

Labor Robotik - Repräsentation

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

Fachbereich Informatik
Hochschule Trier

WS17/18



LABORROBOTIK
Informatik - Computer Science

Informatik
Hauptcampus

H O C H
S C H U L E
T R I E R

AMR Repräsentation

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).

Repräsentation - Inhalte I

U.U. können einige z.T. auch komplexe Aufgaben ohne Darstellung der "Umwelt" geplant und durchgeführt werden – in der Regel wird aber eine Darstellung benötigt. Diese enthält z.B.:

- ▶ eine Karte über die Räume, Plätze und Objekte der Umgebung,
- ▶ Informationen über die Objekte (Eigenschaften),
- ▶ unsichere Bereiche (schwer oder nicht passierbar),
- ▶ Informationen von früheren "Besuchen" (z.B. Exploration).

Repräsentation - Aufgaben I

Die Repräsentation dient den folgende drei Aufgabenbereichen:

1. Ergründung, welche Bereiche der Umgebung können befahren werden; d.h. sind frei von Hindernissen und frei für die Navigation (*free space*),
2. Erkennung der Bereiche und Objekte in der Umgebung,
3. Erkennung spezifischer Objekte in der Umgebung (Wände, Ecken, ... Türen usw.).

Umgebungsdarstellung I

Somit ergibt sich die Frage, in welcher Form die Umgebung darzustellen ist. Wie oftmals, gibt es einfache und komplexe Lösungsansätze:

Selbstdarstellung: Es gibt **keine interne** Repräsentation (sehr einfach!?) – die Umgebung stellt sich selbst dar: sie wird mit den Sensoren aufgenommen und ausgewertet (Kontrollprinzip: Subsumption, reaktiv).

Planung und Aktion ergeben sich abhängig von der externen (Umwelt-) Information, weswegen eine Langzeitplanung kaum möglich ist; statt dessen: reaktive Planung (*low level, real time*).

Umgebungsdarstellung II

Interne Darstellung: Je nach Ansatz (und Aufwand) werden sogen. Elementarobjekte (Primitive wie z.B. Objekte, -Eigenschaften, symbolische Beschreibung, Raumgitter mit Belegungs-Information, aufgabenspezifische Routinen usw.) zur Beschreibung eingesetzt.

Für die Beschreibung des Raums (Umwelt) kann man grundsätzlich zwei Ansätze unterscheiden:

Metrische (räumliche) und topologische Repräsentation.

Umweltdarstellung räumlich I

Die Beschreibung einer Umgebung erhält man am einfachsten, wenn man den Raum diskret (2D- oder 3D) abtastet.

- ▶ Dadurch wird der Raum selbst und nicht über seine einzelnen Objekte dargestellt.
- ▶ Damit muss man auch die Objekte nicht erkennen und identifizieren – was sehr aufwendig sein kann.

- ▶ Die Abtastung kann auf unterschiedliche Art und Weise erfolgen – eine einfache Methode ist:

Man definiert ein regelmäßiges Abtastraster (-gitter) über den Raum und erfasst (tastet ab) den Raum in den einzelnen Rasterzellen: **leer** oder **besetzt** (siehe Abb. 1).

Alternativ: Unterteilung des Rasters anhand der vorliegenden Objekte.

Umweltdarstellung räumlich II

- ▶ Ist die Abtastinformation binär, dann erhält man bei 2D-Darstellung sogen. *bitmaps*.
- ▶ Enthält die Abtastinformation Besetzungsgrad usw., dann erhält man bei 2D-Darstellung sogen. *pixel maps* oder Besetzungs-Gitter (*occupancy grids*).
- ▶ Neben der Besetzung der Raums bzw. bestimmter Bereiche, kann die pixel map noch weitere Information bereit stellen; wie z.B.:
Zuverlässigkeit der Besetzung, Sicherheit oder Wunsch, eine Zelle zu besetzen, "Gelände"-Information usw.

Umweltdarstellung räumlich III

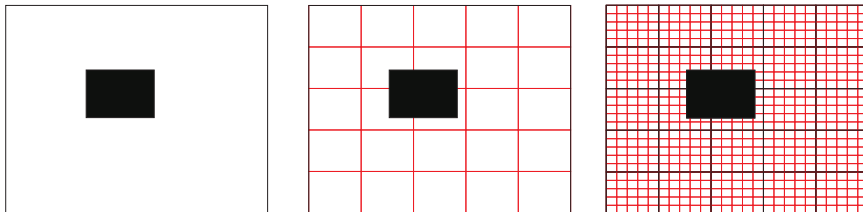


Abbildung: Raster-Darstellung eines 2D-Raums

Vor- und Nachteile I

Je nach Darstellungsart ergeben sich **Vor-** und **Nachteile**:

- ▶ Gittermodelle sind sehr generell (einsetzbar), da keine feste Annahmen über den Objekttyp (Zelle) gemacht werden.
Sie können somit beliebige Dinge darstellen.
- ▶ Nachteilig ist die begrenzte Auflösung (Rasterauflösung) und der hohe Speicherbedarf (Bsp.: ein Raum der Dimension $100 \times 100 \times 100 \text{ m}^3$ benötigt bei einer (geringen) Auflösung von $\Delta x = 1 \text{ m}$ bereits ein 3D-Gitter mit 10^6 Zellen (*voxel*) – bei $\Delta x = 10 \text{ cm}$ sind es bereits 10^9 voxels).
Dies ist dann noch besonders Speicher-ineffizient, wenn der Raum weitestgehend leer ist (Zellen: *free*).

Vor- und Nachteile II

Da die Rasterdarstellung so einfach aber leider auch speicherineffizient ist, versucht man günstigere Darstellungsformen dafür zu finden.

Dabei benutzt man die Eigenschaften, dass die Zellen gleiche Eigenschaften $\{empty, full\}$ haben und lokal benachbarte Gebiete häufig den gleichen Status haben.

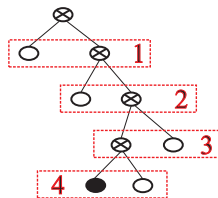
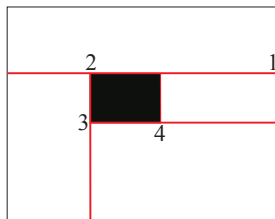
Hierzu können zwei unterschiedliche Ansätze betrachtet werden:

1. Verwendung uneinheitlicher Gitterabstände und -form; dies ist schwierig umzusetzen, weswegen meist
2. rekursive hierarchische Darstellungen benutzt werden. Hierdurch lassen sehr oft der Speicherbedarf und auch der Rechenaufwand drastisch reduzieren.

Beispiele hierfür sind: **Quadtrees** und **BSP** (*binary space partitioning trees*).

Binäre Raumteilung (BSP) I

BSP trees sind in der Computer Grafik häufig benutzte hierarchische Darstellungen, die man jedoch für andere Aufgaben einsetzen kann (siehe Abb. 2).



○ homogen leer

● homogen besetzt

⊗ inhomogen

Abbildung: BSP-Darstellung eines 2D-Raums

Binäre Raumteilung (BSP) II

- ▶ BSP trees erzeugen eine hierarchische Darstellung, bei der eine Rasterzelle Blatt des Baums ist oder wieder in zwei BSP-Bäume aufgeteilt wird,
- ▶ die Aufteilung erfolgt immer mit einer Ebene parallel zu einer der externen Grenzen.
- ▶ der freie Bereich (des 2D-Raums) wird in Regionen mittels parallelen Linien zu den äußeren Grenzen der Umwelt abgegrenzt.
- ▶ Jede neue Grenze teilt eine Zelle in zwei – aber nicht unbedingt im Zentrum einer Region.

Zusammenfassung

Eigenschaften der betrachteten Darstellungsmethoden für 2D-Räume

- ▶ Quadrees liefert eine Aufteilung nach Teilquadranten-Größen in Potenzen von 2; nicht alle Umweltgrenzen passen in dieses Modell: z.B. ein um 45° gedrehtes Rechteck. Hierdurch erhält unterschiedliche Repräsentationen selbst bei gleicher Form der Umwelt.
- ▶ BSP trees und exakte Raumteilung liefern keine eindeutige Darstellung eines 2D-Raums. Abhängig von der freien Wahl der Teilungsebenen erhält man völlig unterschiedliche hierarchische Darstellungen.

Geometrische Darstellungen I

Im Gegensatz zur räumlichen (*spatial*) Darstellung der Umwelt benutzen *Geometrische Karten* geometrische Elementarobjekte (Primitive) wie z.B. Punkte, Linien und Polygone oder (polynomial) Funktionen. Dies hat einige Vorteile, wie z.B.

- ▶ eine sehr gute Speichereffizienz, da ein beliebig großer Raum sich mit einem Modell mit einigen wenigen Parametern darstellen lässt (siehe Abb. 3).
- ▶ Darstellung/Speicherung von Besetzungsdaten mit beliebig feiner Auflösung ist praktisch möglich (ohne auf Speicherbegrenzungen wie bei der räumlichen Darstellung zu stoßen).
- ▶ Geometrische Karten entstehen aus der Vereinigung von einfachen geometrischen Primitiven wie z.B. Punkte, Polygone, Kreise usw. (siehe Mapper 3).

Geometrische Darstellungen II

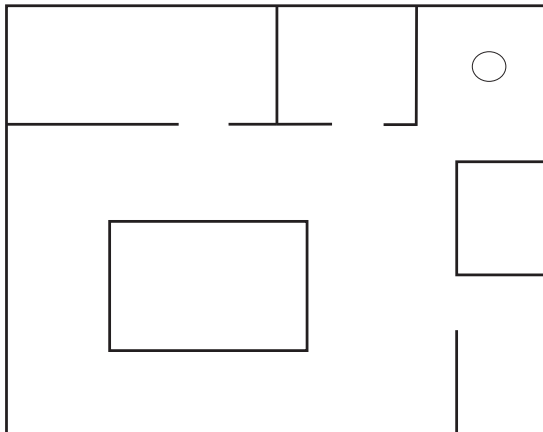


Abbildung: Geometrische Darstellung

Geometrische Karten

Geometrische Karten sind durch zwei wichtige Eigenschaften gekennzeichnet:

1. Die Menge der (geometrischen) Primitiven für die Beschreibung von Objekten, wie z.B. Punkte, Linien (-segmente), Polylinien und Polygone und
2. die Menge von Operatoren zur Manipulation der Objekte wie z.B. starre Transformationen (Translation, Rotation), anpassende Transformationen (Form-erhaltende Transformationen) und affine Abbildungen; Boole-sche Operationen (Vereinigung, Schnitt, als konstruktive feste Geometrie.

Topologische Darstellungen I

Geometrische Modelle sind oftmals korruptiert z.B. wegen des Rauschanteils der Sensordaten. Um diese Unzuverlässigkeit zu vermeiden, kann man nicht-metrische topologische Darstellungen benutzen.

- ▶ Topologische Darstellungen bieten sich an, großräumige Bereiche – ähnlich wie der Mensch das auch tut – darzustellen.
Z.B. Beschreibung eines Weges (FH → Hbf, L03 → Mensa usw.), die eher topologische als metrische Eigenschaften aufweist (gerade aus – Tür öffnen – den Flur entlang usw.).
- ▶ Der Schlüssel zu einer topologischen Beschreibung ist die explizite Repräsentation der Verbindung zwischen Regionen oder Objekten. In der reinsten Form treten dann keine metrische Daten mehr auf.

Topologische Darstellungen II

- ▶ Eine topologische Darstellung basiert auf einer Abstraktion der Umgebung in Termen wie Orte mit Kanten, die sie verbinden.
Ein Graph $G = (V, E)$ mit V : Menge der Knoten, E : Menge der Kanten, die die Knoten verbinden.
Oftmals kann der Graph in einen metrischen Raum eingebettet sein: dann haben z.B. die Kanten Längen und sie sind gerichtet bzgl. der Knoten.
- ▶ Graphen werden sehr häufig in für AMRs eingesetzt.
Dabei korrespondieren die Knoten oft mit Landmarken (z.B. in der Umgebung angebrachte, eindeutige Kennzeichen), die zur lokalen Orientierung und letztendlich zur Navigation verwendet werden.

Darstellung des AMR I

Mobile Roboter selbst haben ja auch eine bestimmte Form und Größe, mit der sie in die Umwelt einzubringen sind und in der sie sich aufgrund ihrer Kinematik bewegen.

Hierzu lassen sich AMRs mit einem so genannten Konfigurationsraum (\mathbf{q}) (*C-space, configuration space*) darstellen – der in vielen praktischen Fällen vereinfacht beschrieben werden kann.

Konfigurationsraum I

Der *C-space* ist ein Formalismus zur Unterstützung der (Bewegungs-) Planung für AMRs.

- Die Konfiguration \mathbf{q} eines Roboters R spezifiziert seinen physikalischen Zustand in Bezug auf einen fixierten Rahmen F seiner Umgebung. Für einen (starren) Roboter R , dessen Kinematik Translation und Rotation ermöglicht, lässt sich seine Konfiguration in der Form

$$\mathbf{q} = (x \ y \ \theta) \quad (1)$$

beschreiben, wobei \mathbf{q} dann eindeutig die Konfiguration des AMR beschreibt. (In anderen Fällen kann \mathbf{q} sehr viel komplexer sein: z.B. AMR mit Beinen.)

- Die physikalische Konstruktion von R kann einige Konfigurationen verhindern – ebenso wie vorhandene Hindernisse in F . Z.B. kann ein (kreisrunder) AMR mit Radius r nicht näher als mit Abstand $d = r$ an ein beliebiges Objekt herankommen.

Konfigurationsraum II

- ▶ Hierdurch ergeben sich also Einschränkungen (*constraints*), die man in folgender Form beschreiben kann:

$$G(\mathbf{q}) = 0 \quad (2)$$

was bedeutet: \mathbf{q} ist die Pose von R , die diese Konfiguration **nicht** ermöglicht.

- ▶ Man nennt dies eine **holonomische Einschränkung** (*holonomic constraint*).
- ▶ Der Konfigurationsraum von R ist der Raum C aller möglichen Konfigurationen von R ; i.a.W.: C definiert alle *zulässigen* Konfigurationen \mathbf{q} von R . Es gilt:

$$C = \{\mathbf{q} | G(\mathbf{q}) \neq 0\} \quad (3)$$

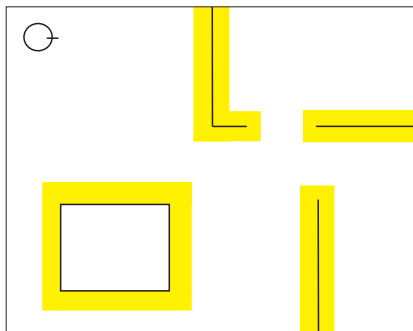
Konfigurationsraum III

- Für jedes Hindernis B_i in der realen Welt ($\in C$) kann man die holonomischen Einschränkungen mit $G(\mathbf{q}) = 0$ eine Menge von Punkten in C überführen.

Dieser Bereich ist also von R wegen realer Hindernisse nicht erreichbar: man nennt dies ein Konfigurationsraum-Hindernis. Der so reduzierte Konfigurationsraum C_B ergibt sich dann zu (Abb. 4):

$$C_B = \{\mathbf{q} \in C \mid R(\mathbf{q}) \cap B_i \neq \emptyset\} \quad , \quad i \in [1, n] \quad . \quad (4)$$

Konfigurationsraum IV



■ holonomische Einschränkung $G(q)$

Abbildung: Holonomische Einschränkung: Objekterweiterung

- Die Planung eines Pfads mit *C-space*-Hindernissen erfolgt systematisch.

Konfigurationsraum V

- ▶ Der analoge, hindernisfreie Raum in *C-space* nennt man C_{frei} .
- ▶ Besitzt ein Roboter Einschränkungen aufgrund seiner Kinematik (Differenziale $d^i \mathbf{q} / dt^i$ können nicht integriert werden), dann spricht man von so genannten **nicht-holonomischen Einschränkungen** (*nonholonomic constraints*).
- ▶ Nicht-holonomische Einschränkungen lassen sich also in folgender Form definieren:

$$G(\mathbf{q}, \frac{d\mathbf{q}}{dt}, \frac{d^2\mathbf{q}}{dt^2}, \dots) = 0 \quad . \quad (5)$$

- ▶ Diese Einschränkungen definieren also, welche Geschwindigkeiten (und Richtungsänderungen bzw. Tangenten) in *C-space* erlaubt sind.
- ▶ Nicht-holonomische Einschränkungen reduzieren den Bereich erlaubter (verfügbarer) differenzieller Bewegungen!

Konfigurationsraum VI

- ▶ Diese Einschränkungen erschweren die Bewegungs-Planung außerordentlich:
Die Bewegung von einem erlaubten Position (\mathbf{q}) zu einer anderen kann eine komplexe Bahn (Trajektorie) erfordern – selbst wenn die beiden Positionen "benachbart" sind.
- ▶ Beispiel: Der Kurvenradius, der sich aufgrund einer Begrenzung des Drehradius, den ein AMR einnehmen kann, ergibt, stellt eine nicht-holonomische Einschränkung dar.