

Funktionsprinzipien und Anwendung von Algorithmen zur Pfadplanung

Eine kurze Einführung mit Beispielen

Bernardo Cordero, Simon Deutscher, Felix Kalchschmid

Hochschule Trier

2. Juli 2020

- 1 Darstellung des Raums
- 2 Graphen

- 1 Definition von Pfadplanungsalgorithmen
- 2 Klassifizierung von Pfadplanungsalgorithmen
- 3 Diskrete Pfadplanung
- 4 Kontinuierliche Pfadplanung

- 1 Rubic's Cube Rätsel
- 2 Videospiele
- 3 Digitales Planen von Fabrikrobotern
- 4 Autonomes Fahren

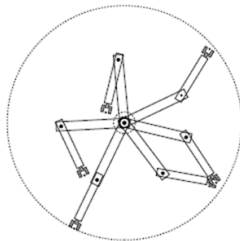
Was ist Pfadplanung?

Eine Einführung

- Pfadplanung ist die Spezifizierung einer Aufgabe in einer Hochsprache, die ein Roboter verstehen und ausführen soll.
- *Roboter, Agent oder Player* ist der Nutzer eines Plans, mit dem er Entscheidungen trifft.
 - Roboter sollen seiner Umgebung verstehen
 - Roboter haben Bewegungseinschränkungen



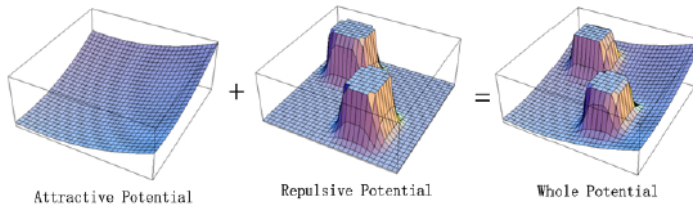
- Der Arbeitsraum ist die Welt, in dem der Roboter sich befindet
- Konfigurationsraum ist der Raum aller seine Konfigurationen
 - Konfiguration ist die Position der Roboter in seiner verständliche Sprache
 - Hindernisse sind ungültige Konfigurationen
 - Freier Raum ist die Gruppe von Konfigurationen ohne Kollisionen



Darstellung des Raums

Potentialfeld

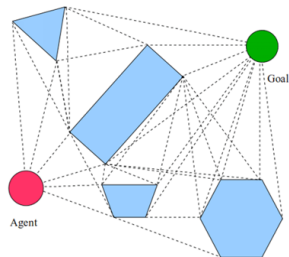
- Punkte in dem Konfigurationsraum bekommen eine Potentialvektor.
 - Sie wehren Hindernisse ab
 - Sie werden vom Ziel angezogen



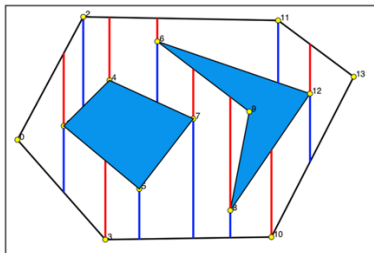
Darstellung des Raums

Roadmap

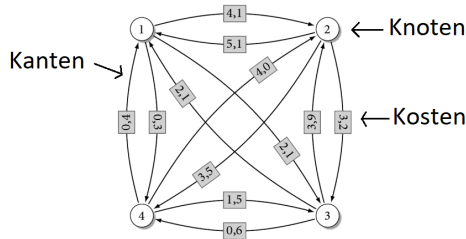
- Die Punkte des Konfigurationsraum verbinden
- Bewegung in Roadmap durch:
 - Roadmap eintreten
 - Roadmap durchlaufen
 - Ziel erreichen
- Sichtbarkeitsgraphen sind Roadmaps



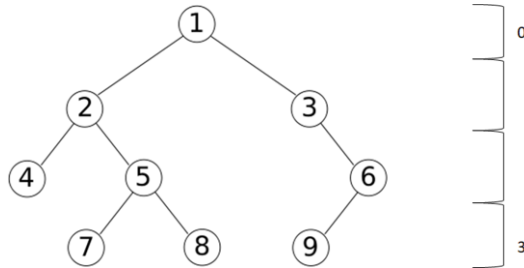
- Teilung des freien Raums in Zellen
 - Verbindungen zwischen anliegenden Zellen speichern
- Zellen enthalten Start- und Endkonfigurationen beim Pfadfindung.



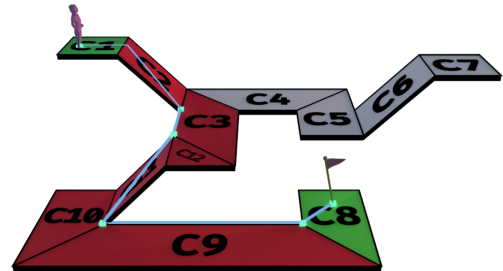
- Graphen sind gut für Pfadfindung
- Ein Graph besteht aus Knoten, Kanten und Kosten
- Gerichteter Graphen unterscheiden ausgehende und eingehende Kanten



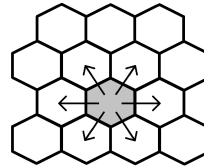
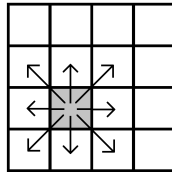
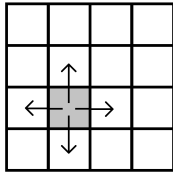
- Hierarchische Darstellung von Graphen
- Ein Baum hat ein Wurzel
- Knoten haben Tiefe: Entfernung von Wurzel



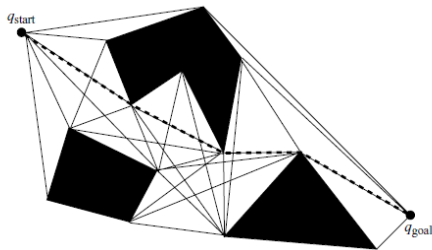
- Navigationsnetze definieren der freie Raum als polygonale Bereiche
- Verbindungen zwischen Bereiche sind in Navigationsgraph gespeichert
- Start und Endkonfigurationen befinden sich in die Bereiche



- Raum in Felder definieren
 - Felder können verschieden Formen haben
- Die Felder kennen ihre Nachbarn



- Roadmap aus sichtbare Punkte bilden
- Die Eckpunkte der Hindernisse verbinden
- Sensoren nehmen neue Information für den Roboter, um neue Kanten zu bilden



- Es gibt keine einfache Definition für Pfadplanungsalgorithmen.
- Die Algorithmen Definition wird aus der *Church-Turing-These* hergeleitet.

Theorem 1 (Turing-Vollständigkeit).

Ein Berechnungsmodell, mit dem alle Funktionen der Klasse WHILE berechnet werden können, heißt Turing-vollständig.

Theorem 2 (Church-Turing-These).

*Jede im intuitiven Sinne berechenbare Funktion ist **WHILE**-berechenbar.*

Algorithmen zur Pfadplanung

Definition von Pfadplanungsalgorithmen

- Diese Definitionen ist Unvollständig
- Es fehlt die Repräsentation der Interaktion des Roboters mit der Umgebung.
- **Lösung:** Ein Planer ist ein Algorithmus der einen Plan erstellt.
 - kann einer Turing-Maschine entsprechen.
 - kann optional erweitert werden.

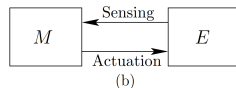
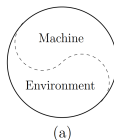


Abbildung 1: (a) Grenze zwischen Maschine und Umgebung ist fließend.
(b) Maschine M interagiert mit der Umgebung U durch Sensorik und Antrieb.

Algorithmen zur Pfadplanung

Definition von Pfadplanungsalgorithmen

- Diese Definitionen ist Unvollständig
- Es fehlt die Repräsentation der Interaktion des Roboters mit der Umgebung.
- **Lösung:** Ein Planer ist ein Algorithmus der einen Plan erstellt.
 - kann einer Turing-Maschine entsprechen.
 - kann optional erweitert werden.

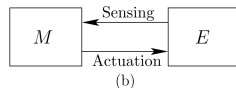
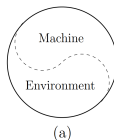


Abbildung 1: (a) Grenze zwischen Maschine und Umgebung ist fließend.
(b) Maschine M interagiert mit der Umgebung U durch Sensorik und Antrieb.

- Diese Definitionen ist Unvollständig
- Es fehlt die Repräsentation der Interaktion des Roboters mit der Umgebung.
- **Lösung:** Ein Planer ist ein Algorithmus der einen Plan erstellt.
 - kann einer Turing-Maschine entsprechen.
 - kann optional erweitert werden.

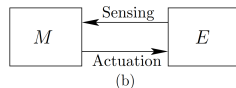
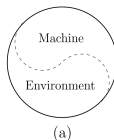


Abbildung 1: (a) Grenze zwischen Maschine und Umgebung ist fließend.
(b) Maschine M interagiert mit der Umgebung U durch Sensorik und Antrieb.

- Diese Definitionen ist Unvollständig
- Es fehlt die Repräsentation der Interaktion des Roboters mit der Umgebung.
- **Lösung:** Ein Planer ist ein Algorithmus der einen Plan erstellt.
 - kann einer Turing-Maschine entsprechen.
 - kann optional erweitert werden.

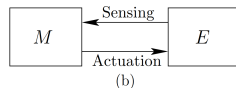
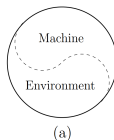


Abbildung 1: (a) Grenze zwischen Maschine und Umgebung ist fließend.
(b) Maschine M interagiert mit der Umgebung U durch Sensorik und Antrieb.

- Diese Definitionen ist Unvollständig
- Es fehlt die Repräsentation der Interaktion des Roboters mit der Umgebung.
- **Lösung:** Ein Planer ist ein Algorithmus der einen Plan erstellt.
 - kann einer Turing-Maschine entsprechen.
 - kann optional erweitert werden.

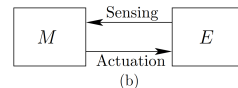
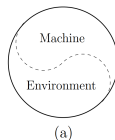


Abbildung 1: (a) Grenze zwischen Maschine und Umgebung ist fließend.
(b) Maschine M interagiert mit der Umgebung U durch Sensorik und Antrieb.

- **Ausführen**

- als kodierte Eingabe den Roboter programmieren
- als Spezialmaschine eine spezielle Aufgabe lösen.

- **Verbesserung**

- Den Plan dem Planer übergeben und mit Fokus auf verschiedene Aspekte neu berechnen.
- Es entsteht ein Verbesserungsprozess der viele Aspekte mit einbezieht.

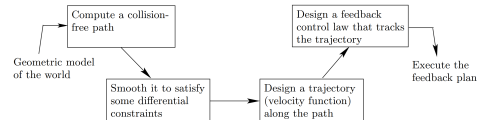


Abbildung 2: Ein Verbesserungsprozess der sich in der Robotik bewährt hat.

- Ausführen
 - als kodierte Eingabe den Roboter programmieren
 - als Spezialmaschine eine spezielle Aufgabe lösen.
- Verbesserung
 - Den Plan dem Planer übergeben und mit Fokus auf verschiedene Aspekte neu berechnen.
 - Es entsteht ein Verbesserungsprozess der viele Aspekte mit einbezieht.

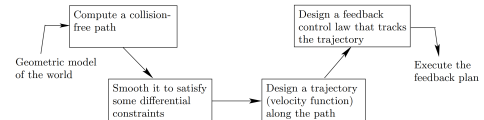


Abbildung 2: Ein Verbesserungsprozess der sich in der Robotik bewährt hat.

- Ausführen
 - als kodierte Eingabe den Roboter programmieren
 - als Spezialmaschine eine spezielle Aufgabe lösen.
- Verbesserung
 - Den Plan dem Planer übergeben und mit Fokus auf verschiedene Aspekte neu berechnen.
 - Es entsteht ein Verbesserungsprozess der viele Aspekte mit einbezieht.

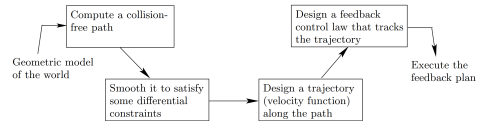


Abbildung 2: Ein Verbesserungsprozess der sich in der Robotik bewährt hat.

- Ausführen
 - als kodierte Eingabe den Roboter programmieren
 - als Spezialmaschine eine spezielle Aufgabe lösen.
- Verbesserung
 - Den Plan dem Planer übergeben und mit Fokus auf verschiedene Aspekte neu berechnen.
 - Es entsteht ein Verbesserungsprozess der viele Aspekte mit einbezieht.

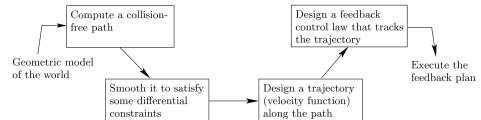


Abbildung 2: Ein Verbesserungsprozess der sich in der Robotik bewährt hat.

- Ausführen
 - als kodierte Eingabe den Roboter programmieren
 - als Spezialmaschine eine spezielle Aufgabe lösen.
- Verbesserung
 - Den Plan dem Planer übergeben und mit Fokus auf verschiedene Aspekte neu berechnen.
 - Es entsteht ein Verbesserungsprozess der viele Aspekte mit einbezieht.

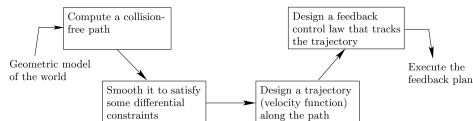


Abbildung 2: Ein Verbesserungsprozess der sich in der Robotik bewährt hat.

- Ausführen
 - als kodierte Eingabe den Roboter programmieren
 - als Spezialmaschine eine spezielle Aufgabe lösen.
- Verbesserung
 - Den Plan dem Planer übergeben und mit Fokus auf verschiedene Aspekte neu berechnen.
 - Es entsteht ein Verbesserungsprozess der viele Aspekte mit einbezieht.

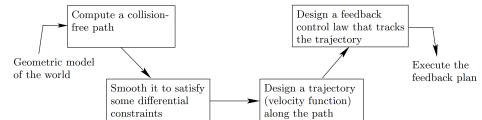


Abbildung 2: Ein Verbesserungsprozess der sich in der Robotik bewährt hat.

- Hierarchische Inklusion
 - Plan wird als Aktion in einen Größeren Plan aufgenommen
 - Ein problem wird in viele kleine Probleme zerteilt
 - Es entsteht eine Baumstruktur
- Wie kann das Objekt von seiner Ursprungsposition (oben, links) zur finalen Position (unten, links) bewegt werden?
- Die Lösung (rechts) erfordert mehrere *Verschiebeoperationen*.

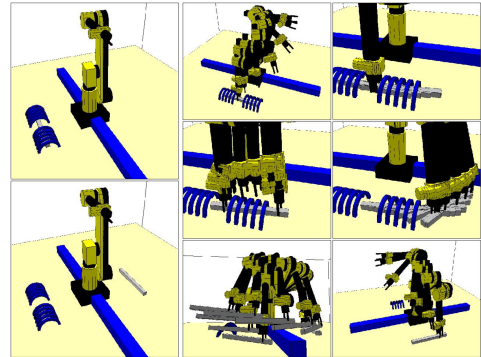


Abbildung 3:

- Hierarchische Inklusion
 - Plan wird als Aktion in einen Größeren Plan aufgenommen
 - Ein problem wird in viele kleine Probleme zerteilt
 - Es entsteht eine Baumstruktur
- Wie kann das Objekt von seiner Ursprungsposition (oben, links) zur finalen Position (unten, links) bewegt werden?
- Die Lösung (rechts) erfordert mehrere *Verschiebeoperationen*.

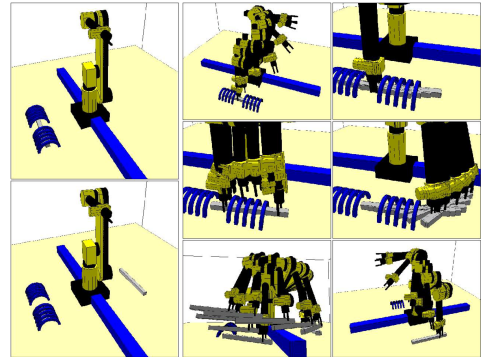


Abbildung 3:

- Hierarchische Inklusion
 - Plan wird als Aktion in einen Größeren Plan aufgenommen
 - Ein problem wird in viele kleine Probleme zerteilt
 - Es entsteht eine Baumstruktur
- Wie kann das Objekt von seiner Ursprungsposition (oben, links) zur finalen Position (unten, links) bewegt werden?
- Die Lösung (rechts) erfordert mehrere *Verschiebeoperationen*.

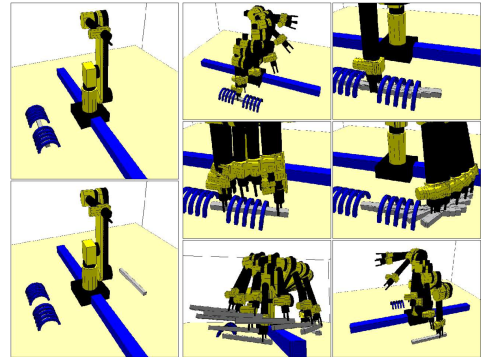


Abbildung 3:

- Hierarchische Inklusion
 - Plan wird als Aktion in einen Größeren Plan aufgenommen
 - Ein problem wird in viele kleine Probleme zerteilt
 - Es entsteht eine Baumstruktur
- Wie kann das Objekt von seiner Ursprungsposition (oben, links) zur finalen Position (unten, links) bewegt werden?
- Die Lösung (rechts) erfordert mehrere *Verschiebeoperationen*.

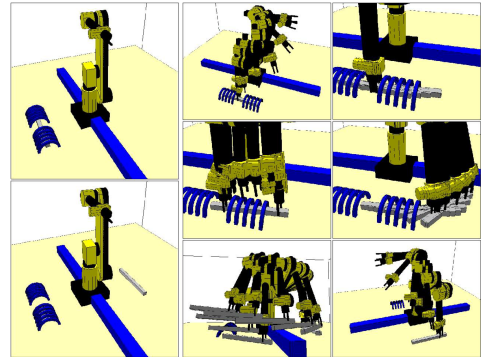


Abbildung 3:

- Hierarchische Inklusion
 - Plan wird als Aktion in einen Größeren Plan aufgenommen
 - Ein problem wird in viele kleine Probleme zerteilt
 - Es entsteht eine Baumstruktur
- Wie kann das Objekt von seiner Ursprungsposition (oben, links) zur finalen Position (unten, links) bewegt werden?
- Die Lösung (rechts) erfordert mehrere *Verschiebeoperationen*.

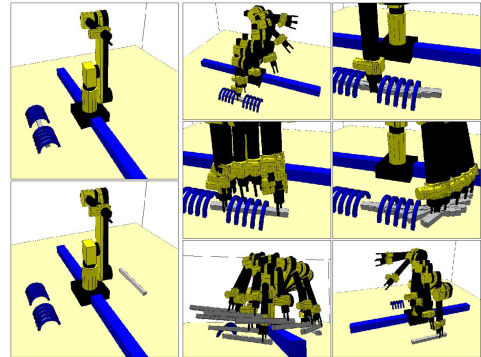


Abbildung 3:

- Hierarchische Inklusion
 - Plan wird als Aktion in einen Größeren Plan aufgenommen
 - Ein problem wird in viele kleine Probleme zerteilt
 - Es entsteht eine Baumstruktur
- Wie kann das Objekt von seiner Ursprungsposition (oben, links) zur finalen Position (unten, links) bewegt werden?
- Die Lösung (rechts) erfordert mehrere *Verschiebeoperationen*.

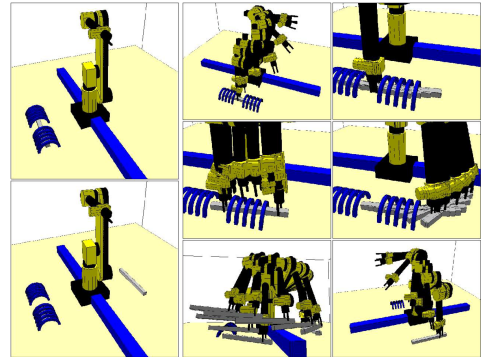


Abbildung 3:

- Abhängigkeit vom Einsatz dadurch stark Heterogen.
- Planen im diskreten und kontinuierlichen Zustandsraum.
 - Alle Umgebungsinformationen (Industrieroboter, kontrollierte Umgebung)
 - Unsicherheit (Roboter mit Kameras)
 - Bewegungseinschränkung (Roboter mit Rädern, Auto)
- Vorherrschaft von diskreten Algorithmen auch für kontinuierliches Pfadplanen.

- Abhängigkeit vom Einsatz dadurch stark Heterogen.
- Planen im **diskreten** und kontinuierlichen Zustandsraum.
 - Alle Umgebungsinformationen (Industrieroboter, kontrollierte Umgebung)
 - Unsicherheit (Roboter mit Kameras)
 - Bewegungseinschränkung (Roboter mit Rädern, Auto)
- Vorherrschaft von diskreten Algorithmen auch für kontinuierliches Pfadplanen.

- Abhängigkeit vom Einsatz dadurch stark Heterogen.
- Planen im diskreten und **kontinuierlichen** Zustandsraum.
 - Alle Umgebungsinformationen (Industrieroboter, kontrollierte Umgebung)
 - Unsicherheit (Roboter mit Kameras)
 - Bewegungseinschränkung (Roboter mit Rädern, Auto)
- Vorherrschaft von diskreten Algorithmen auch für kontinuierliches Pfadplanen.

- Abhängigkeit vom Einsatz dadurch stark Heterogen.
- Planen im diskreten und **kontinuierlichen** Zustandsraum.
 - Alle Umgebungsinformationen (Industrieroboter, kontrollierte Umgebung)
 - Unsicherheit (Roboter mit Kameras)
 - Bewegungseinschränkung (Roboter mit Rädern, Auto)
- Vorherrschaft von diskreten Algorithmen auch für kontinuierliches Pfadplanen.

- Abhängigkeit vom Einsatz dadurch stark Heterogen.
- Planen im diskreten und **kontinuierlichen** Zustandsraum.
 - Alle Umgebungsinformationen (Industrieroboter, kontrollierte Umgebung)
 - Unsicherheit (Roboter mit Kameras)
 - Bewegungseinschränkung (Roboter mit Rädern, Auto)
- Vorherrschaft von diskreten Algorithmen auch für kontinuierliches Pfadplanen.

- Abhängigkeit vom Einsatz dadurch stark Heterogen.
- Planen im diskreten und **kontinuierlichen** Zustandsraum.
 - Alle Umgebungsinformationen (Industrieroboter, kontrollierte Umgebung)
 - Unsicherheit (Roboter mit Kameras)
 - Bewegungseinschränkung (Roboter mit Rädern, Auto)
- Vorherrschaft von diskreten Algorithmen auch für kontinuierliches Pfadplanen.

- Zustandsraum ist abzählbar unendlich
 - Es müssen keine geometrischen Modelle beachtet werden.
 - Es gibt grundsätzlich keine Bewegungseinschränkungen.
- Systematisch: wenn ein pfad existiert wird er gefunden
- Feasible Planning: ein Ziel wird gefunden
- Optimales Planen: Optimierung auf Zeit, Distanz, Anzahl Drehungen, usw.

- Zustandsraum ist abzählbar unendlich
 - Es müssen keine geometrischen Modelle beachtet werden.
 - Es gibt grundsätzlich keine Bewegungseinschränkungen.
- Systematisch: wenn ein pfad existiert wird er gefunden
- Feasible Planning: ein Ziel wird gefunden
- Optimales Planen: Optimierung auf Zeit, Distanz, Anzahl Drehungen, usw.

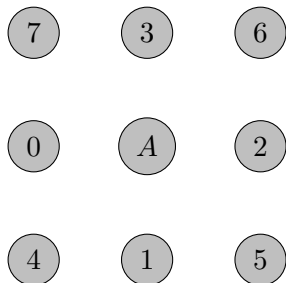
- Zustandsraum ist abzählbar unendlich
 - Es müssen keine geometrischen Modelle beachtet werden.
 - Es gibt grundsätzlich keine Bewegungseinschränkungen.
- Systematisch: wenn ein pfad existiert wird er gefunden
- Feasible Planning: ein Ziel wird gefunden
- Optimales Planen: Optimierung auf Zeit, Distanz, Anzahl Drehungen, usw.

- Zustandsraum ist abzählbar unendlich
 - Es müssen keine geometrischen Modelle beachtet werden.
 - Es gibt grundsätzlich keine Bewegungseinschränkungen.
- Systematisch: wenn ein pfad existiert wird er gefunden
- Feasible Planning: ein Ziel wird gefunden
- Optimales Planen: Optimierung auf Zeit, Distanz, Anzahl Drehungen, usw.

- Algorithmen des Typ Forward Search werden hier verwendet.
- Systematisches Vorgehen
 - Endlich viele Zustände müssen alle besucht werden.
 - Bei unendlich vielen Zuständen reicht es aus zu einer Lösung zu kommen.
- Entdecktes Feld, Unentdecktes Feld und Totes Feld
- Einfluss der Sortierung auf das Verhalten
 - FIFO
 - LIFO
 - Dijkstra
 - A*

```
FORWARD_SEARCH
1   $Q.Insert(x_I)$  and mark  $x_I$  as visited
2  while  $Q$  not empty do
3       $x \leftarrow Q.GetFirst()$ 
4      if  $x \in X_G$ 
5          return SUCCESS
6      forall  $u \in U(x)$ 
7           $x' \leftarrow f(x, u)$ 
8          if  $x'$  not visited
9              Mark  $x'$  as visited
10              $Q.Insert(x')$ 
11          else
12              Resolve duplicate  $x'$ 
13  return FAILURE
```

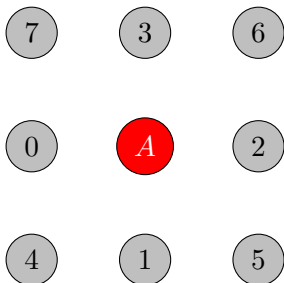
Abbildung 4:



- Sei ein Knoten A als Startknoten gegeben
- Sei eine Menge $U(x)$ gegeben, die alle unentdeckten Knoten enthält.

$$U(x) = \{A, 0, 1, 2, 3, 4, 5, 6, 7\}$$

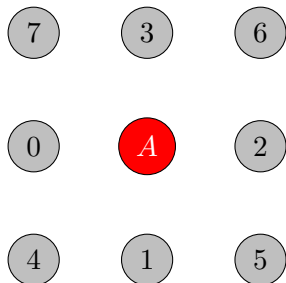
- Knoten 4 wird als Zielknoten definiert.



- Sei ein Knoten A als Startknoten gegeben
- Sei eine Menge $U(x)$ gegeben, die alle unentdeckten Knoten enthält.

$$U(x) = \{A, 0, 1, 2, 3, 4, 5, 6, 7\}$$

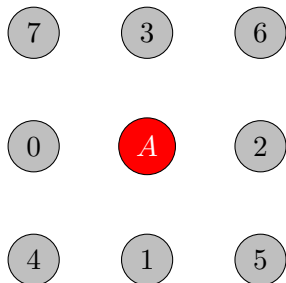
- Knoten 4 wird als Zielknoten definiert.
- Dieser Hinweis wird nur für Schritt 2 angezeigt.



- Sei ein Knoten A als Startknoten gegeben
- Sei eine Menge $U(x)$ gegeben, die alle unentdeckten Knoten enthält.

$$U(x) = \{A, 0, 1, 2, 3, 4, 5, 6, 7\}$$

- Knoten 4 wird als Zielknoten definiert.



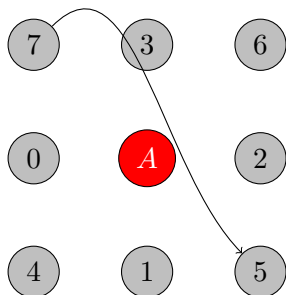
- Sei ein Knoten A als Startknoten gegeben
- Sei eine Menge $U(x)$ gegeben, die alle unentdeckten Knoten enthält.

$$U(x) = \{A, 0, 1, 2, 3, 4, 5, 6, 7\}$$

- Knoten 4 wird als Zielknoten definiert.

Algorithmen zur Pfadplanung

Feasible Planning



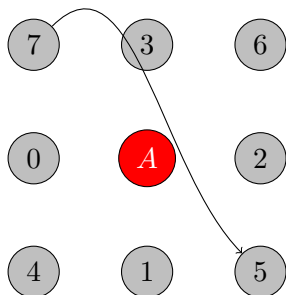
- Sei ein Knoten A als Startknoten gegeben
- Sei eine Menge $U(x)$ gegeben, die alle unentdeckten Knoten enthält.

$$U(x) = \{A, 0, 1, 2, 3, 4, 5, 6, 7\}$$

- Knoten 4 wird als Zielknoten definiert.

Algorithmen zur Pfadplanung

Feasible Planning



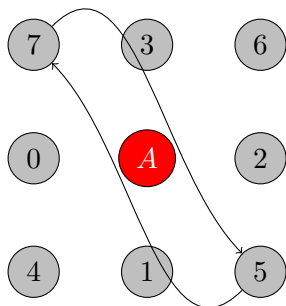
- Sei ein Knoten A als Startknoten gegeben
- Sei eine Menge $U(x)$ gegeben, die alle unentdeckten Knoten enthält.

$$U(x) = \{A, 0, 1, 2, 3, 4, 5, 6, 7\}$$

- Knoten 4 wird als Zielknoten definiert.

Algorithmen zur Pfadplanung

Feasible Planning



- Sei ein Knoten A als Startknoten gegeben
- Sei eine Menge $U(x)$ gegeben, die alle unentdeckten Knoten enthält.

$$U(x) = \{A, 0, 1, 2, 3, 4, 5, 6, 7\}$$

- Knoten 4 wird als Zielknoten definiert.

Unterschiede zur diskreten Pfadplanung

- Der Zustandsraum ist überabzählbar unendlich
- Der Zustandsraum ist zu groß, um explizit dargestellt zu werden
 - Die Anzahl Zustände ist zu groß.
 - Die kombinatorische Komplexität ist zu groß.

Das Piano Mover's Problem

- Ein Körper muss einen kontinuierlichen Bewegungspfad finden.
- Geometrische Einschränkungen
- kein Kontakt mit Hindernissen im Raum

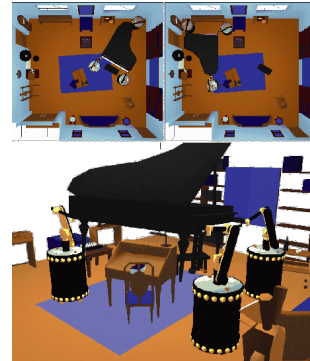


Abbildung 5: mobile Roboter bewegen einen Flügel [?, Abb. 1.5, S.9]

Zwei wichtige Motive in der Pfadplanung:

Implizite Representation des Zustandsraumes

- Explizite Darstellung des Zustandsraumes nicht moglich
- Daher Implizite Darstellung
- kein Kontakt mit Hindernissen im Raum

Transformation von kontinuierlichen Modellen in diskrete

- combinatorial motion planning
- sampling based motion planning

Anwendungen der Pfadplanung

Das Rubic's Cube Rätsel

- Zustandsraum: Alle möglichen Farbverteilungen
- Aktionsrahmen: Die Menge aller möglichen Drehungen (12)
- bestmöglicher Pfad: die geringste Anzahl an Drehungen, bis der Würfel einfarbige Seiten hat
- diskret

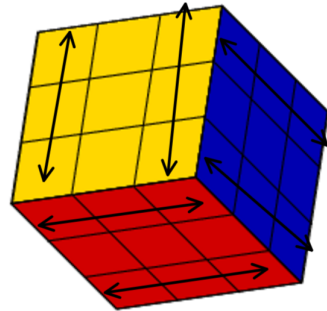


Abbildung 6: mobile Roboter bewegen einen Flügel

Anwendungen der Pfadplanung

Videospiele

- Zustandsraum: Alle möglichen Positionen der zu bewegendenden Einheit und alle möglichen Map-Gegebenheiten
- Aktionsrahmen: alle möglichen Bewegungsrichtungen
- bestmöglicher Pfad: der schnellste Weg um Ziel
- diskret



Abbildung 7: Screenshot of Civilisation V

Anwendungen der Pfadplanung

Digitales Planen von Fabrikrobotern

- Zustandsraum: Alle möglichen Position der Fabrikroboter, des Autos und des zu montierenden Teils
- Aktionsrahmen: Die Menge aller möglichen Bewegungen des Roboters
- kontinuierlich
- Alle Umgebungsinformationen sind bekannt

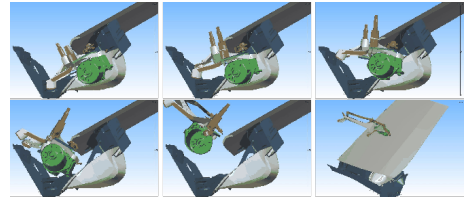


Abbildung 8:

- Zustandsraum: Jede mögliche Position auf der Welt und jede mögliche Umgebung
- Aktionsrahmen:
 - Beschleunigen
 - Bremsen
 - Lenken
 - blinken
 - Gang wechseln
- kontinuierlich
- Planung mit Ungewissheit
- Planung mit Bewegungseinschränkungen

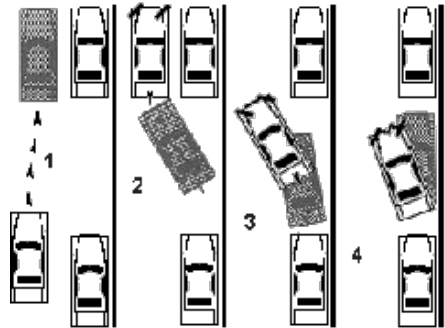


Abbildung 9:

Was ist Beamer?

Eine Übersicht

Das Beamer Paket für \LaTeX ermöglicht es, Präsentationsfolien zu erstellen und unterstützt dabei Features wie Animationen, die manuell nur mit viel Aufwand umgesetzt werden können. Für die Details zu den Features siehe den Beamer User Guide[TWM17].

Folien in Beamer werden durch frame-Umgebungen definiert.

Festlegen des Titels

... und des Subtitels

Die Frames verfügen über einen Titel und einen Subtitel. Diese können entweder beim Öffnen der frame-Umgebung angegeben werden.

Festlegen des Titels

... und des Subtitels

Oder sie werden durch entsprechende Befehle angegeben.

- Fließtext ist meistens nicht sinnvoll für Präsentationen
 - Aufzählungen sind oft besser geeignet
 - Geben Struktur
 - Sorgen für Übersicht
- 1 Auf nummerierte Aufzählungen können verwendet werden
 - 2 Wie hier zu sehen ist
 - 1 Auch mit
 - 2 Unterpunkten

Wichtige Teile im Text können durch alert hervorgehoben werden.

- Schritt 1
- Schritt 2
- Schritt 3

Wichtige Teile im Text können **durch alert hervorgehoben** werden.

- Schritt 1
- **Schritt 2**
- Schritt 3

Wichtige Teile im Text können durch alert hervorgehoben werden.

- Schritt 1
- Schritt 2
- Schritt 3

Theorem 3.

Umgebungen wie theorem können auch in Beamer genutzt werden.

Beweis.

Proof wird auch unterstützt.



Abbildung 10: Testbild in einem figure float



Testbild in center-Umgebung

Der Mathe-Modus kann wie in LaTeX üblich benutzt werden:

$$\mathcal{F} : x = y + \frac{z}{3}, y \in \mathbb{N}, z \in \mathfrak{B}$$

Dies ist ein Block

Blocks können zur Strukturierung des Frame-Inhalts genutzt werden.

Dies ist ein Beispielblock

Inhalt...

Dies ist ein Alert-Block

Inhalt...

Die Option `allowframebreaks` erlaubt es frames mit zu viel Inhalt umzuberechnen. Dies ist besonders bei generiertem Inhalt wie dem Quellenverzeichnis sinnvoll.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla,

malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
1 //Quellcode-Listings
2 //und andere Verbatim-Umgebungen
3 //setzen die fragile-Option des Frames voraus.
4 #include <iostream>
5 void main() {
6     std::cout<<"Hallo Welt"<<std::endl;
7 }
```

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Ganzen Frame automatisch animieren

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Ganzen Frame automatisch animieren

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Ganzen Frame automatisch animieren

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Ganzen Frame automatisch animieren

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Einzelne Umgebung automatisch animieren

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Einzelne Umgebung automatisch animieren

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Einzelne Umgebung automatisch animieren

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Einzelne Umgebung automatisch animieren

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Abweichende Animationsreihenfolge

selbstdefiniert

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Abweichende Animationsreihenfolge

selbstdefiniert

- Schritt 1
- **Schritt 2**
- Schritt 3
- Schritt 4

Abweichende Animationsreihenfolge

selbstdefiniert

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Abweichende Animationsreihenfolge

selbstdefiniert

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Abweichende Animationsreihenfolge

selbstdefiniert

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3

Block-übergreifende Animation

Nebeneinander

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Block-übergreifende Animation

Nebeneinander

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Block-übergreifende Animation

Nebeneinander

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Block-übergreifende Animation

Nebeneinander

Bereich 1

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Bereich 2

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Beliebige Inhalte ein- und ausblenden

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Nachfolgender Text kann sich aber verschieben.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Dieser Hinweis wird nur für Schritt 2 angezeigt. Nachfolgender Text kann sich aber verschieben.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Dieser Hinweis wird nur für Schritt 3 angezeigt. Nachfolgender Text kann sich aber verschieben.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Nachfolgender Text kann sich aber verschieben.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Mit overlayarea verschiebt sich nachfolgender Text nicht.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Dieser Hinweis wird nur für Schritt 2 angezeigt.

Mit overlayarea verschiebt sich nachfolgender Text nicht.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Dieser Hinweis wird nur für Schritt 3 angezeigt.

Mit overlayarea verschiebt sich nachfolgender Text nicht.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Mit overlayarea verschiebt sich nachfolgender Text nicht.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Inhalte können auch durch Animationen aufgedeckt werden.

Dieser Hinweis wird nur für Schritt 2 angezeigt.

Dieser Hinweis wird nur für Schritt 3 angezeigt.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Inhalte können auch durch Animationen aufgedeckt werden.

Dieser Hinweis wird nur für Schritt 2 angezeigt.

Dieser Hinweis wird nur für Schritt 3 angezeigt.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Inhalte können auch durch Animationen aufgedeckt werden.

Dieser Hinweis wird nur für Schritt 2 angezeigt.

Dieser Hinweis wird nur für Schritt 3 angezeigt.

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Inhalte können auch durch Animationen aufgedeckt werden.

Dieser Hinweis wird nur für Schritt 2 angezeigt.

Dieser Hinweis wird nur für Schritt 3 angezeigt.

Animationen in Verbindung mit Zeichnungen

Integration von Beamer mit tikz



- Sei ein Knoten A gegeben
- Wir fügen einen weiteren Knoten B hinzu
- Dann ziehen wir eine Kante von A nach B
- ... und eine von B nach A

Animationen in Verbindung mit Zeichnungen

Integration von Beamer mit tikz



- Sei ein Knoten A gegeben
- Wir fügen einen weiteren Knoten B hinzu
- Dann ziehen wir eine Kante von A nach B
- ... und eine von B nach A

Animationen in Verbindung mit Zeichnungen

Integration von Beamer mit tikz



- Sei ein Knoten A gegeben
- Wir fügen einen weiteren Knoten B hinzu
- Dann ziehen wir eine Kante von A nach B
- ... und eine von B nach A

Animationen in Verbindung mit Zeichnungen

Integration von Beamer mit tikz



- Sei ein Knoten A gegeben
- Wir fügen einen weiteren Knoten B hinzu
- Dann ziehen wir eine Kante von A nach B
- ... und eine von B nach A

Animationen in Verbindung mit Zeichnungen

Integration von Beamer mit tikz



- Sei ein Knoten A gegeben
- Wir fügen einen weiteren Knoten B hinzu
- Dann ziehen wir eine Kante von A nach B
- ... und eine von B nach A

Animationen in Verbindung mit Zeichnungen

Integration von Beamer mit tikz



- Sei ein Knoten A gegeben
- Wir fügen einen weiteren Knoten B hinzu
- Dann ziehen wir eine Kante von A nach B
- ... und eine von B nach A

Animationen in Verbindung mit Zeichnungen

Integration von Beamer mit tikz



- Sei ein Knoten A gegeben
- Wir fügen einen weiteren Knoten B hinzu
- Dann ziehen wir eine Kante von A nach B
- ... und eine von B nach A

Vielen Dank für die Aufmerksamkeit

Fragen?

[TWM17] TANTAU, TILL, JOSEPH WRIGHT und VEDRAN MILETIĆ: *Beamer Documentation*, 2017.

<https://www.ctan.org/pkg/beamer>.