Vorlesung: Einführung in die Robotik

Prof. J. Zhang
zhang@informatik.uni-hamburg.de

Universität Hamburg
Fachbereich Informatik
AB Technische Aspekte Multimodaler Systeme

13. Januar 2005

Grundlage zur Programmierung auf Aufgabenebene

Ziel: die Aufgaben-Spezifikation mit symbolisch beschriebenen Zuständen zu ermöglichen,

wobei: die Planung der notwendigen Bewegung dem Robotersystem zu überlassen.

Bezüglich der Roboterbewegung soll es lediglich nötig sein, einem Roboter zu befehlen, wohin er sich bewegen soll, anstatt zu spezifizieren, wie er sich genau bewegen soll.

Ein zentrales Problem der Programmierung auf Aufgabenebene:
Kollisionsvermeidung

Ein Hauptlösungsansatz:

geometrische Bahnplanung - kollisionsfreie Bewegungen zu planen für die bekannten Manipulator- und Hindernisse-Modelle im Arbeitsraum.

Inhaltsverzeichnis

Objekt-Darstellung

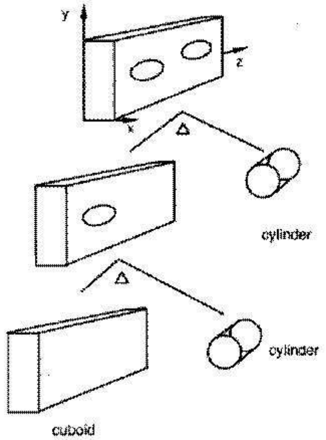
von Robotern,

Umwelt

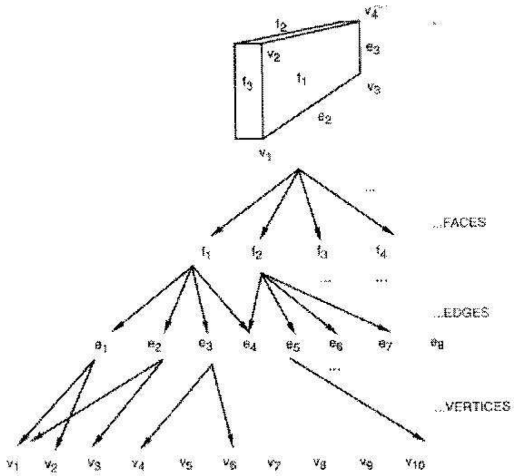
und Konfigurationsräumen

- Approximierte Verfahren:
  Bounding-Box, Konvexe-Hülle, Kugel und Ellipse
- "*Constructive Solide Geometry*" (CSG)
- "Boundary Representation" (BR)
- "Sweep Representation"
- Zellen-Darstellung:
  - Gitter-Modell ("*Spatial Occupancy Enuermation*")
  - Hierarchische Darstellung: Vierer-Baum ("*quadtree*"), Achter-Baum ("*octree*")

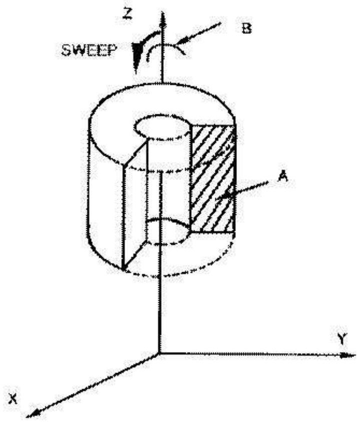
Ein Beispiel der CSG-Darstellung



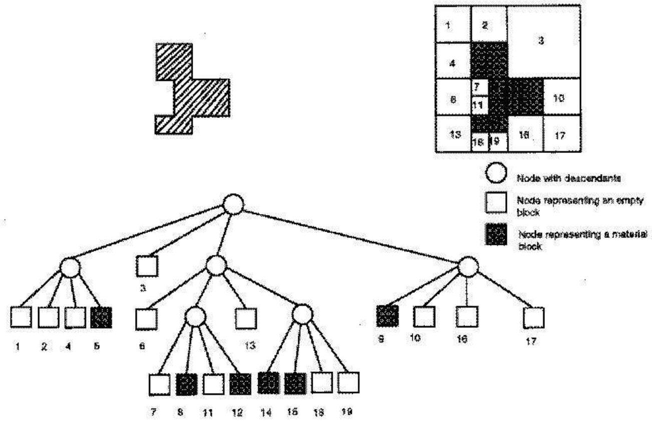
Ein Beispiel der BR



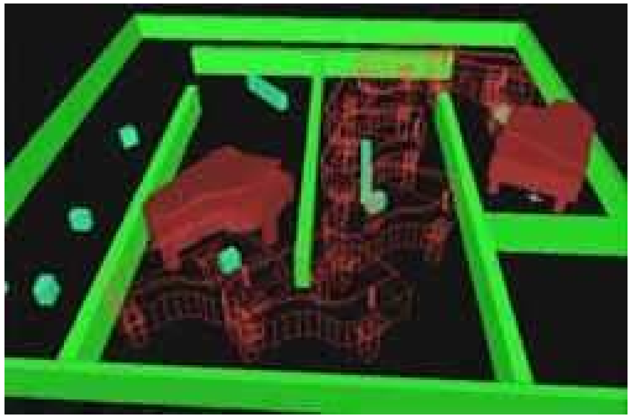
Ein Sweeping-Modell durch Drehung



Eine Quadtree-Darstellung

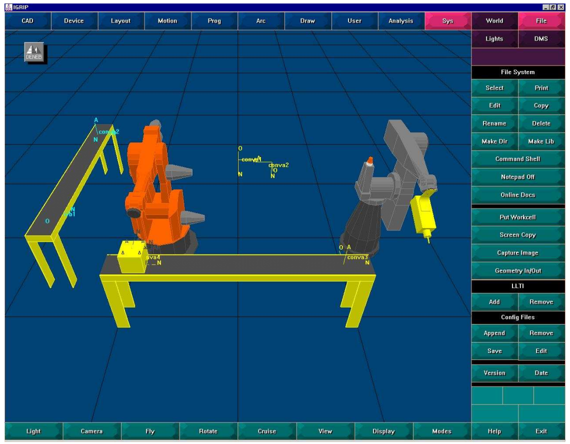


Motivation: “Piano-Mover”

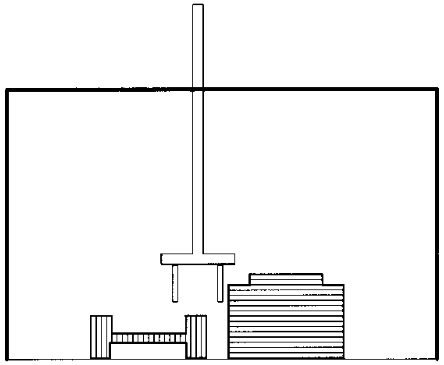


Lösung eines Knobelspiels über einen Computer?

Motivation: Roboterprogrammierung



Motivation: Ausrichten eines Greifers



Bewegungsplanung

Aufgaben umfassen:

1. geometrische Bahnen
2. Trajektorien (die Positions-, Geschwindigkeits- und Beschleunigungsfunktionen in Bezug auf die Zeit)
3. Befehlsreihenfolge bei sensorbasierten Bewegungen

Ziele sind z.B.:

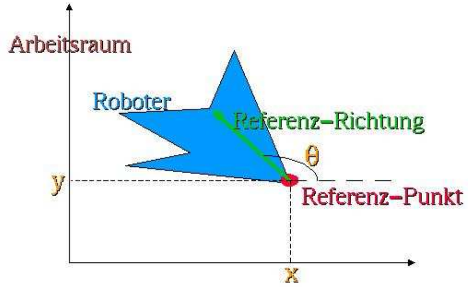
1. kollisionsfreie Bewegung zum Ziel
2. autonome Montage eines Aggregates
3. Raumkognition

Konfiguration eines Artefaktes

Artefakt: ein virtueller oder realer Körper, der seinen Ort und/oder seine Form zeitlich verändern kann.

- Eine *Konfiguration* eines Artefaktes: eine Menge unabhängiger Parameter, welche die Positionen aller seiner Punkte bezüglich eines Referenzkoordinatensystems spezifizieren.
- Darstellbar durch einen geometrischen Zustandsvektor.
- Die Anzahl der Parameter für die Spezifikation einer Konfiguration: Freiheitsgrade.

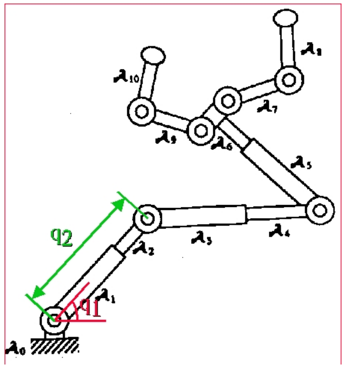
Konfiguration eines starren Körpers

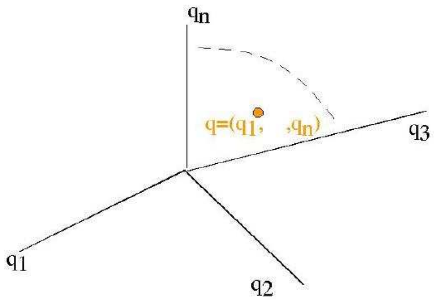


Die Konfiguration des Objektes: (x, y, θ)

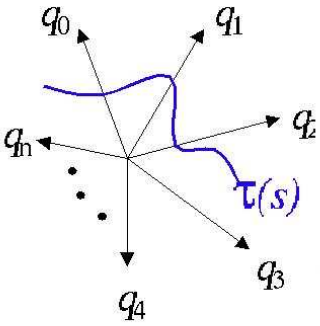
In einem 3D Arbeitsraum: $(x, y, z, \alpha, \beta, \gamma)$

Konfigurationen eines Mehrgelenkarms





Konfigurationen und Bahn eines Menschen



Eine Bahn: eine stetige Kurve, die zwei Konfigurationen verbindet

$$\tau : s \in [0,1], \tau(s) \in Konfigurationsraum$$

Planung geometrischer Bahnen: Definition

Definition des Basisproblems (*generalized mover's problem*):

> „Werden eine begrenzte Anzahl m von statischen Hindernissen und ein Artefakt mit d Freiheitsgraden vorgegeben, dann besteht die Aufgabe der geometrischen Bahnplanung darin, eine kollisionsfreie Bahn zwischen zwei Konfigurationen zu bestimmen.“

Ein *vollständiger* Bahnplaner soll immer einen zulässigen Plan liefern, wenn ein solcher existiert, und ansonsten eine Aussage über die Nichtexistenz einer Bahn treffen.

Planung geometrischer Bahnen: Ein- und Ausgabe

Bekannt seien:

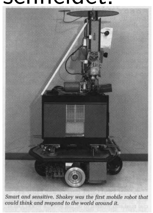
- vollständig *a priori* modellierte Geometrie des Artefaktes sowie der Hindernisse
- Kinematik des Artefaktes (eines starren bzw. formveränderlichen Körpers)
- Start- und Zielkonfiguration

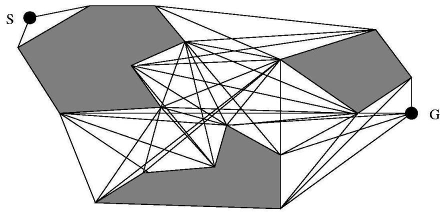
Zu berechnen sei:

- eine Reihenfolge von stetigen Verbindungen kollisionsfreier Konfigurationen des Artefaktes von der Start- zur Zielkonfiguration.

Sichtbarkeitsgraph

Der Sichtbarkeitsgraph wird durch Verbindung der sichtbaren Eckpunkte der Hindernisse konstruiert, wobei die Sichtbarkeit zweier Eckpunkte bedeutet, daß die Verbindungsgerade der beiden Punkte kein Hindernis schneidet.





(Nilsson 69)

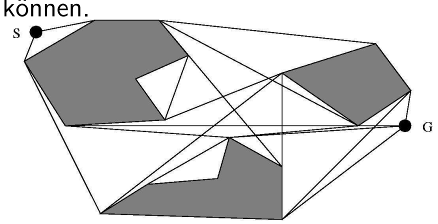
Komplexität: $O(m^2)$ (m: Anzahl der Eckpunkte der Hindernispolygone)

Tangentengraph

Der T-Graph wurde als Teilgraph des Sichtbarkeitsgraph en eingeführt.

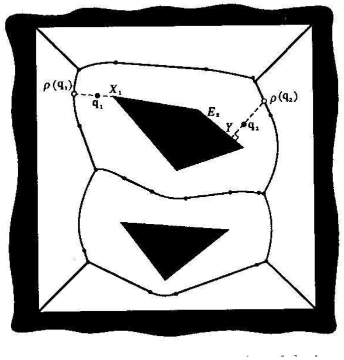
Dabei wurde bewiesen, daß jede kürzeste Route zwischen Start und Ziel eine Teilmenge der T-Graph-Kanten ist.

Die Haupteigenschaft des T-Graphen gegenüber dem Sichtbarkeitsgraphen liegt darin, daß die für d ie Routenplanung nicht benötigten Kanten eliminiert werden können.



Komplexität: $O(m^2)$ (m: Anzahl der Eckpunkte der Hindernispolygone)

Voronoi-Diagramm



Komplexität: Knoten: $O(m)$ (m: Anzahl der Eckpunkte der vieleckigen Hindernisse), Erstellungszeit: $O(m \log m)$

Heuristische Suche

Für die Routensuche im einem Graphen wird der A^*-Algorithmus verwendet.

Man sucht ausgehend von dem Knoten $\mathbf{s}$, in dem der Startpunkt liegt, eine Route zu jenem Knoten $\mathbf{z}$, in dem der Zielpunkt liegt.

Der A^*-Algorithmus benutzt eine heuristische Kostenfunktion f, die jeder Route vom Startknoten zu einem Knoten $\mathbf{q}$ einen Wert zuordnet, der die Gesamtkosten der Route vom Start- zum Zielknoten über diesen Knoten $\mathbf{q}$ abschätzt.

Dabei läßt sich f als additive Verknüpfung zweier Funktionen g und h wählen, wobei g die bekannten Kosten der Route vom Startknoten zu $\mathbf{q}$ angibt und h die geschätzten Kosten der kürzesten Route von $\mathbf{q}$ zum Zielknoten angibt.

Wenn h so gewählt wird, daß die wahren Kosten nicht überschätzt werden,

dann wird der Suchalgorithmus als A^* bezeichnet. Es wird garantiert, daß die kürzeste existierende Route mit dem A^*-Algorithmus gefunden werden kann.

Um nicht nur die kürzeste, sondern auch die glatteste Route zu finden, beinhalten die Kosten einer Route außer der euklidischen Länge auch einen Faktor für Richtungsänderungen. g and h sind exakt folgendermaßen definiert:

- $g=$ euklidische Länge der Route vom Start $\mathbf{s}$ zu $\mathbf{q}$
  $+$ Gewichtsfaktor * Krümmungsmaß;

- h =euklidische Länge der Route von $\mathbf{q}$ zum Zielpunkt
  $+$ Gewichtsfaktor * (das geschätzte Krümmungsmaß)

Alle möglichen Kandidatenrouten von $\mathbf{s}$ nach $\mathbf{q}$ werden in eine Offen-Liste eingefügt.

Jedesmal wird eine Kandidatenroute aus der Offen-Liste herausgeschoben, deren f-Wert minimal ist.

Diese Kandidaten route wird dann mit allen freien Nachbarknoten expandiert, und die neue f Funktion wird dazu ausgewertet.

Dies geschieht so lange, bis der Zielknoten erreicht ist – eine Route wird gefunden, oder wenn die Offen-Liste leer ist – dann es gibt keine Route von $\mathbf{s}$ nach $\mathbf{z}$.

Untere und obere Schranke der Bahnplanungsalgorithm

- Die erste untere Schranke: *PSPACE*-hart, d.h. mindestens so schwer wie ein *NP-Problem* – im ungünstigsten Fall eine exponentielle Rechenzeit für jeden Algorithmus zur Lösung des Problems (Reif 79).
- Die erste obere Schranke: zweifach exponentielle Zeitkomplexität mit dem Freiheitsgrad d (Schwartz und Sharir 87).
- Die zweite obere Schranke: einfach exponentielle Rechenzeit mit dem Silhouetten-Verfahren (Canny 88).