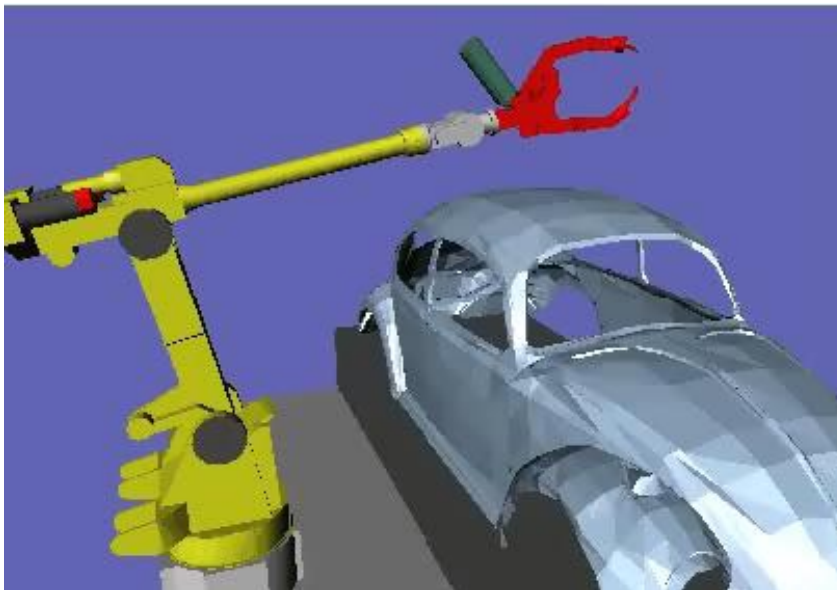


Inhalt

- Motivation
- Grundlagen der Bewegungsplanung
- Pfadplanung für mobile Roboter
- Bewegungsplanung für Manipulatoren

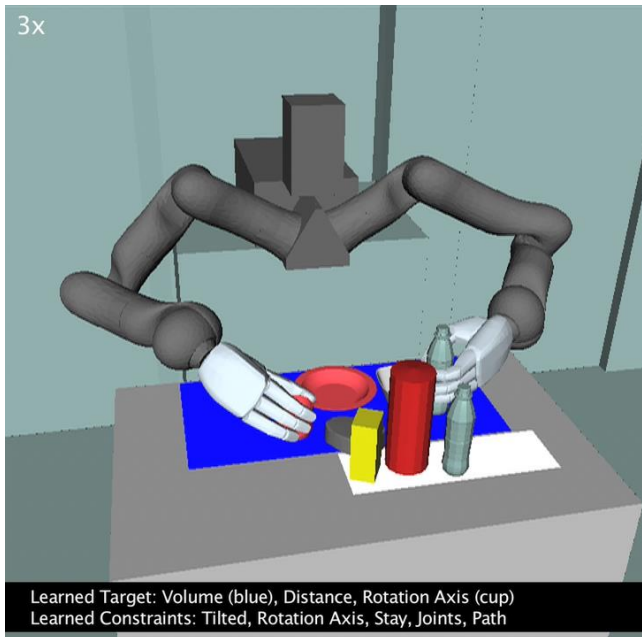
Bewegungsplanung: Motivation

Erzeugen einer **kollisionsfreien** Trajektorie unter Berücksichtigung verschiedener **Ziele** und **Einschränkungen**



Bewegungsplanung: Motivation

Erzeugen einer **kollisionsfreien** Trajektorie unter Berücksichtigung verschiedener **Ziele** und **Einschränkungen**



- Motivation
- Grundlagen der Bewegungsplanung
 - Problemstellung
 - Definitionen
 - Begriffsbildung
 - Problemklassen
- Pfadplanung für mobile Roboter
- Bewegungsplanung für Manipulatoren

Grundlagen der Bewegungsplanung: Problemstellung

■ Gegeben

- Konfigurationsraum $\mathcal{C}$
- Startkonfiguration $\mathbf{q}_{start} \in \mathcal{C}$
- Zielkonfiguration $\mathbf{q}_{ziel} \in \mathcal{C}$

■ Gesucht

- Stetige Trajektorie $\tau: [0,1] \rightarrow \mathcal{C}$ mit
 - $\tau(0) = \mathbf{q}_{start}$
 - $\tau(1) = \mathbf{q}_{ziel}$
- Unter Berücksichtigung von
 - Gütekriterien
 - Neben- und Randbedingungen
 - Zwangsbedingungen

Grundlagen der Bewegungsplanung: Definitionen I

■ Konfiguration

Eine **Konfiguration** $q \in \mathcal{C}$ beschreibt den Zustand eines Roboters

- als Lage und Orientierung im euklidischen Raum oder
- als Gelenkwinkelvektor im Gelenkwinkelraum.

■ Konfigurationsraum

Der **Konfigurationsraum** $\mathcal{C}$ eines Roboters R ist der Raum aller möglicher Konfigurationen von R .

Grundlagen der Bewegungsplanung: Definitionen II

■ Arbeitsraumhindernis

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

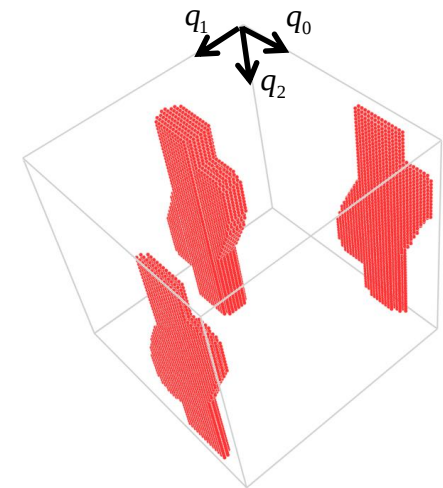
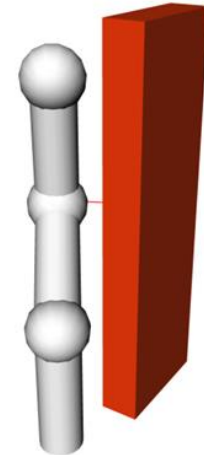
■ Konfigurationsraumhindernis

Ein **Konfigurationsraumhindernis** C_H ist die Menge aller Punkte des Konfigurationsraumes C , welche zu einer Kollision mit dem Hindernis H führen.

■ Hindernisraum

Der **Hindernisraum** C_{obs} ist die Menge aller Konfigurationsraumhindernisse

$$C_{obs} = \bigcup_i C_{H_i}$$



Grundlagen der Bewegungsplanung: Definitionen III

■ Freiraum

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

$$\mathcal{C}_{free} = \{\mathbf{q} \in \mathcal{C} \mid \mathbf{q} \notin \mathcal{C}_{obs}\} = \mathcal{C} \setminus \mathcal{C}_{obs}$$

- Aufwand für die Berechnung des Freiraums: $O(m^n)$
 - n : Anzahl der Bewegungsfreiheitsgrade des Roboters
 - m : Anzahl der Hindernisse
- Für komplexere Kinematiken kann $\mathcal{C}_{free}$ nicht effizient berechnet werden
- Verwendung **approximativer Verfahren** zur vereinfachten Repräsentation von $\mathcal{C}_{free}$

Grundlagen der Bewegungsplanung: Definitionen IV

■ Umweltmodellierung

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

■ Planungsverfahren

■ Vollständig

- Liefert immer eine korrekte Lösung
- Kann ermitteln, ob keine Lösung existiert

■ Probabilistisch vollständig

- Falls eine Lösung existiert geht die Wahrscheinlichkeit, dass eine Lösung gefunden wird, bei fortschreitender Zeit gegen 1
- Existiert keine Lösung, terminiert das Verfahren nicht

Grundlagen der Bewegungsplanung: Begriffsbildung I

■ Pfadplanung

- Starres Objekt (z.B. mobiler Roboter, autonomes Fahrzeug)
- 2D Problem (Position: x,y)
- 3D Problem (Position: x,y ; Rotation: α)
→ Piano Mover's Problem

■ Bewegungsplanung

- Mehrkörpersystem (z.B. Roboterarme, Systeme mit mehreren Robotern)
- Hochdimensionale Problemstellungen

■ Randbedingungen, auch Zwangsbedingungen

- Globale Randbedingungen: Limitieren den gültigen Konfigurationsraum
z.B. aufrechte Position des Endeffektors, maximale Motorströme, etc.
- Lokale Randbedingungen: Schränken die Übergänge zwischen Konfigurationen ein
z.B. Nicht-holonome Fahrzeuge, max. Geschwindigkeit/Beschleunigung

■ Komplexität

Allgemeine Planungsaufgaben sind PSPACE-vollständig (engl. PSPACE-complete).

- Können von deterministischen Turingmaschinen mit polynomielltem Platz (Speicherplatz) entschieden werden
- Untere und obere Schranke der Komplexität $NP \subseteq PSPACE \subseteq EXPTIME$
 - zumindest NP-hart („schweres Problem“)
 - Alle Probleme können in exponentieller Zeit gelöst werden

■ Vollständiger Algorithmus

Ein vollständiger Algorithmus findet für spezielle Planungsprobleme mindestens eine Lösung oder erkennt in endlicher Zeit, dass keine Lösung existiert.

■ Randomisierter Algorithmus

Randomisierte Algorithmen verwenden Zufallsgrößen, um den Ablauf zu steuern, wobei oft heuristische Annahmen genutzt werden, um die Berechnung zu beschleunigen.

■ Auflösungsvollständiger Algorithmus

Ist ein approximativer Algorithmus für eine diskretisierte Problemstellung vollständig, gehört er zur Klasse der auflösungsvollständigen (engl. resolution complete) Algorithmen

■ Probabilistisch-vollständiger Algorithmus

Ein probabilistisch-vollständiger (engl. probabilistically complete) Algorithmus findet mindestens eine Lösung falls sie existiert. D.h. die Wahrscheinlichkeit, dass eine Lösung gefunden wird, konvergiert mit fortlaufender Zeit gegen eins.

Allerdings kann mit probabilistisch-vollständigen Algorithmen nicht ermittelt werden, ob keine Lösung existiert.

Grundlagen der Bewegungsplanung: Problemklassen I

■ Klasse a)

Bekannt: vollständiges Umweltmodell,
vollständige Neben-, Rand- und Zwangsbedingungen

Gesucht: Kollisionsfreie Trajektorie vom Start- zum Zielzustand

■ Klasse b)

Bekannt: unvollständiges Umweltmodell,
unvollständige Neben-, Rand- und Zwangsbedingungen

Gesucht: Kollisionsfreie Trajektorie vom Start- zum Zielzustand

Problem: Kollision mit unbekannten Objekten

Grundlagen der Bewegungsplanung: Problemklassen II

■ Klasse c)

Bekannt: zeitvariantes Umweltmodell (bewegliche Hindernisse)
Gesucht: Kollisionsfreie Trajektorie vom Start- zum Zielzustand
Problem: Hindernisse in Ort und Zeit variant

■ Klasse d)

Bekannt: kein Umweltmodell
Gesucht: Kollisionsfreie Trajektorie vom Start- zum Zielzustand
Problem: Kartographieren

■ Klasse e)

Bekannt: zeitvariantes Umweltmodell
Gesucht: Trajektorie zu einem beweglichen Ziel (Rendezvous-Problem)
Problem: Zielzustand in Ort und Zeit beweglich

Inhalt

- Motivation
- Grundlagen der Bewegungsplanung
- Pfadplanung für mobile Roboter
 - Voronoi-Diagramme
 - Sichtgraphen
 - Zellzerlegung
 - Potentialfelder
 - A*
- Bewegungsplanung für Manipulatoren

Pfadplanung für mobile Roboter: Teilaufgaben

- Gegeben
 - 2D Weltmodell (z.B. Straßenkarte)
 - Start- und Zielpositionen q_{start} und q_{ziel}
- Gesucht: Günstigste Verbindung von q_{start} nach q_{ziel}
- Ansatz:
 - **Konstruiere** ein Netz W von Wegen in C_{free}
 - Bilde q_{start} und q_{ziel} auf die nächsten Knoten q'_{start} und q'_{ziel} in W ab
 - **Suche** in W einen Weg von q'_{start} nach q'_{ziel}
 - Finde einen Weg zwischen q_{start} und q'_{start} , sowie zwischen q'_{ziel} und q_{ziel}

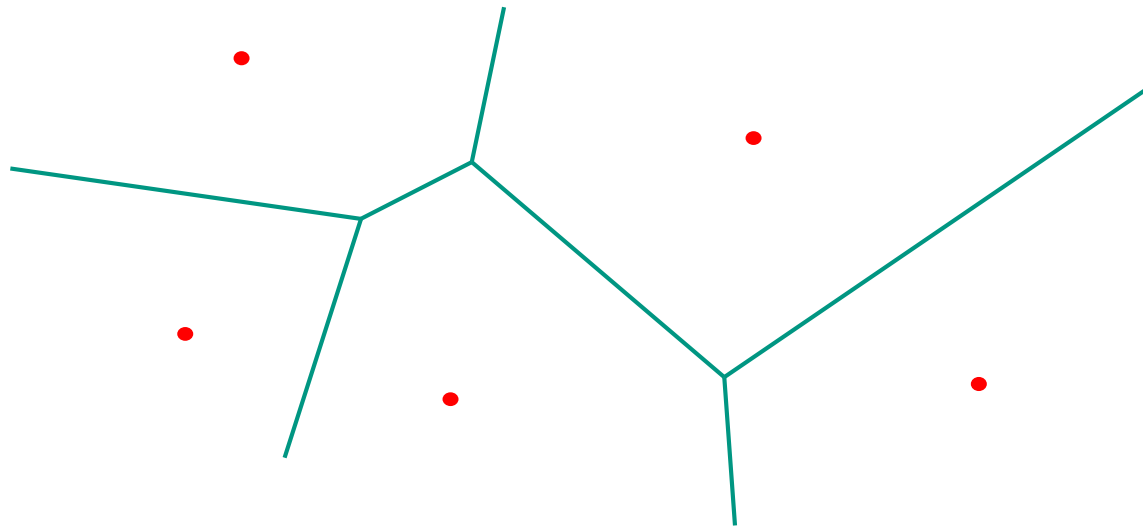
Pfadplanung für mobile Roboter: Teilaufgaben

- Konstruktion des Wegenetzes W
 - Retraktionsverfahren, z.B. Voronoi-Diagramm
 - Sichtgraphen
 - Zellzerlegung

- Suche in W
 - Euklidischer Abstand
 - Potentialfelder
 - Baumsuche
 - A^*

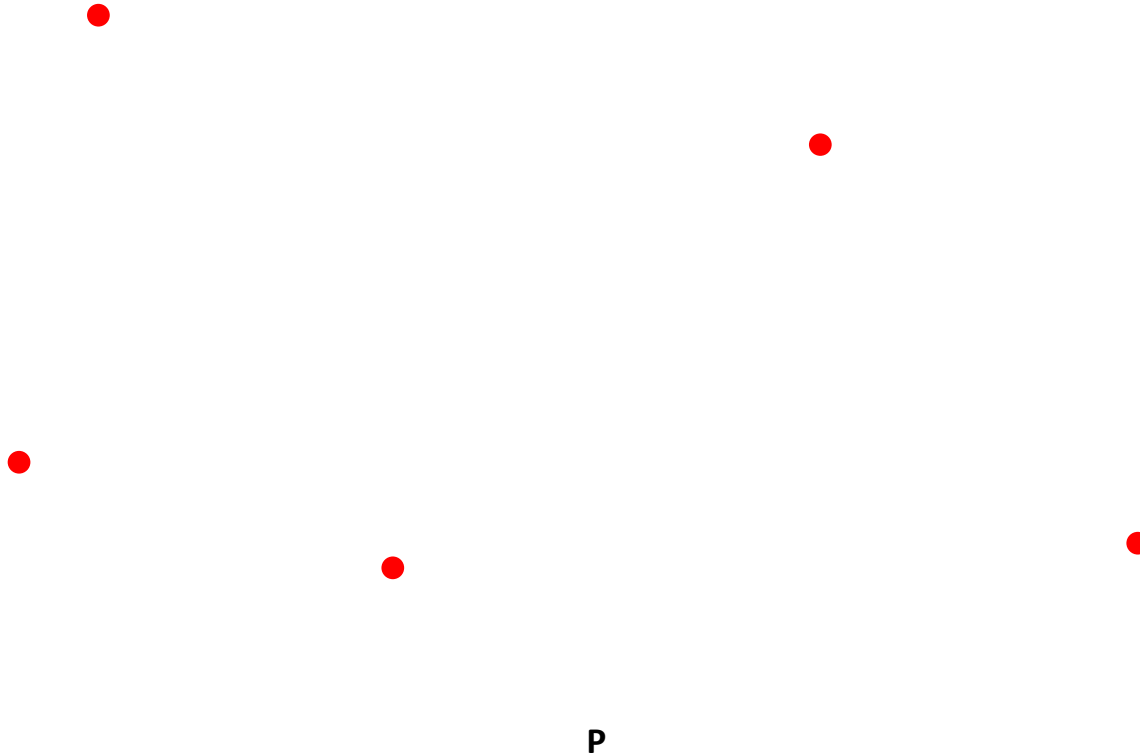
Voronoi-Diagramme

- Visualisiert **die Zerlegung eines Raumes** in Regionen basierend auf vorgegebenen Punkten.
- Eine **Region** ist definiert als die **Menge aller Punkte**, deren **Abstand zum Zentrum geringer** ist **als zu allen anderen Zentren**.
- Alle **Punkte auf der Grenze** zwischen zwei Regionen besitzen den **gleichen Abstand** zum eigenen und zum **benachbarten Zentrum**.



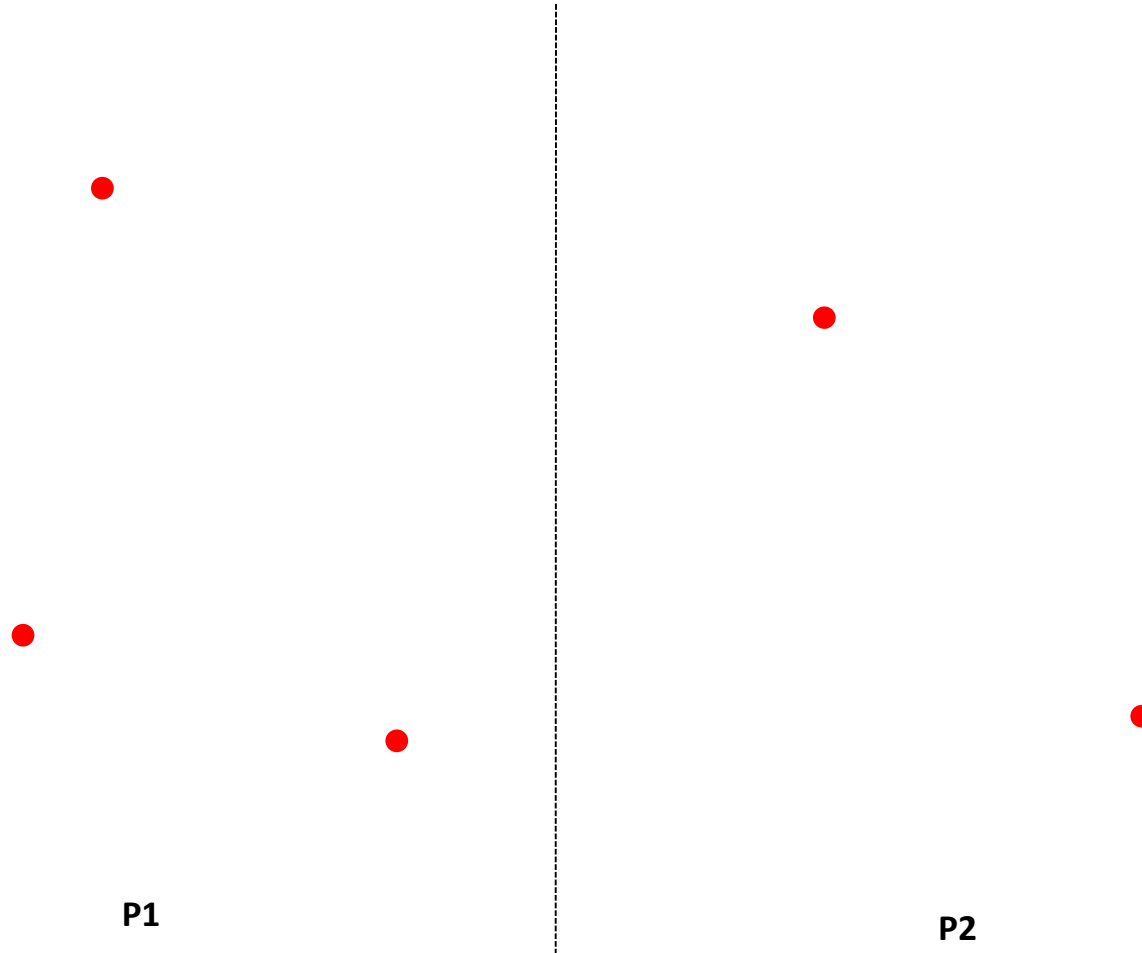
Voronoi-Diagramm: Konstruktion I

■ Gegebene **Punktmenge P**



Voronoi-Diagramm: Konstruktion II

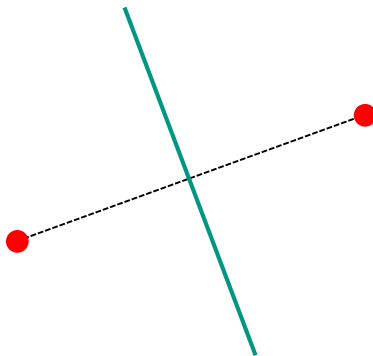
■ Teile P in zwei etwa gleich große Teilmengen P_1 und P_2



Voronoi-Diagramm: Konstruktion III

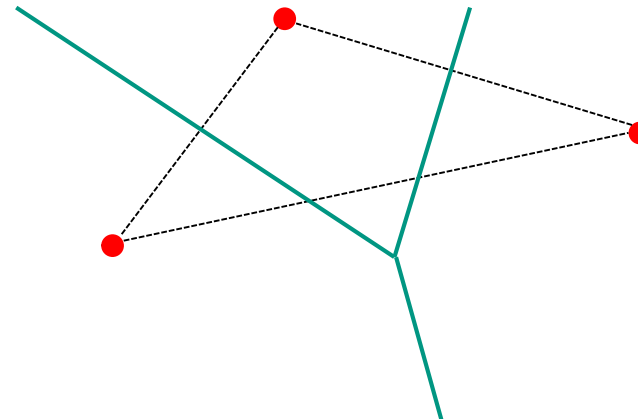
- Durch **rekursive Unterteilung** der Punktmengen kann das Problem der Erstellung eines Voronoi-Diagramms auf **zwei einfache Fälle** reduziert werden.

Fall 1: 2 Punkte



Die **Mittelsenkrechte** bildet das Voronoi-Diagramm

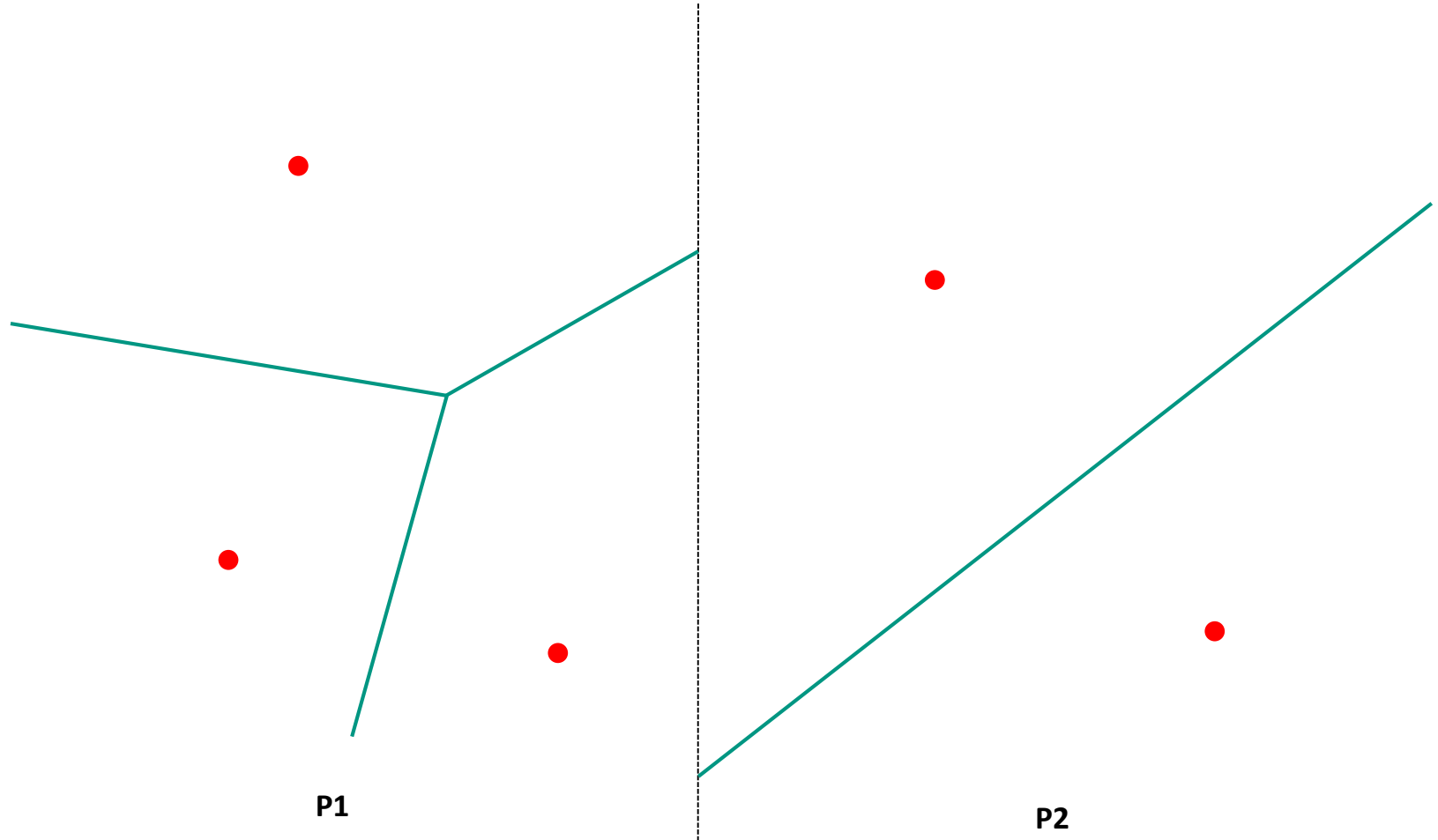
Fall 2: 3 Punkte



Die **Mittelsenkrechten aller Punktpaare** werden am gemeinsamen **Schnittpunkt abgeschnitten**

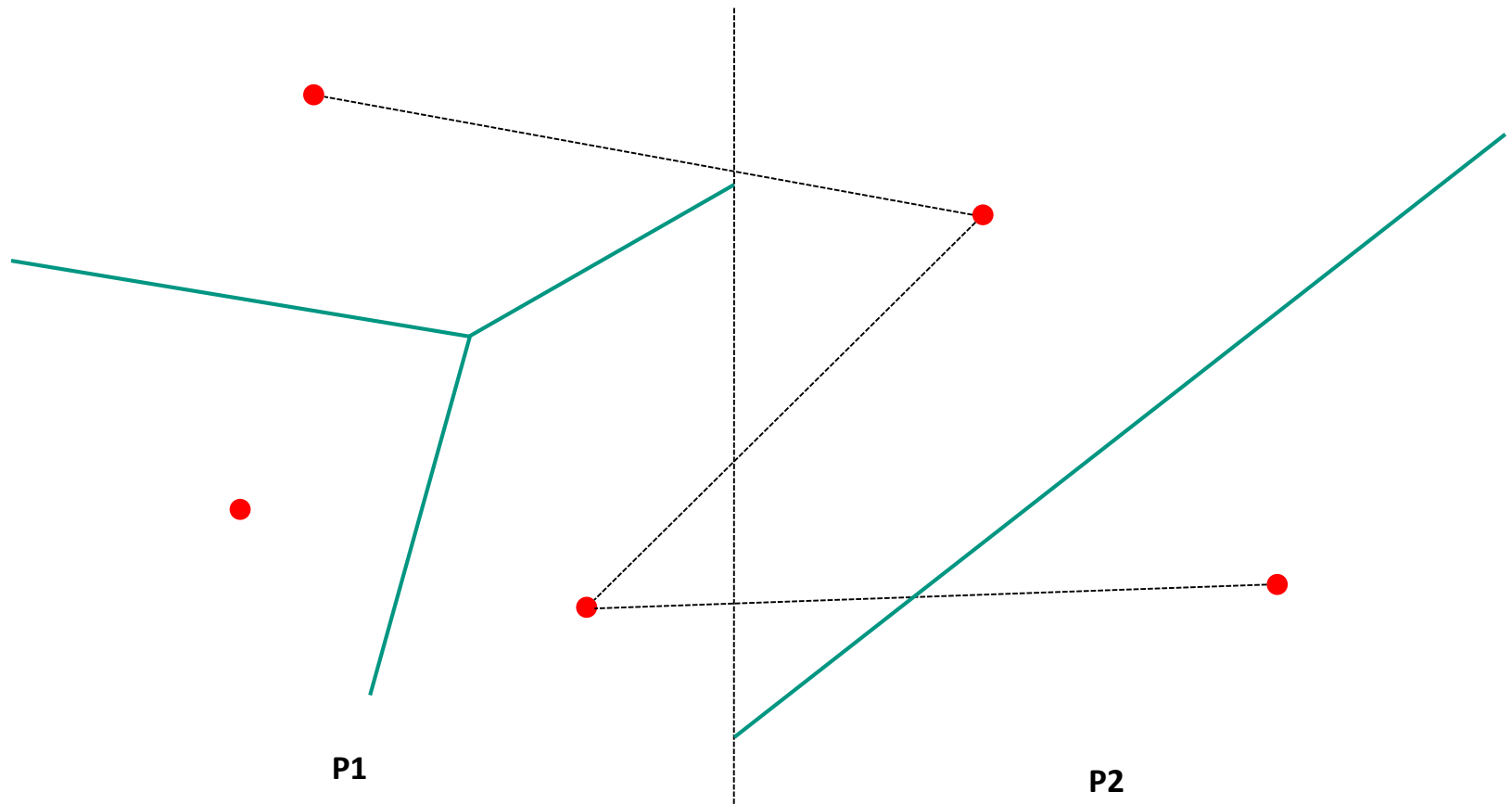
Voronoi-Diagramm: Konstruktion IV

■ Konstruiere **Voronoi-Diagramme** für **P1** und **P2**



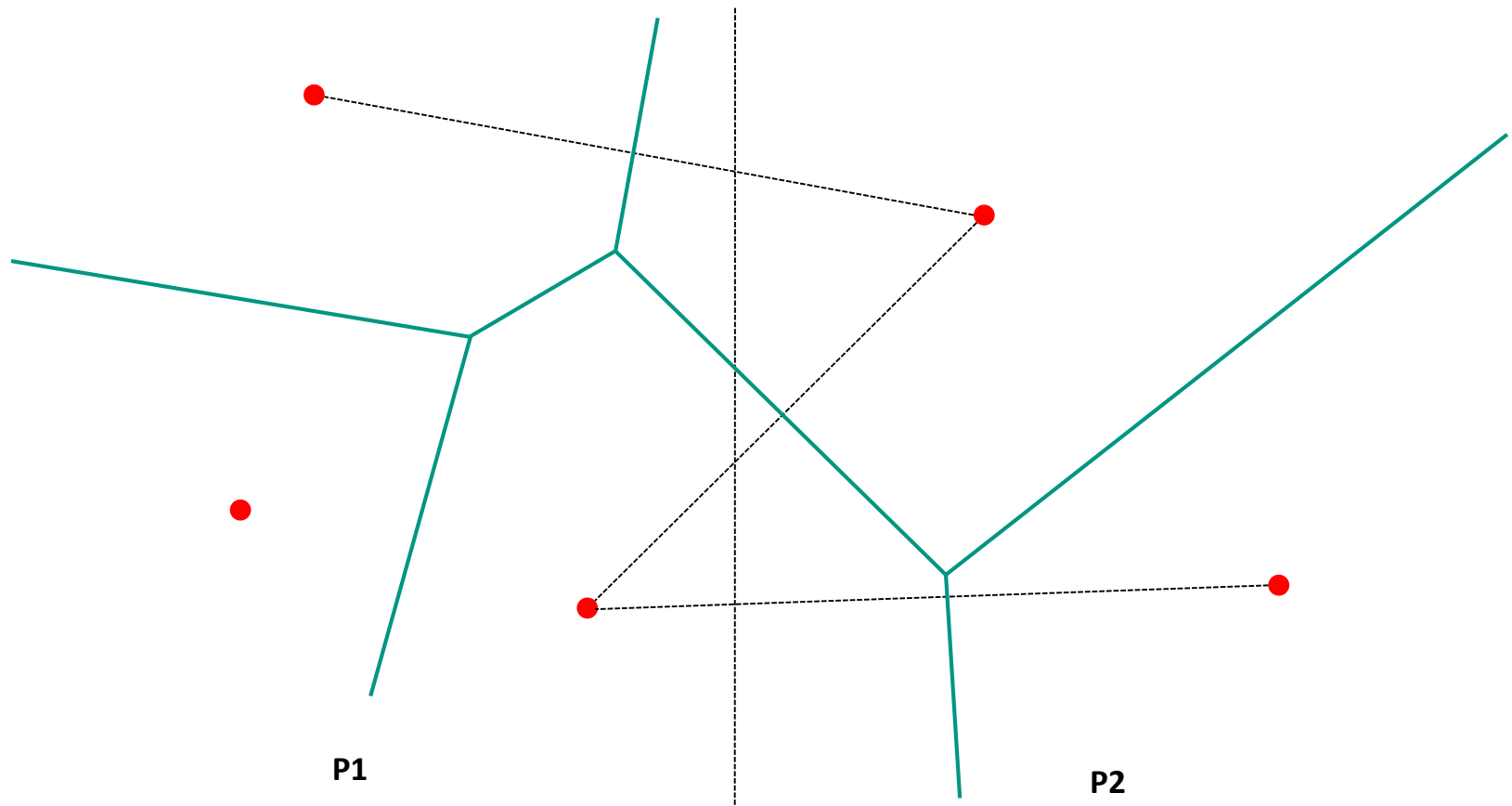
Voronoi-Diagramm: Konstruktion V

- **Verschmelze** die Voronoi-Diagramme für **P1** und **P2**
 - **Verbinden** der nächsten **Nachbarn** entlang der Trennungslinie



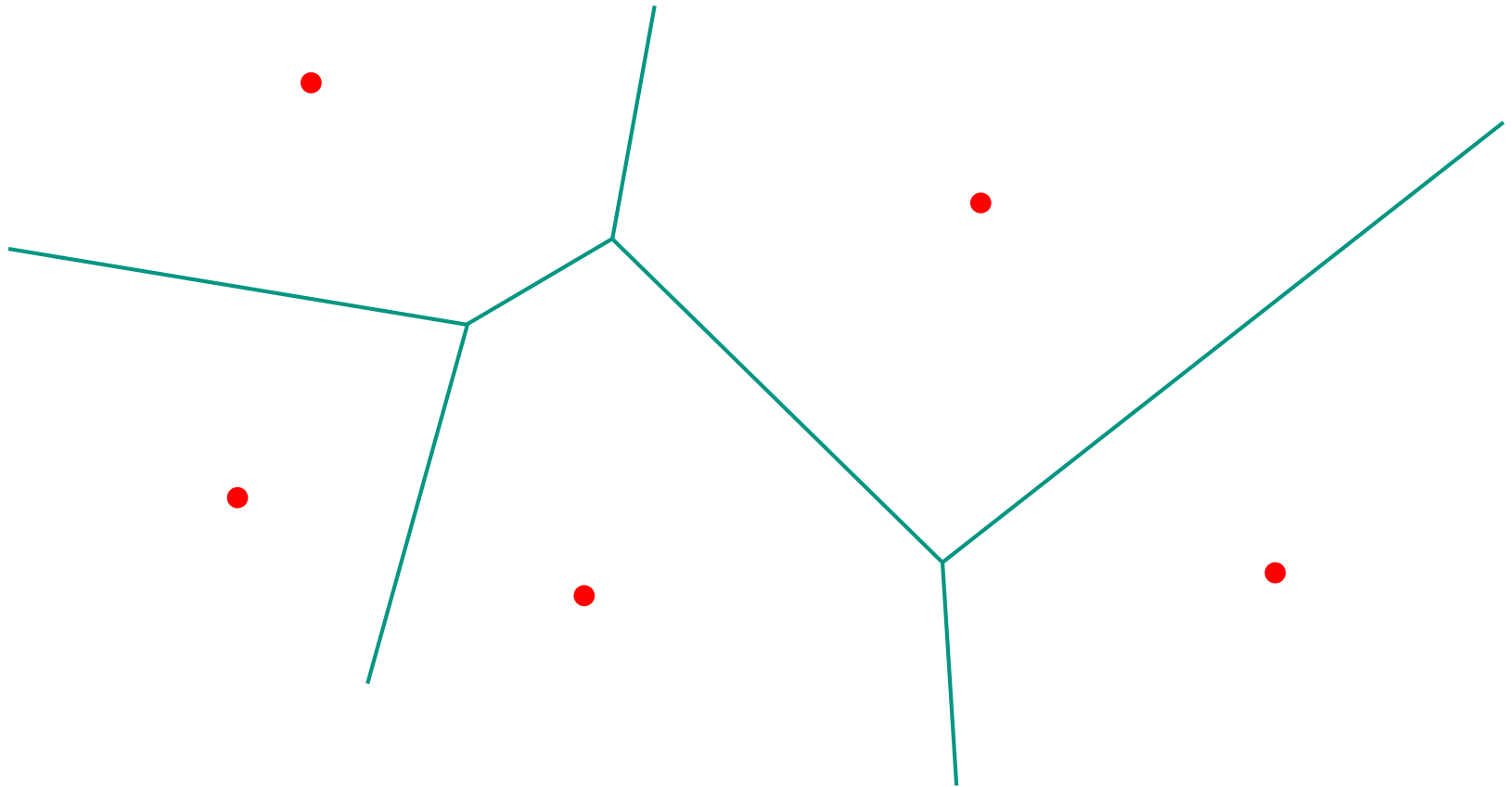
Voronoi-Diagramm: Konstruktion VI

- **Verschmelze** die Voronoi-Diagramme für **P1** und **P2**
 - Einzeichnen und Abschneiden der neuen **Mittelsenkrechten**

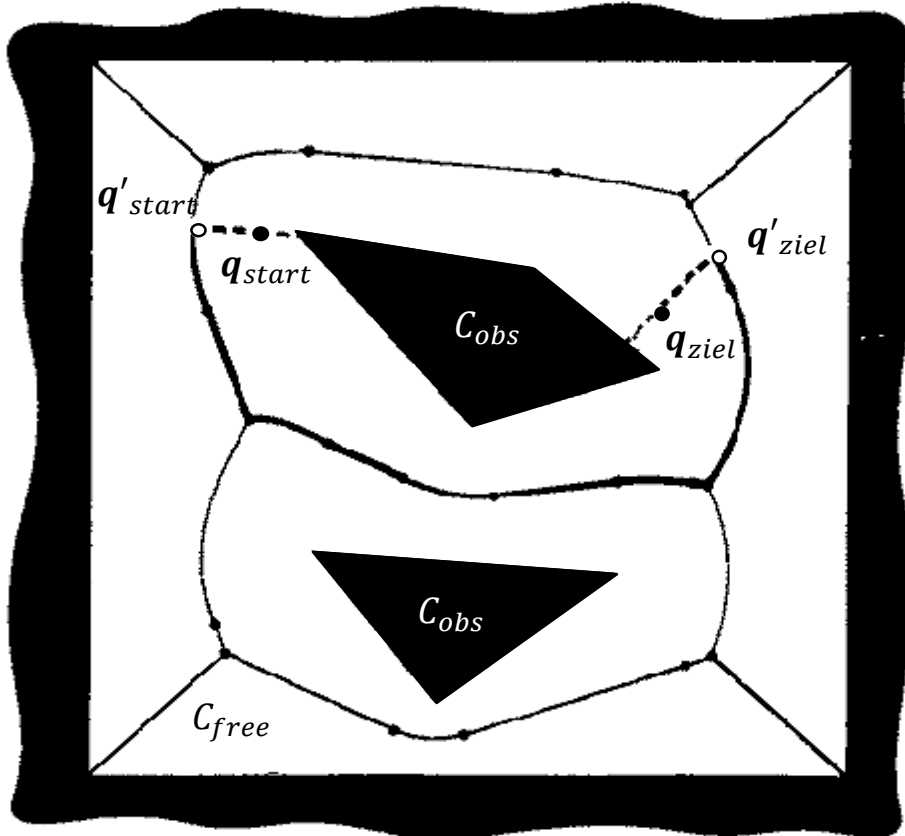


Voronoi-Diagramm: Konstruktion VII

■ Fertiges Voronoi-Diagramm für P



Voronoi-Diagramme: Vor- und Nachteile



■ Vorteile:

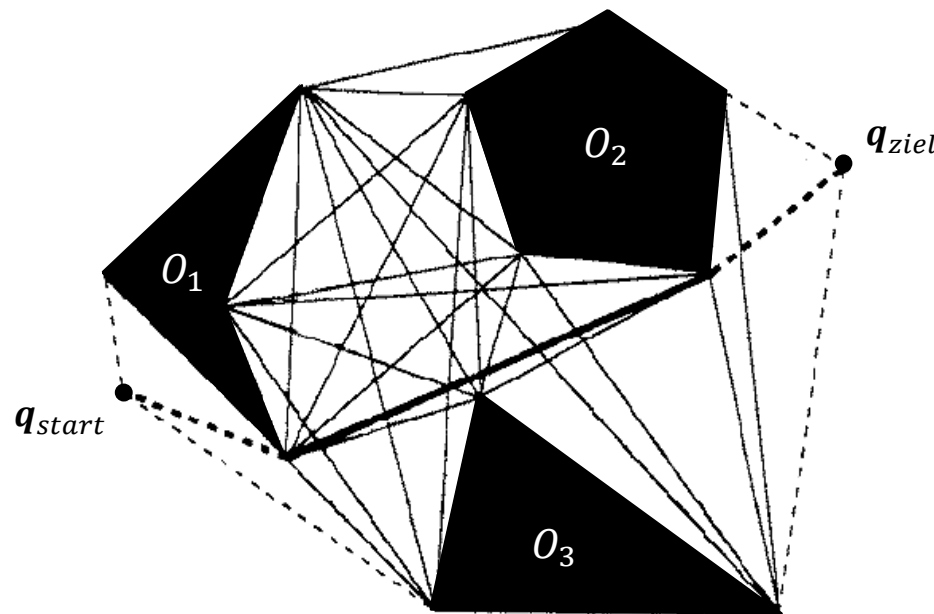
- **Maximaler Abstand** zu Hindernissen
- Ein Roboter kann mit Hilfe von Abstandssensoren **leicht prüfen** ob der richtige Weg abgefahren wird

■ Nachteile

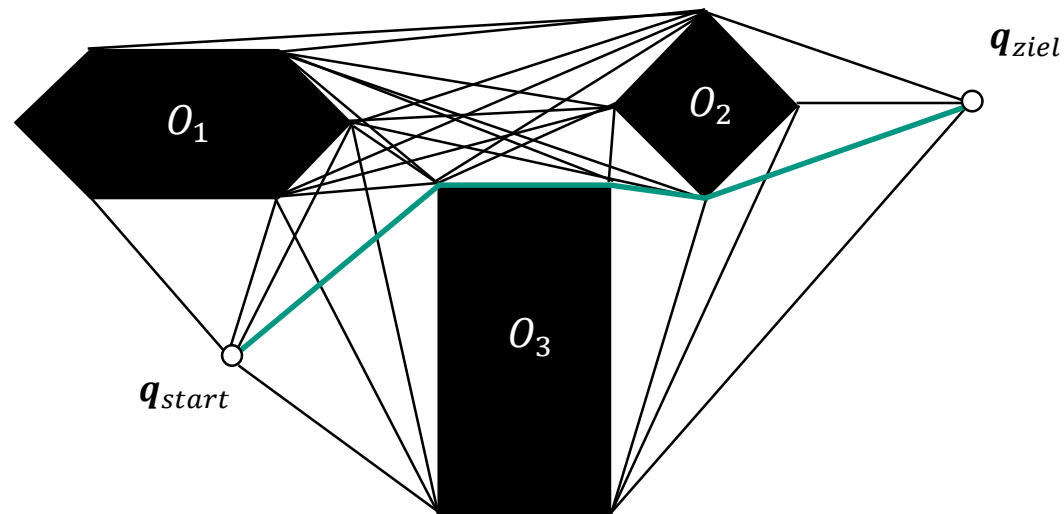
- In der Regel ist der **Weg nicht der kürzeste**.
- Bei **wenigen Hindernissen** werden nur **wenige Wege** generiert.

Sichtgraphen: Konstruktion

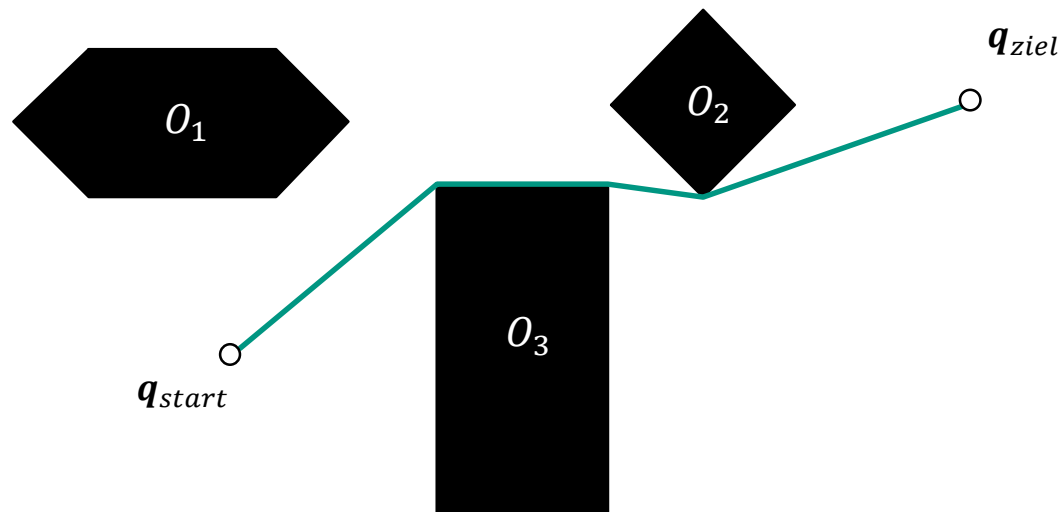
- **Verbinde** jedes Paar von **Eckpunkten** auf dem **Rand** von $\mathcal{C}_{\text{free}}$ durch ein gerades Liniensegment, wenn das Segment **kein Hindernis** schneidet
- Verbinde q_{start} und q_{ziel} analog



Sichtgraphen: Beispiel I



Sichtgraphen: Beispiel II



Sichtgraphen: Vor- und Nachteile

■ Vorteile:

- Wenn ein **Weg gefunden** ist, ist es auch der **kürzeste** Weg.
- Methode ist **exakt**, wenn nur **zwei translatorische Freiheitsgrade** existieren und sowohl Roboter als auch Hindernisse durch **konvexe Polygone** dargestellt werden können.

■ Nachteile:

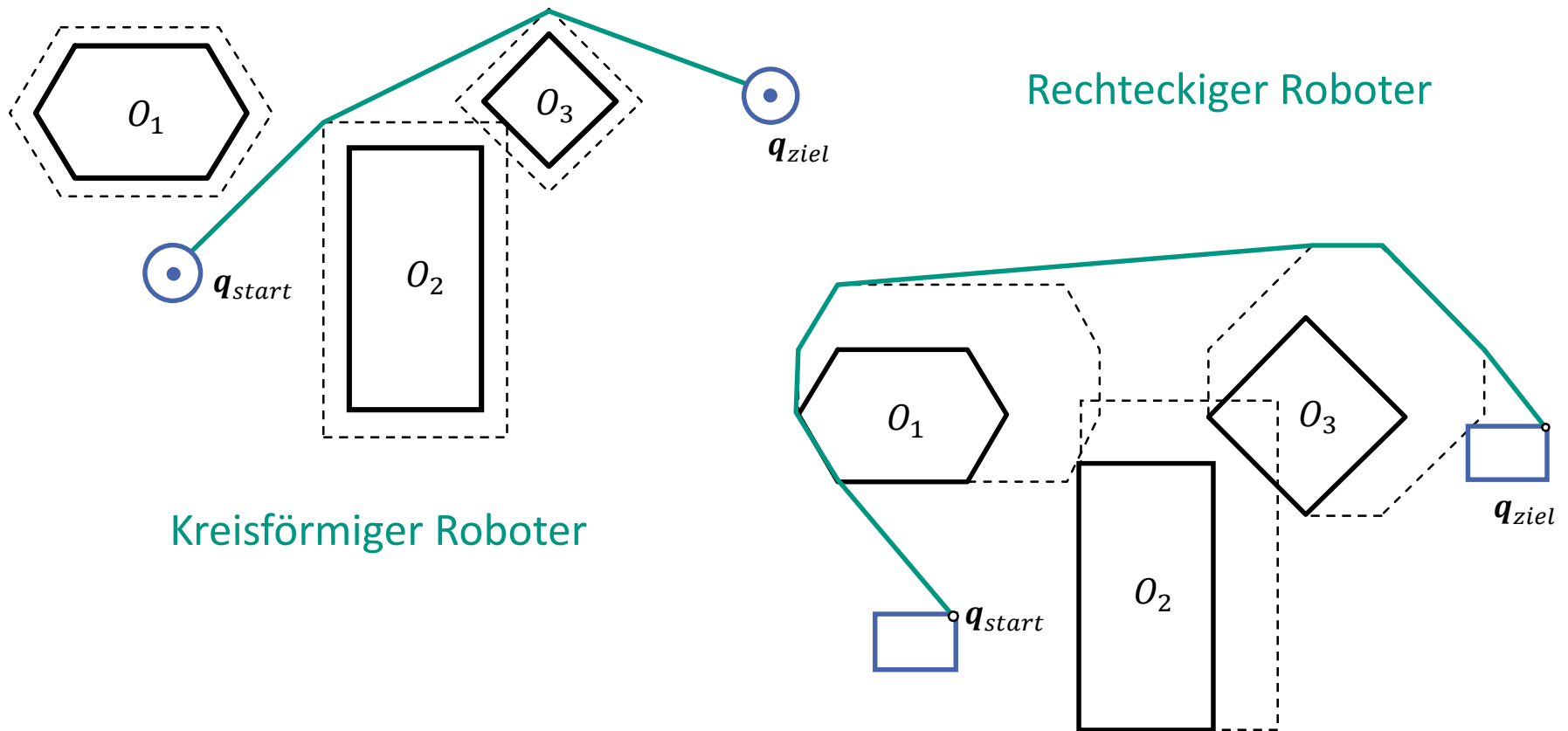
- Wege sind nicht zwingend kollisionsfrei, da **Hinderniskanten** auch **Wegsegmente** sein können.

Abhilfe durch **Erweiterung der Hindernisse**.

- Methode auch **im $\mathbb{R}^3$** anwendbar, jedoch sind die gefundenen Wege i. A. **keine kürzesten Wege** mehr.

Sichtgraphen: Erweiterung der Hindernisse

- Hindernisse werden um Roboterform erweitert



Zellzerlegung

■ Vorgehen:

1. Zerlege C_{free} in Zellen, so dass ein Weg zwischen zwei Konfigurationen innerhalb einer Zelle leicht zu finden ist
2. Stelle die Nachbarschaft (Adjazenz) in einem Graphen dar
3. Suche den optimalen Weg von $\mathbf{q}_{start}$ nach $\mathbf{q}_{ziel}$ in dem Graphen

■ Es gibt zwei Zerlegungsarten:

- Exakte Zerlegung
- Approximative Zerlegung

Exakte Zellzerlegung

■ Zerlegung des Freiraumes C_{free} in Zellen Z_i , so dass:

■ Die Zellen sich nicht überlappen

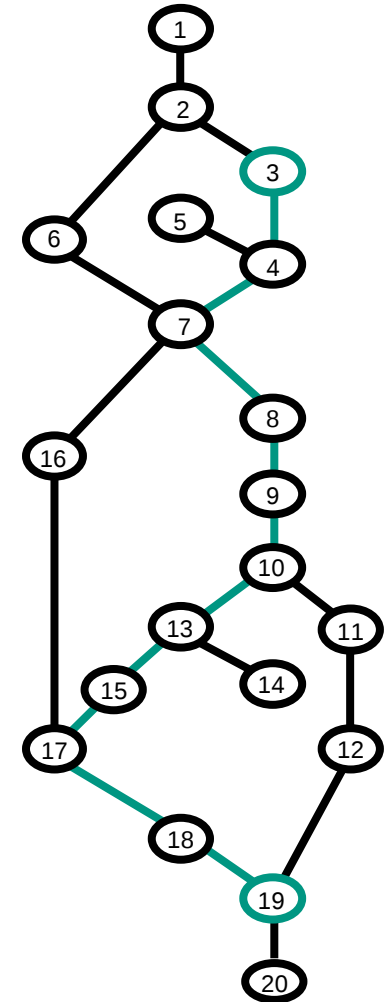
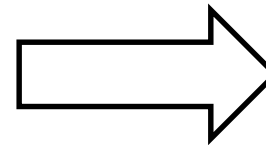
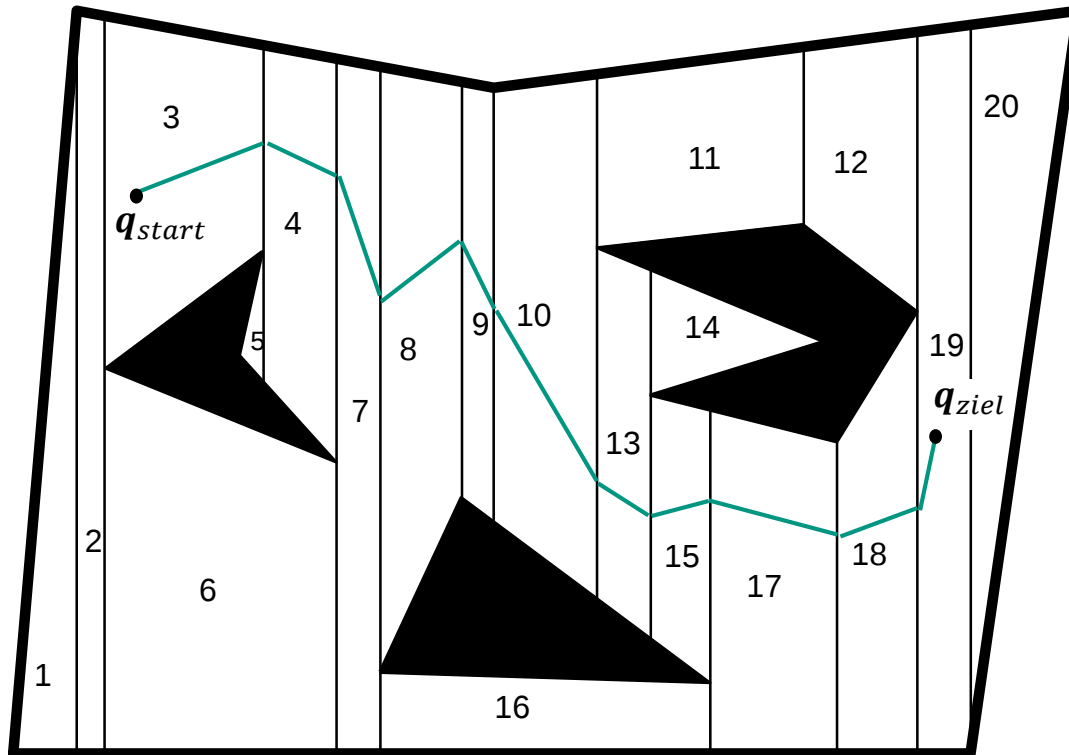
$$\forall i, k, i \neq k: Z_i \cap Z_k = \emptyset$$

■ Die Vereinigungsmenge aller Z_i ist C_{free}

$$\bigcup_{i=1}^n Z_i = C_{free}$$

Exakte Zellzerlegung: Beispiel

■ Exakte Zellzerlegung mit Line-Sweep



Approximative Zellzerlegung

■ Vorgehen:

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

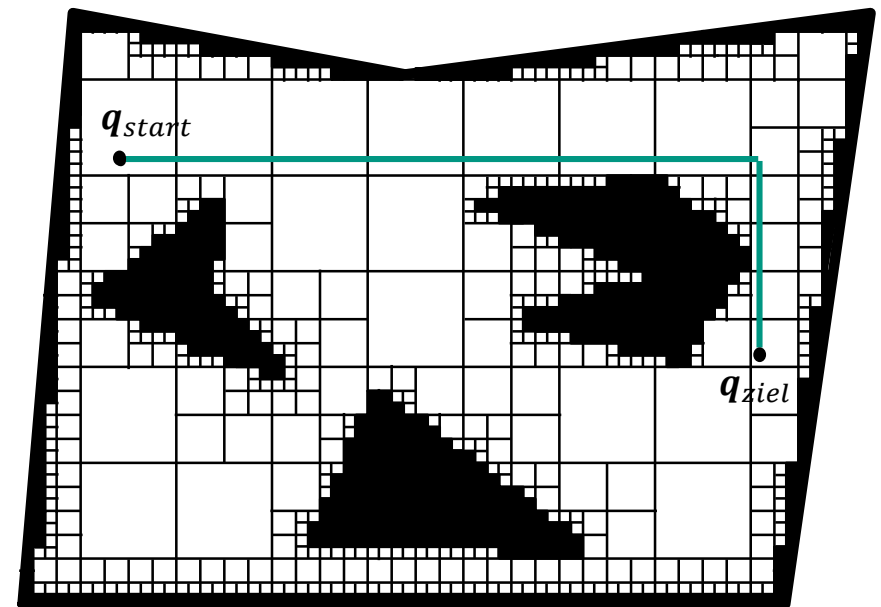
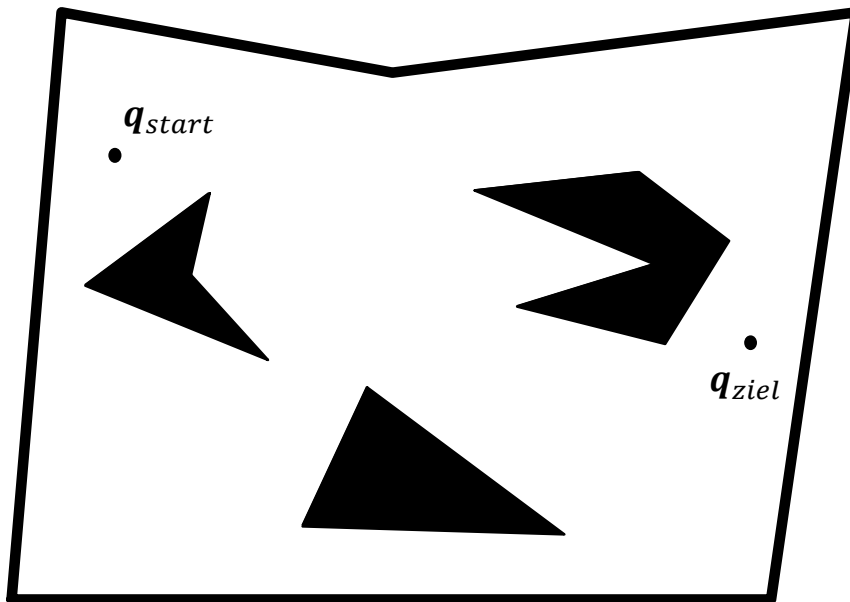
■ Vorteil

- Einfache Zerlegung und damit einfachere Wegsuche

■ Nachteil

- Der Freiraum kann i.A. nur annähernd beschrieben werden

Approximative Zellzerlegung: Beispiel



Bewegungsplanung für mobile Roboter: Teilaufgaben

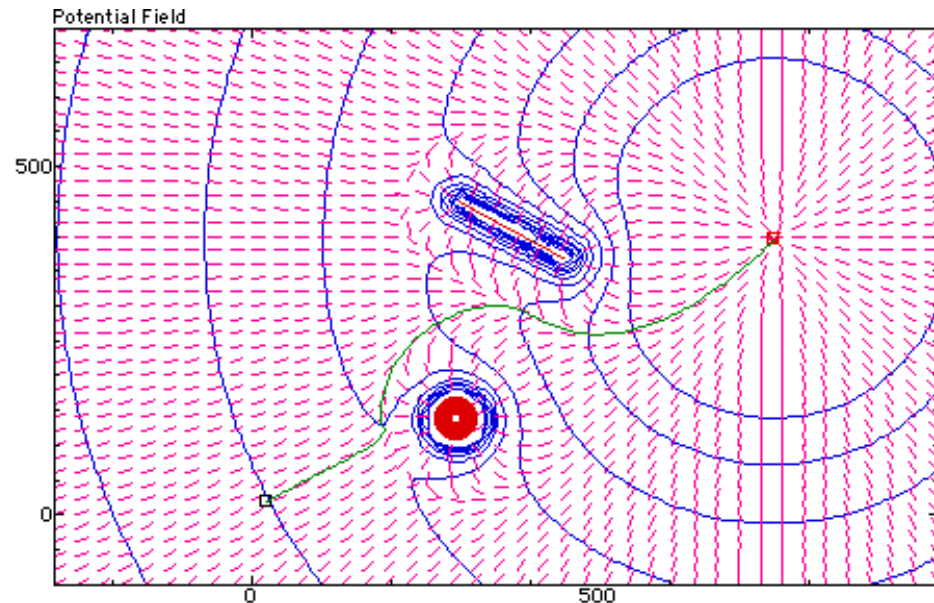
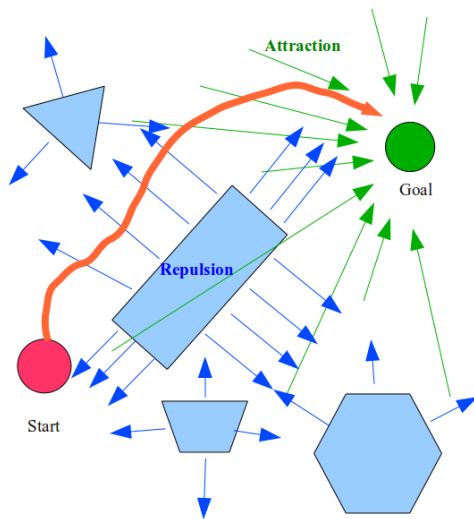
- Konstruktion des Wegenetzes W
 - Retraktionsverfahren, z.B. Voronoi-Diagramm
 - Sichtgraphen
 - Zellzerlegung

- Suche in W
 - Euklidischer Abstand
 - Potentialfelder
 - Baumsuche
 - A^*

Potentialfield Methode

Methode entwickelt für

- Bewegungsplanung [Kathib 1986]
- SLAM bei mobilen Roboter, d.h. [Prestes 2002]



O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots" , International Journal on Robotics Research (IJRR), 5(1):90--98, Spring, 1986

Potentialfelder I

- Der Roboter bewegt sich unter dem Einfluss von Kräften, welche ein Potentialfeld auf ihn ausübt

- **Definition:**

- Ein Potentialfeld U ist eine Skalarfunktion über dem Freiraum

$$U: C_{free} \rightarrow \mathbb{R}$$

- Die Kraft in einem Punkt $\mathbf{q}$ des Potentialfeldes ist der negative Gradient in diesem Punkt

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

Potentialfelder II

■ Abstoßendes Potenzial

- Hindernisse erzeugen ein abstoßendes Potential
- In großem Abstand zu Hindernissen ($> \rho_0$) soll der Roboter nicht beeinflusst werden

■ Beispiel:

$$U_{ab}(\mathbf{q}) = \begin{cases} \frac{1}{2} v \left(\frac{1}{\rho(\mathbf{q}, \mathbf{q}_{obs})} - \frac{1}{\rho_0} \right)^2 & \text{für } \rho(\mathbf{q}, \mathbf{q}_{obs}) \leq \rho_0 \\ 0 & \text{sonst} \end{cases}$$

$\rho(\mathbf{q}, \mathbf{q}_{obs}) = \|\mathbf{q} - \mathbf{q}_{obs}\|$ ist die minimale Distanz zwischen Roboter und Hindernis

$$F_{ab} = -\nabla U_{ab} = v \left(\frac{1}{\rho(\mathbf{q}, \mathbf{q}_{obs})} - \frac{1}{\rho_0} \right) \cdot \frac{1}{\rho(\mathbf{q}, \mathbf{q}_{obs})^2} \cdot \frac{\mathbf{q} - \mathbf{q}_{obs}}{\rho(\mathbf{q}, \mathbf{q}_{obs})}$$

Potentialfelder III

■ Anziehendes Potential

- Es soll möglichst nur ein Minimum in $\mathbf{q}_{ziel}$ geben

■ Lineare Funktion der Distanz zum Ziel:

$$U_{ab}(\mathbf{q}) = \xi \|\mathbf{q} - \mathbf{q}_{ziel}\|$$

$$F_{ab}(\mathbf{q}) = -\nabla U_{ab}(\mathbf{q}) = -\xi \frac{\mathbf{q} - \mathbf{q}_{ziel}}{\|\mathbf{q} - \mathbf{q}_{ziel}\|}$$

- Für kleine Distanzen wird die Kraft sehr groß

Potentialfelder IV

■ Quadratische Funktion der Distanz

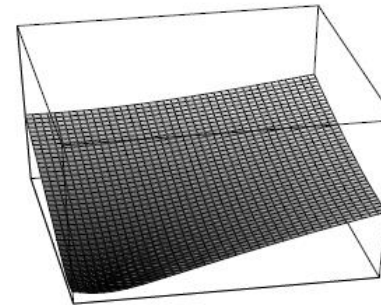
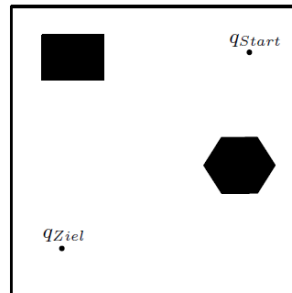
$$U_{an}(\mathbf{q}) = \xi \cdot \frac{1}{2} \|\mathbf{q} - \mathbf{q}_{ziel}\|^2$$

$$F_{an}(\mathbf{q}) = -\nabla U_{an}(\mathbf{q}) = -\xi(\mathbf{q} - \mathbf{q}_{ziel})$$

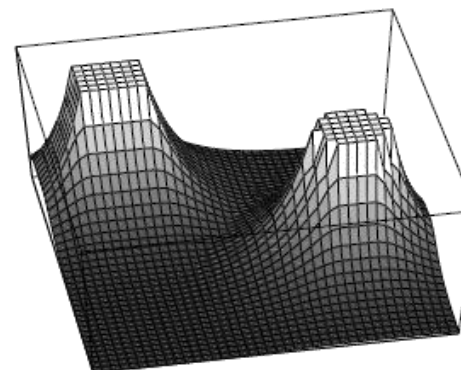
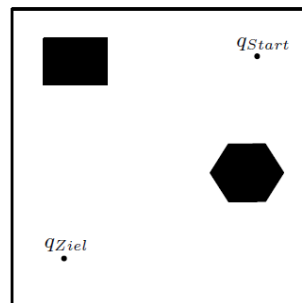
- Oft wird die Kombination aus linearer und quadratischer Funktion verwendet
 - Lineare Funktion, wenn weit vom Ziel entfernt
 - Quadratisch Funktion, wenn nah am Ziel

Potentialfelder: Beispiel I

- Die **Zielstellung** q_{ziel} hat das anziehende Potential U_{an}

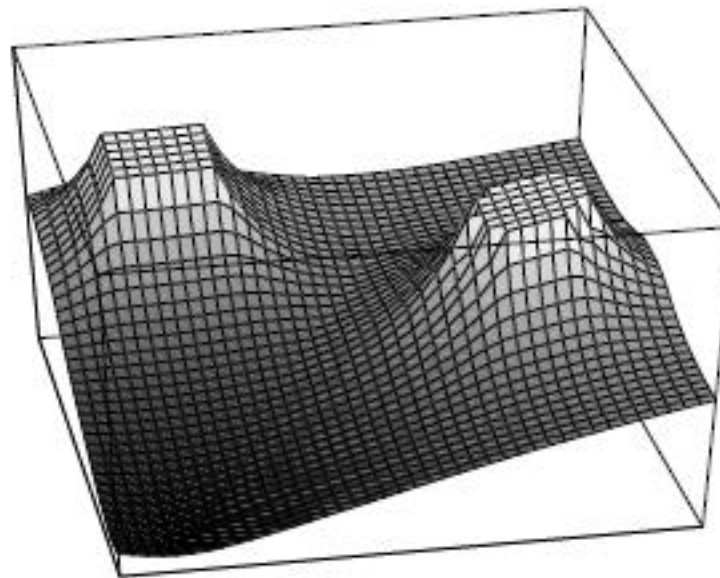


- Der **Hindernisraum** C_{obs} hat das abstoßende Potential U_{ab}



Potentialfelder: Beispiel II

- Die Summe der einwirkenden Kräfte bestimmt die Richtung der Bewegung.
- Für das Potentialfeld gilt: $U(\mathbf{q}) = U_{an}(\mathbf{q}) + U_{ab}(\mathbf{q})$
- Für das Kräftefeld gilt: $F(\mathbf{q}) = F_{an}(\mathbf{q}) + F_{ab}(\mathbf{q})$



Potentialfelder: Lokale Minima

■ 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:

- U_{an} und U_{ab} so definieren, dass U kein lokales Minimum hat, außer in $\mathbf{q}_{ziel}$
- Im Suchalgorithmus Techniken zur “Flucht” aus lokalen Minima anwenden

Baumsuche

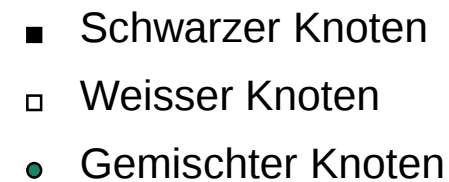
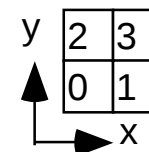
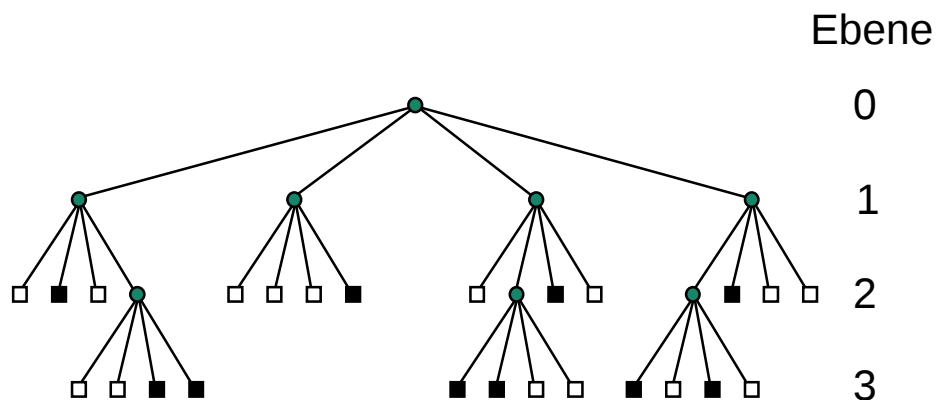
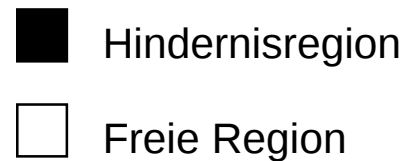
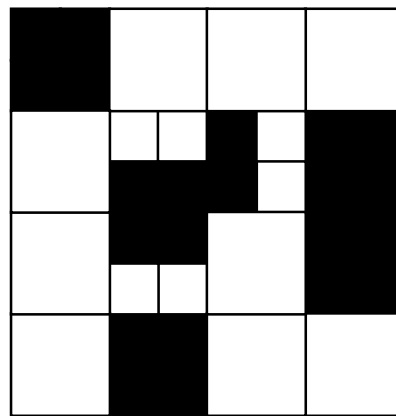
- Anwendungsfall:
 - Mobiler Roboter
 - 2D-Arbeits- und Konfigurationsraum

- Darstellung des Konfigurationsraums als **Quadtrees**
 - Rekursive Unterteilung des Konfigurationsraums in Kacheln
 - Kacheln sind entweder frei oder ein Hindernis

- Bewegungsplanung:
 - Kacheln finden, in denen sich Start- bzw. Zielkonfiguration befinden
 - Benachbarte freie Kacheln des Baums vom Start zum Ziel verbinden
 - Kollisionsfreie Routenplanung durch freie Kacheln

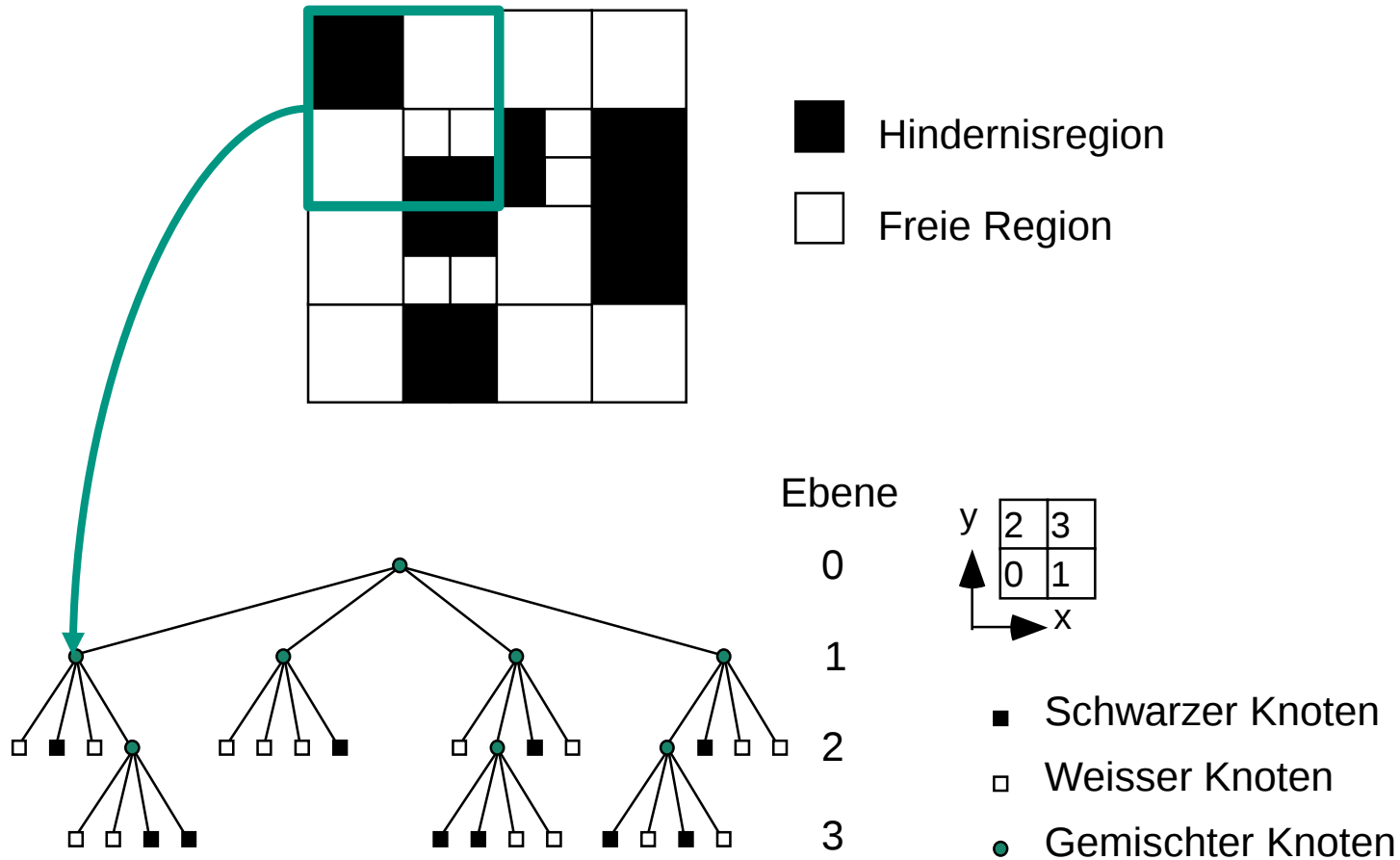
Baumsuche: Quadtree I

Darstellung des Konfigurationsraums als Quadtree



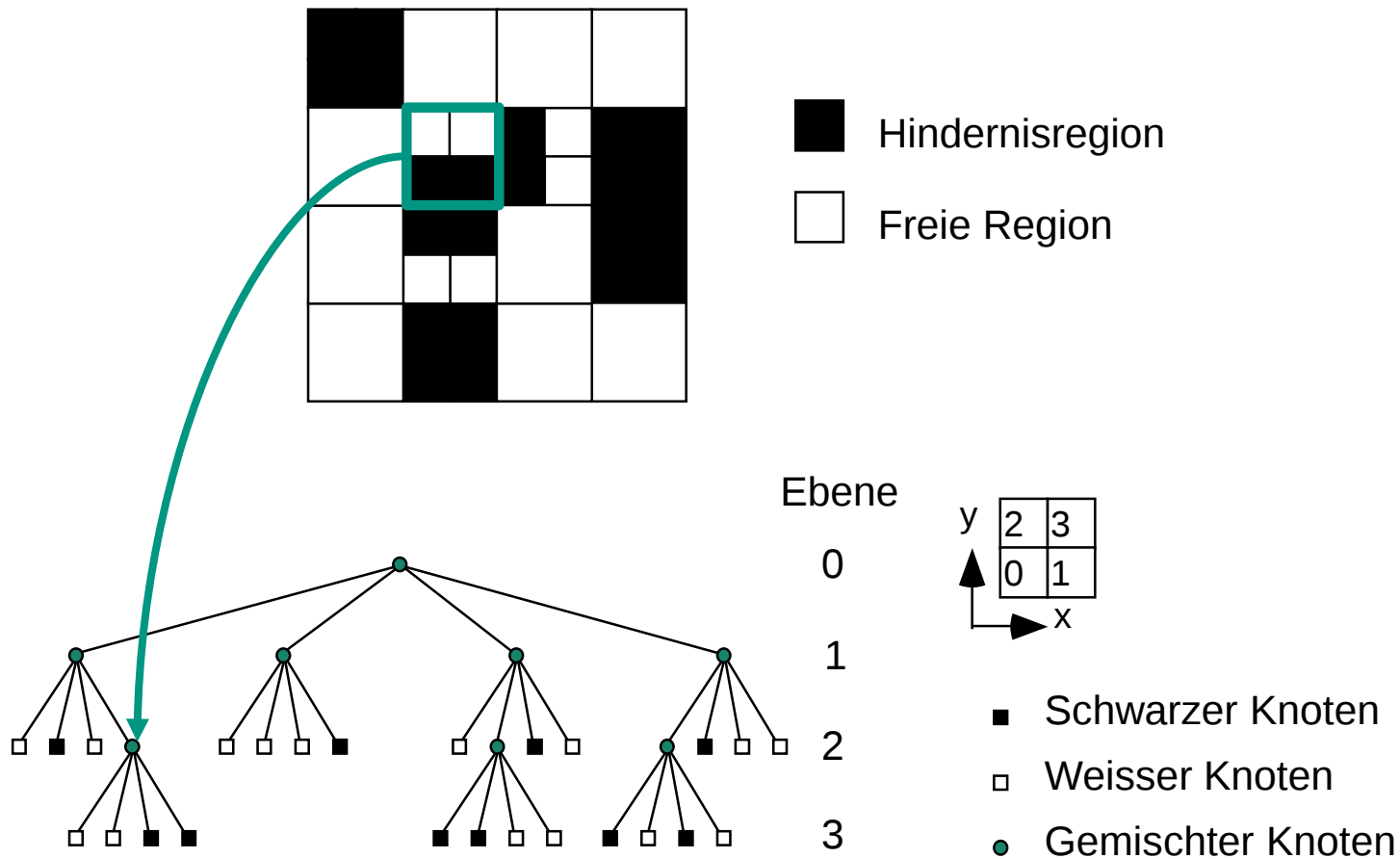
Baumsuche: Quadtree II

Beispiel: Ebene 1, Bereich 2



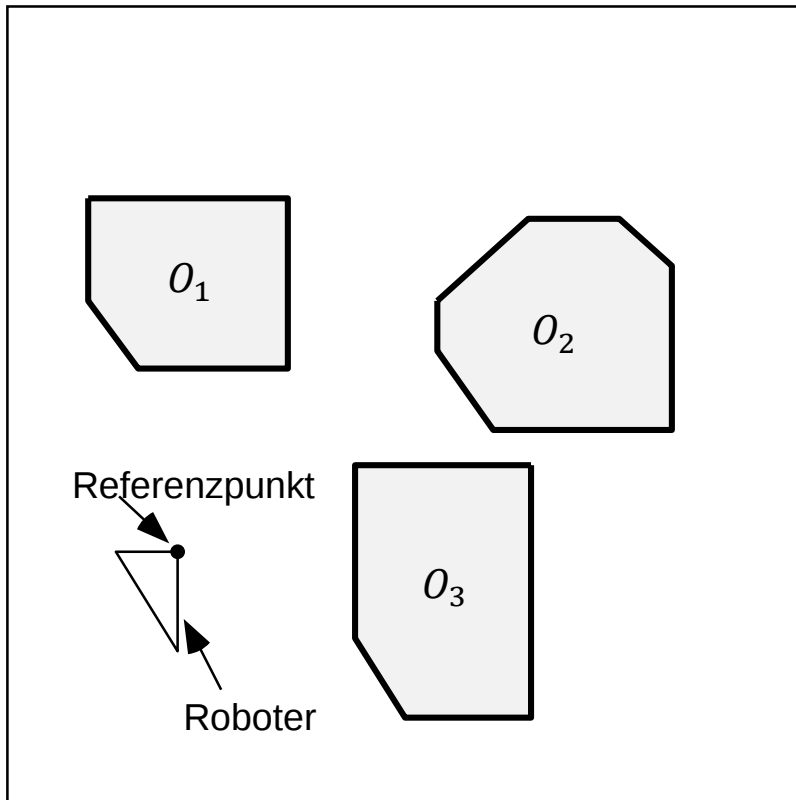
Baumsuche: Quadtree III

Beispiel Ebene 2, Bereich 1



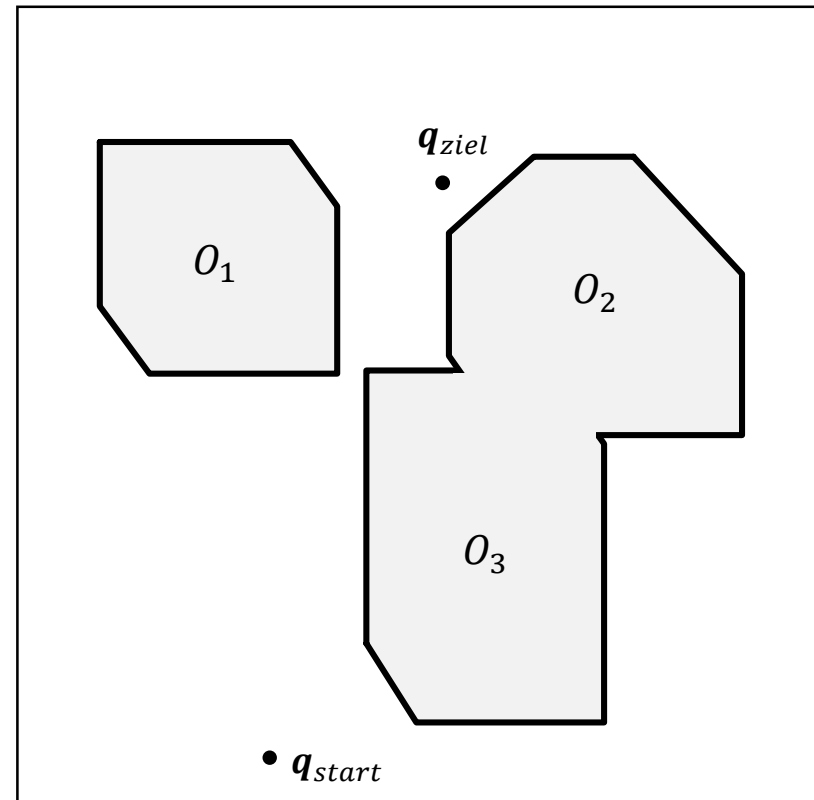
Baumsuche: Beispiel I

Arbeitsraum



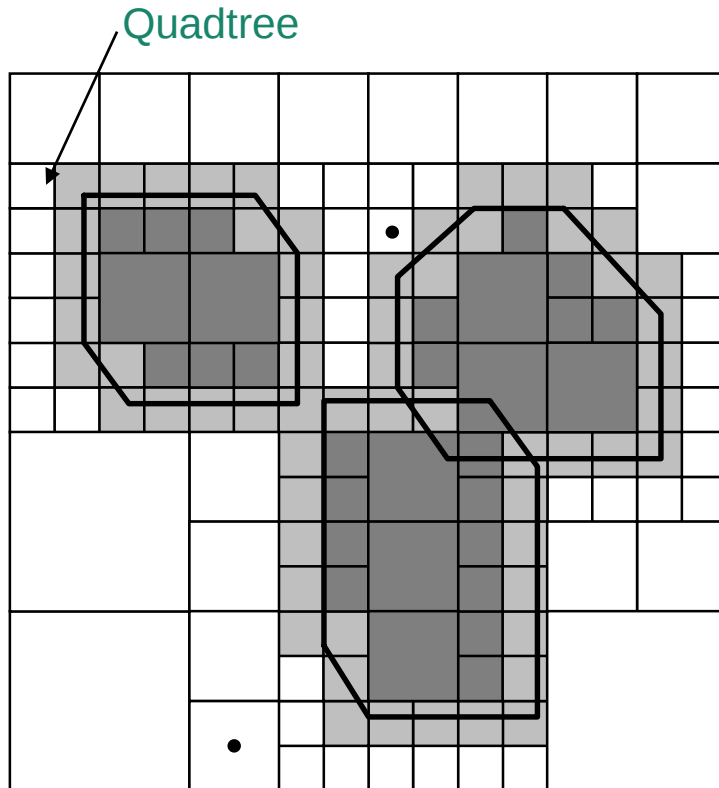
Arbeitsraum eines Roboters mit Hindernissen

Konfigurationsraum

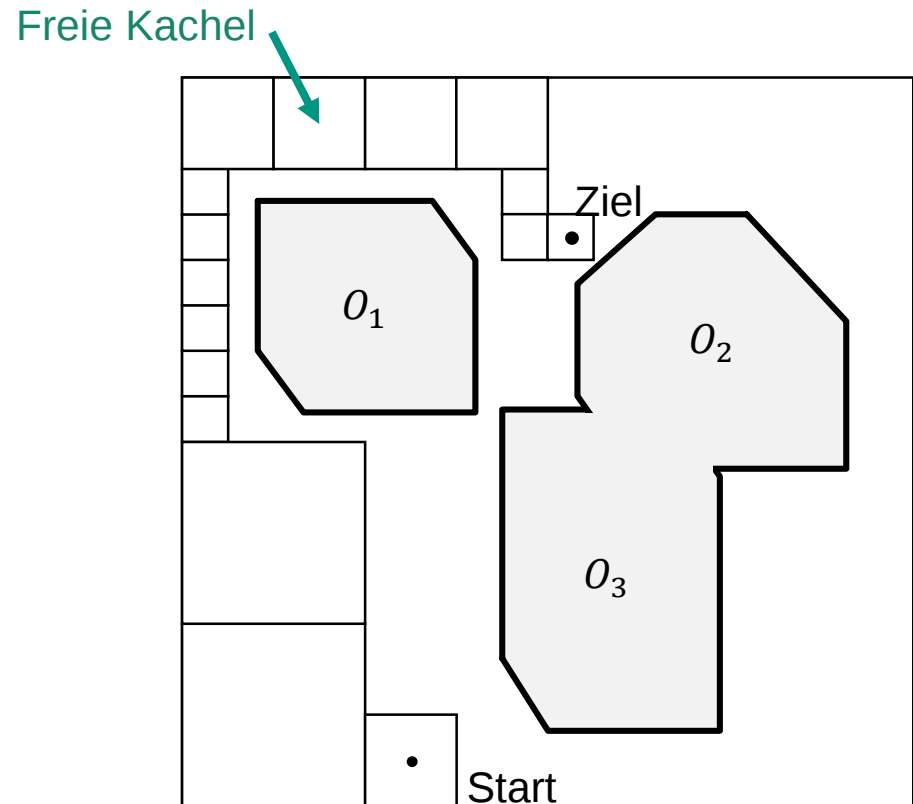


Konfigurationsraum für das vorliegende Robotersystem

Baumsuch: Beispiel II

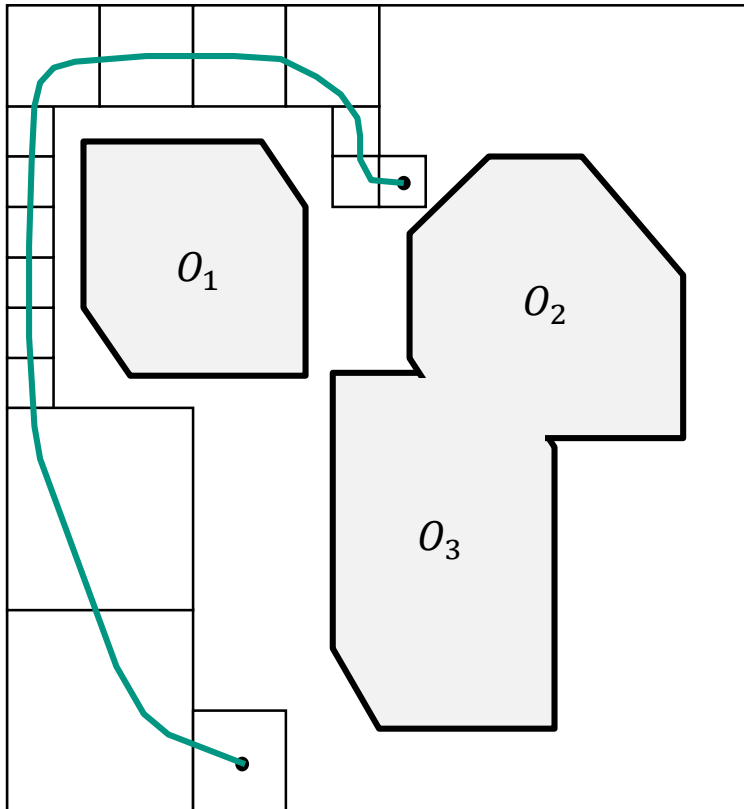


Zerlegung des Konfigurationsraums
in Kachelzonen

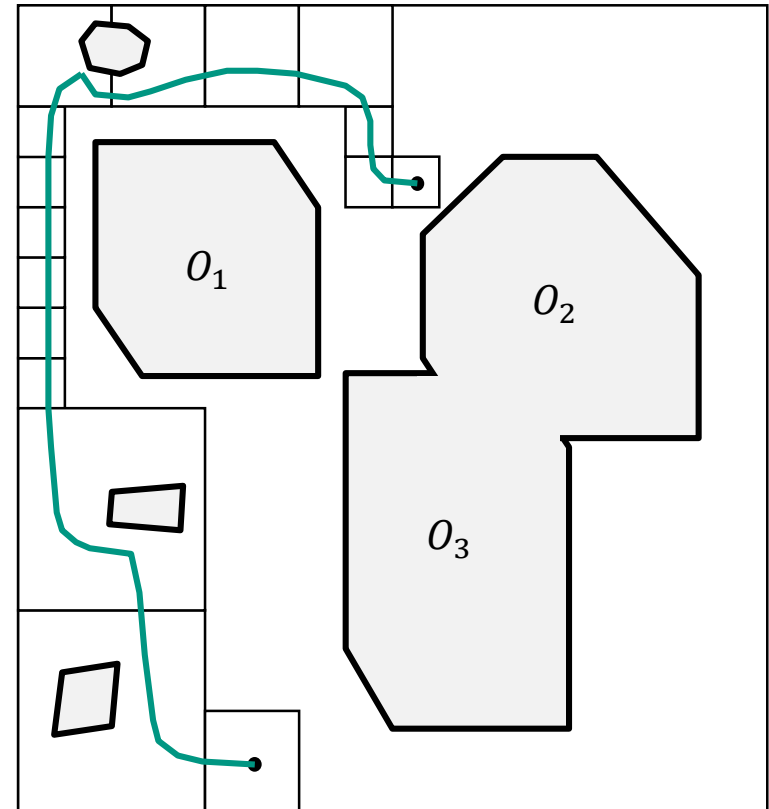


Gesucht: Folge von freien Kacheln
vom Start- zum Zielpunkt

Baumsuche: Beispiel III



Hindernisfreie Verfahrroute



Ausweichmanöver um lokale Hindernisse

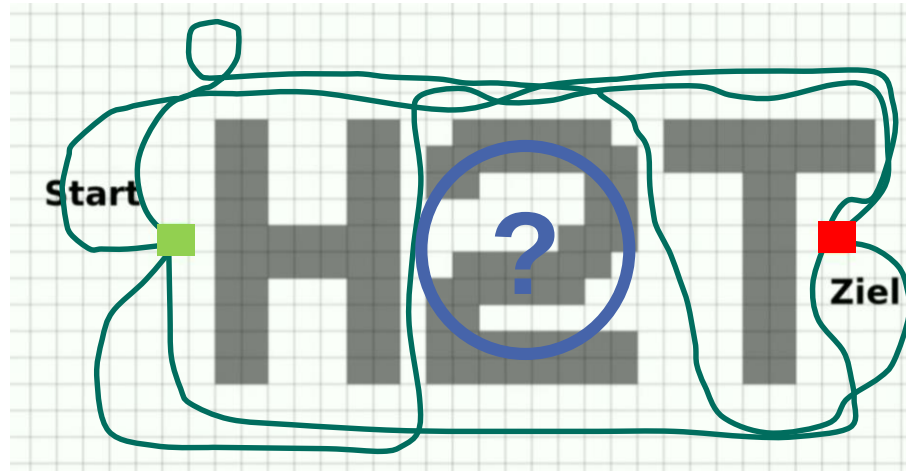
A*-Algorithmus

- Motivation: Kürzester Pfad von Start nach Ziel



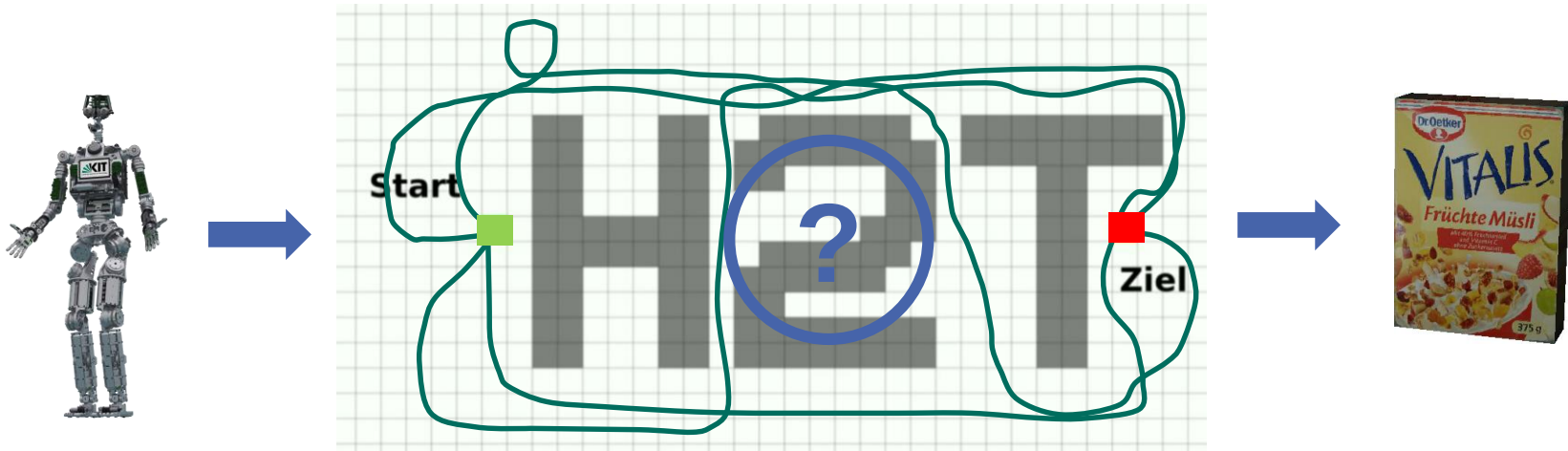
A*-Algorithmus

- Motivation: Kürzester Pfad von Start nach Ziel



A*-Algorithmus

- Motivation: Kürzester Pfad von Start nach Ziel

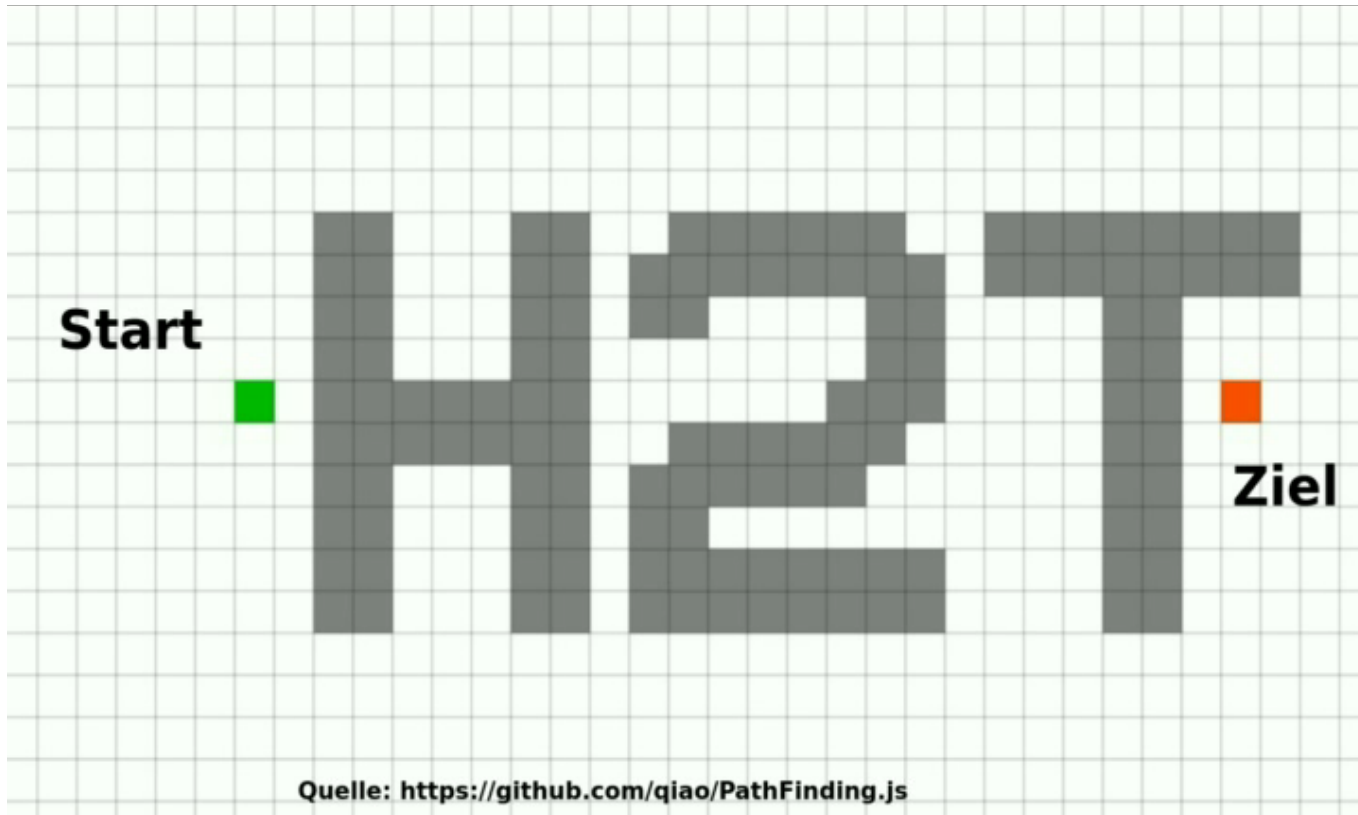


- A* ist einer der beliebtesten Algorithmen zur Routenplanung
- Kostenfunktion ist $f(x) = g(x) + h(x)$
 - $g(x)$ entspricht Kosten von Start-Knoten nach Knoten x
 - $h(x)$ entspricht **geschätzten** Kosten von Knoten x nach Ziel-Knoten

A*-Algorithmus: Funktionsweise

- Unterteile Knoten in
 - **Open Set**: Enthält zu Beginn alle Knoten außer dem Start-Knoten
 - **Current Set**: Enthält zu Beginn nur den Start-Knoten
 - **Closed Set**: Enthält bereits untersuchte Knoten, zu welchen der kürzeste Weg bekannt ist, zu Beginn leer.
- Knoten aus dem **Current Set** mit geringstem $f(x)$ wird als nächstes untersucht
- Wenn ein Knoten x **abschließend untersucht** wurde, dann werden die Nachfolgeknoten in das **Open Set** eingefügt und x in das Closed Set aufgenommen
- Algorithmus **terminiert**, wenn der **Ziel-Knoten abschließend untersucht** worden ist.
- Wenn **Open Set** leer ist, terminiert der Algorithmus ohne Lösung

A*-Algorithmus: Visualisierung



Englische Begriffe

Deutsch	Englisch
Bewegungsplanung	Motion planning
Freiraum	Free space
Hindernis	Obstacle
Konfiguration	Configuration
Konfigurationsraum	Configuration space
Pfadplanung	Path planning
Potentialfeld	Potential field
Sichtgraph	View graph
Trajektorie	Trajectory
Zellzerlegung	Cell decomposition
Zwangsbedingung	Constraint
Neben- und Randbedingung	Constraint