Labor Robotik - Planung

Christoph Zinnen
(Laborkonzept & Manuskript: Prof. Dr. Peter Gemmar)

Fachbereich Informatik
Hochschule Trier

WS17/18

Planung

AMRs müssen ihre Aktionen in einer Umwelt durchführen, dazu benötigen sie Lösungen für wesentliche (Kern-) Aufgaben :

- eine Darstellung ihrer Umwelt (und sich selbst); also eine Darstellung des Raums, von Objekten (statische, dynamische) darin usw.
- eine Planung (siehe "Paradigmen") ihres Verhaltens in diesem Raum;
- ein Konzept und eine Vorgehensweise zur Planung innerhalb dieser Darstellung (z.T. komplexe bis weit reichende Planung).

Aspekte zu Planung

- Planung erfolgt unter verschiedenen Randbedingungen und stellt ein breites Problemfeld dar.
- Berücksichtigung der:
  - Umgebung (statische und dynamische Objekte)
  - Möglichkeiten des AMR (Lokomotion und Kinematik)
- Hauptaufgaben:
  1. Pfad zwischen Start- und Zielpose bestimmen.
  2. dabei Vermeidung von Hindernissen und Berücksichtigung der Kinematik des AMR.

Zielsetzungen

- Randbedingung: Pfad kürzester Länge
- zu beachten: Pfad kürzester Zeit $\neq$ Pfad kürzester Länge,
- Einfluss der Randbedingungen der Umgebung und AMR-Kinematik.
- Jederzeit (*anytime*) Algorithmus: bei Unterbrechung gibt es dennoch eine Lösung – je länger die Laufzeit umso besser das Ergebnis.
- Auch andere Randbedingungen möglich: ”sicherer Pfad” usw.

Pfad-Planung - Algorithmus

Allgemeiner Formalismus und Vereinfachung der Umgebung.

- Starrer AMR, meist als Punkt-AMR,
- Statische und bekannte Umgebung (Welt W) (Bereich oder Arbeitsraum des AMR).
- Eine Menge von Hindernissen,
- AMR bewegt sich optimal auf geraden Linienabschnitten.

Generelles Problem bei Pfadplanung

Finde einen Pfad $p(P_S, P_Z)$, der vom Startpunkt P_S zum Zielpunkt P_Z führt. Dabei ist zu beachten:

- Umgebung und Roboter (holonomische und nicht-holonomische Einschränkungen,
- Korrekte Lösung (keine Hindernisse bzw. kollisionsfrei),
- Vollständigkeit, der Algorithmus findet mindestens eine existierende Lösung (Auflösungs-Vollständigkeit: vollständiger Algorihtmus entsprechend den Auflösungsgrenzen);
- Optimalität: Kosten des aktuellen Pfads verglichen mit denen des optimalen od. idealen Pfads,
- Raum- oder Zeit-Komplexität, Zeit- und Speicheraufwand zum Finden einer Lösung.

Methoden zur Pfadplanung

- Graphensuche,
- dynamische Programmierung (iterative, rekursive Berechnung, Bellmanns Optimalitätsprinzip: $A \rightarrow C$ über B entspricht dem (zu suchenden) optimalen Pfad $A \rightarrow B$ und $B \rightarrow C$, Kostentabelle).

Methoden zur Pfadermittlung

- **Diskreter Suchraum**
- **Sichtbarkeitsgraph**
- **Potentialfeld**
- Voronoi-Diagramme
- Vektorfeld-Histogramm
- Bug - Algorithmus
- ...

Diskreter Suchraum

- Aufgabe der Pfadplanung:
  1. *Erzeugung eines Zustandsraums* steht im Vordergrund.
  2. Zustandsraum markiert die Zellen, in denen der AMR sich bewegen kann.
- Wurde die Umgebung (Welt), in der der AMR sich bewegen kann, in einen Zustandsraum transferiert, dann existieren geeignete *Such*- und *Optimierungsalgorithmen*, um einen geeigneten Pfad zwischen beiden Positionen, z.B. Start- und Zielpunkt des AMR zu ermitteln.

Erzeugung Zustandsraum I

Zur Erzeugung des Zustandsraums bietet sich ein einfacher Ansatz an:

1. Man betrachtet eine geometrische Repräsentation des freien Raums und *diskretisiert* ihn mit einem bekannten Verfahren.
2. Danach erzeugt man einen Graphen, der benachbarte Zellen verbindet. Abb. 1 zeigt die Konstruktion eines solchen Graphs.
3. Zellen, die keine Hindernisse für den AMR erzeugen werden entfernt und dann wird der resultierende Graph nach einem geeigneten Pfad abgesucht.
4. Diese Methode eignet sich für große Suchräume - auch wenn der Arbeitsraum des AMR mit hoher Auflösung betrachtet wird.

Erzeugung Zustandsraum II

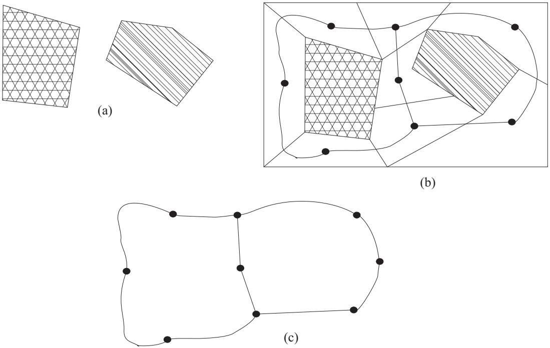

Abbildung: Diskreter Suchraum mit Graph-Repräsentation
a) Umgebung, b) Diskretisierung, c) Graph mit den diskreten Regionen als Knoten

Sichtbarkeitsgraph I

Die Methode des *Sichbarkeits- oder V-Graphen* erzeugt einen Pfad **minimaler** Länge zwischen Start- und Endpunkt mittels Graph-Durchwanderungsalgorithmus.
Der Sichtbarkeitsgraph $G = (V, E)$ wird definiert und erzeugt wie folgt (siehe Abb. 2):

1. V ist eine Menge von Eckpunkten, die sich aus der Vereinigung aller Eckpunkte der polygonalen Hindernisse in der Umgebung plus dem Start- und dem Zielpunkt ergibt.
2. Die Menge der Kanten E ergibt sich aus allen Kanten, die die Eckpunkte verbindet, die zueinander sichtbar sind; dabei wird also keine Hindernis geschnitten.
3. Der Graph G ergibt sich nun als Untermenge der Eckpunkte der Hindernisse und Start- sowie Zielpunkt als Knoten und den Kanten, die Orte verbinden, die direkt ohne Hindernisberührung erreicht werden können.

Sichtbarkeitsgraph II

4. Den kürzesten Pfad erhält man als den Pfad, der Start- und Zielpunkt auf kürzestem Weg im resultierenden Graph verbindet.

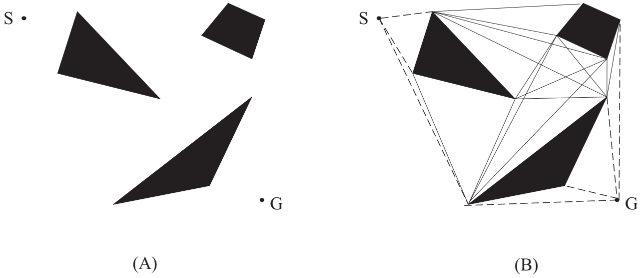

Abbildung: Sichbarkeitsgraph $G = (V, E)$ mit generierten Kanten E
a) Eckpunkte der Hindernisse, Start- und Zielpunkt, b) Kanten E des Graphen (Kanten der Hindernisse gehören dazu, nur die gestrichelten Kanten ändern sich bei anderen Start- und Zielpunkten)

Sichtbarkeitsgraph III

- Da kein Pfad auf dem endgültigen Graphen durch einen konkaven Eckpunkt des betroffenen Hindernisses geht, braucht man nur konvexe Eckpunkte zu betrachten.
- Man kann zeigen, dass man weiterhin nur vier Kanten zwischen zwei Objekten betrachten muss: die sind kotangent zu den beiden Objekten (siehe Abb. 3).

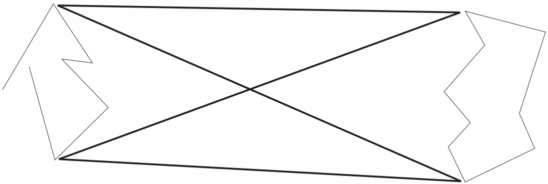

Abbildung: Tangentengraph

Potentialfeld-Methode I

- Der Roboter wird einem virtuellen Potentialfeld ausgesetzt, das in quasi ins Ziel ziehen bzw. lenken soll.
- Dabei wird der AMR als geladenes Partikel unter dem Einfluss eines Potentialfeldes U betrachtet.
- Hindernisse werden ebenfalls als geladene Teile mit gleicher Ladung wir der AMR betrachtet – sie stoßen sich also ab (Kraft F proportional zum negativen Gradienten).

Abb. 4 zeigt ein praktisches Beispiel.

Potentialfeld-Methode II

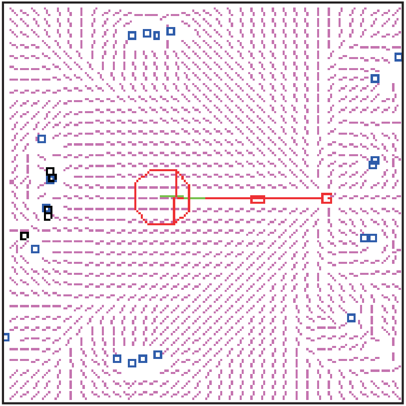

Abbildung: Potentialfedfunktion ohne Hindernisse

Potentialfeld-Methode III

1. Man betrachtet also zwei Arten von Potentialfunktionen: **anziehende** und **abstoßende**.
2. Die anziehende Potentialfunktion U_g modelliert die Zielkonfiguration; sie ist so definiert, dass die auf den Roboter wirkende Kraft F zur Zielkonfiguration $\mathbf{q}_g$ gerichtet ist.
3. Dies lässt sich mit Hilfe einer Potentialfunktion $U(\mathbf{q})$ beschreiben:

$$U(\mathbf{q}) = U_g(\mathbf{q}) \quad .$$

Hindernisse I

1. Enthält der Konfigurationsraum zusätzlich Hindernisse, so werden diese über eine zweite, abstoßende Potentialfunktion U_{O_i} pro Hindernis O_i modelliert.
2. Man erhält die resultierende Potentialfunktion somit zu:

$$U(\mathbf{q}) = U_g(\mathbf{q}) + \sum U_{O_i}(q) \quad .$$

3. Das Kraftfeld F, das auf den AMR wirkt und ihn ins Ziel zieht, erhält man durch Differentiation von U (muss ableitbar sein, negativer Gradient, da Zielpunkt $\mathbf{q}_g$ kleinstes Potential):

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

Abb. 5 zeigt das resultierende Feld mit Hindernissen.

Hindernisse II

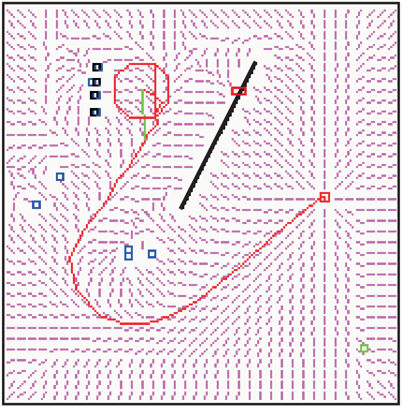

Abbildung: Potentialfedfunktion mit Hindernissen

Hindernisse (2) I

- Abstoßende Potentiale repräsentieren Hindernisse;
- damit ein sicherer Abstand D_O eingehalten wird, wächst die abstoßende Kraft mit abnehmender Entfernung $d_{O_i}(q)$ zum Hindernis.

Die abstoßenden Potentialfunktionen U_{O_i} erhält man zu:

$$U_{O_i} = \begin{pmatrix} c_2\left(\frac{1}{d_{O_i}(\mathbf{q})} - \frac{1}{D_0}\right)^2 & | & \text{falls } d_{O_i}(\mathbf{q}) \leq p_0 \\ 0 & & \text{sonst} \end{pmatrix} .$$

- Große praktische Probleme können lokale Minima in $U(\mathbf{q})$ darstellen, die durch die Überlagerung von anziehenden und abstoßenden Potentialfunktionen entstehen. Abb. 6 zeigt ein praktisches Beispiel.

Hindernisse (2) II

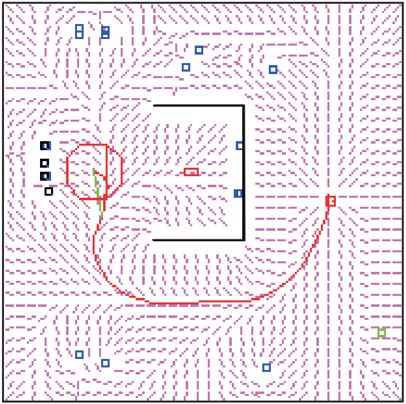

Abbildung: Lokale Minima in Potentialfunktion durch U-förmiges Hindernis