

Typische Fragestellungen aus 3D-Modellierung und mobiler Robotik

Christian Rink

German Aerospace Center (DLR) Institute of Robotics and Mechatronics

28.4.2014

Knowledge for Tomorrow



Contents

1. Persönlicher Hintergrund
2. Monte Carlo Lokalisierung
3. Objektlageschätzung
4. Exploration und Kartierung
5. Ausblick



Persönlicher Hintergrund

allgemein

- Dipl. Math. an TU München (2004), 'Nichtkinematische Algebren mit Quadrikenmodell vom Typ A'
- Master in Statistik an LMU München (2007), 'Globale merkmalsbasierte Registrierung homogener Dreiecksnetze'
- seit 2007 (2005) am Institut für Robotik und Mechatronik, Abteilung Perzeption und Kognition
- Arbeitsbereiche: 3d-Modellierung, mobile Robotik
- entwickeln und programmieren von Software, c++, python, arbeiten mit Hardware, technische Anwendung statistischer Verfahren



Persönlicher Hintergrund

bearbeitete Fragestellungen

- Lageschätzung / Registrierung
- 3D-Modellierung / Kartierung unbekannter Gebiete
- Lokalisierung mobiler Roboter
- Exploration
- Pfadplanung
- Rotationen



Autonome Mobile Roboter

- Zweck: Exploration, Kartierung und Navigation in
 - Katastrophengebieten (Erdbeben, Tsunami, Atomkraftwerke etc.)
 - anderweitig unzugängliche Gebiete (Höhlen, andere Planeten)
 - Haushalten und Fertigungsstätten der Industrie (Servicerobotik)
 - Verkehr und Landwirtschaft (Navigationssysteme)
- Anforderungen/Fähigkeiten:
 - Erkundung und Kartierung unbekannter Umgebungen (Exploration, Mapping)
 - Bestimmung der Position und Lage in bekannten Umgebungen (Lokalisierung)
 - Navigation zu vorgegebenen Zielen in Karten (Navigation)
 - gleichzeitige Kartierung und Lageschätzung in der Karte (slam - simultaneous localization and mapping)



Lokalisierung mobiler Roboter

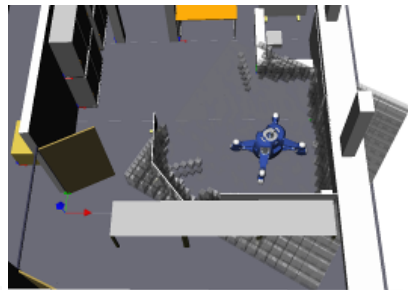
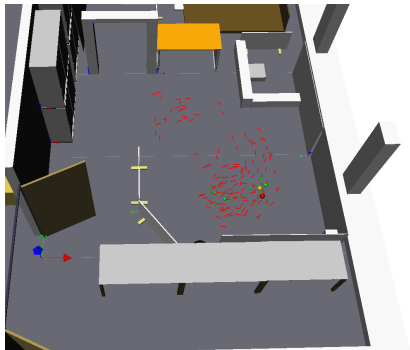
basics

- Problem: schätze die Lage eines mobilen Systems in einer bekannten Umgebung
 - System/Roboter ist ausgestattet mit Sensoren für Umgebungswahrnehmung, insbesondere Abstandssensoren (Laser)
 - Umgebung wird repräsentiert durch ein 3D-Modell (in dem die Simulation der Abstandssensoren möglich ist)
- Ziel: Roboter sollten autonom in bekannten Umgebungen navigieren können (ohne menschliche Hilfe)
- Anwendungen: Serviceroboter in Industrie (Logistik), Haushalt (Staubsauger, Rasenmäher), Verkehr (Navis), Landwirtschaft (Mähdrescher mit GPS-Programmierung)
- Lösungsmöglichkeit: rekursive Zustandsschätzung mit Bayes-Filtern (Kalmanfilter, Partikelfilter)



Monte Carlo Lokalisierung

Partikelfilter und Kartierung

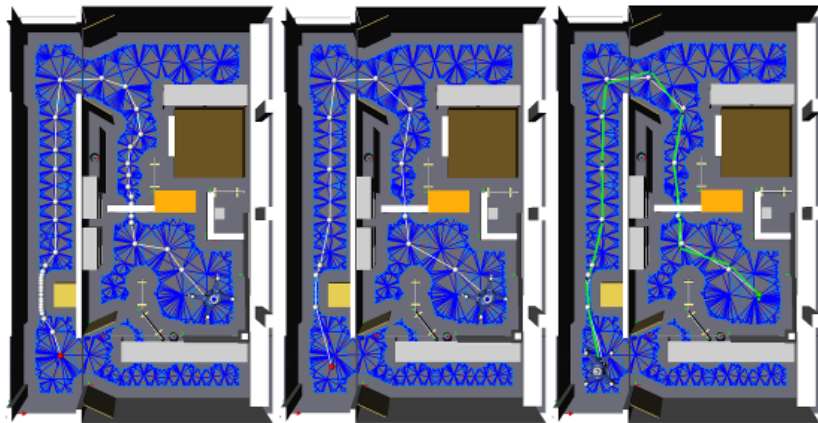


links: Partikel in einer 3D-Karte, rechts: Lokale Karte des ML-Partikels



Monte Carlo Lokalisierung

Navigation



Pfadplanung, Pfadoptimierung, und tatsächlich gefahrener Pfad



Monte Carlo Lokalisierung

Problemstellung

- gegeben: Karte der Umgebung, Messung der Roboterbewegung (z.B. Radodometrie), Abstandssensormessungen (Laser, Kamera)
- gesucht: Lage X des Roboters der gegebenen Karte M
- Lage des Roboters X_t zum Zeitpunkt t besteht aus Position (Translation) und Orientierung (Rotation) in M
- Bewegung: Übergangsfunktion zum Zeitpunkt t :

$$X_t = U_t(X_{t-1}) + \epsilon_t$$

mit Verteilung von ϵ_t , aus der man Realisierungen simulieren kann

- bekannte Dichte der (Abstandssensor-) Messungen Z_t gegeben die Karte M und die Lage X_t :

$$P(Z_t|X_t, M)$$



Monte Carlo Lokalisierung

grundlegende Idee

- ML-/MPD-Schätzung mit $P(X|Z, M, U)$
- Idee: iteratives/rekursives Bayes-Update:

$$P(X|Z, M, U) \propto P(Z|X, M, U) \cdot P(X|M, U)$$

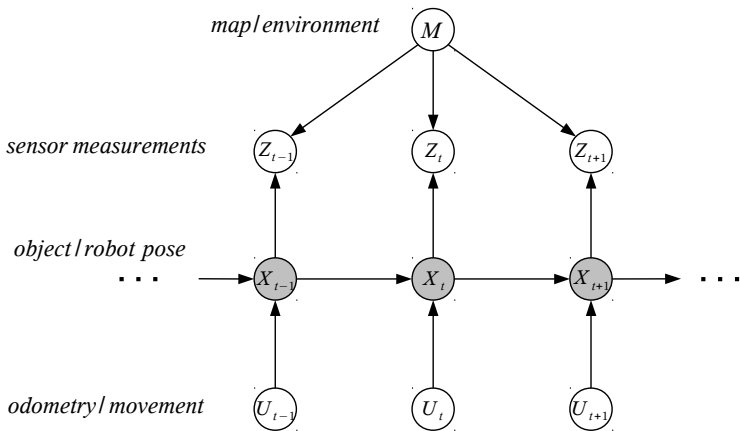
mit unbekanntem Lagen X , Messungen Z , Odometrie U Karte M

- Markov(1)-Annahme vereinfacht die Situation



Monte Carlo Lokalisierung

Bayes-Netzwerk



$$P(X_{t+1}|X_t, U_{t+1}, M, Z_{t+1}) \propto P(Z_{t+1}|X_{t+1}, M) \cdot P(X_{t+1}|X_t, U_{t+1}, M)$$



Monte Carlo Localization

Basisalgorithmus eines Partikelfilter

- Diskretisierung der Dichte $P(X|Z, M, U)$ durch Partikel (gewichtete Samples/Realisierungen von X)
- MCL eine Art "Sampling Importance Resampling" (SIR) mit Vorschlagsdichte $P(\cdot|X_t, U_{t+1}, M)$
- Typischer Algorithmus
 1. initiales Sampling der Partikel (oft gleichverteilt): X_0
 2. Roboterbewegung: bewege Partikel entsprechend dem Bewegungsmodell (Sampling)

$$X_{t+1} = U_{t+1}(X_t) + \epsilon_{t+1}$$

3. Sensormessungen: gewichte Partikel mit $P(Z_{t+1}|X_{t+1}, M)$
4. Erneuerung der Partikel entsprechend den Gewichten (Resampling)
5. weiter mit Schritt 2



Monte Carlo Localization

Ergebnis - Vergleich mit SIR

SIR:

- Sampling von x aus Dichte $g(X)$
- Resampling mit Gewicht (importance weight) $f(x)/g(x)$
- Ergebnis: Samples y aus $f = f/g \cdot g$

hier:

- Sampling aus Dichte $g = P(X_{t+1}|X_t, U_{t+1}, M)$
- Resampling mit Gewicht $f/g = P(z_{t+1}|X_{t+1}, M)$
- Ergebnis: Samples aus
 $f/g \cdot g = P(Z_{t+1}|X_{t+1}, M) \cdot P(X_{t+1}|X_t, U_{t+1}, M)$

Erinnerung: Ziel war ML-/MPD-Schätzung aus

$$P(X_{t+1}|X_t, U_{t+1}, M, Z_{t+1}) \propto P(Z_{t+1}|X_{t+1}, M) \cdot P(X_{t+1}|X_t, U_{t+1}, M)$$



Objektlageschätzung

in Kürze: Vergleich mit Lokalisierung

Lokalisierung:

- Modell der Umgebung bekannt
- Umgebung wird durch Sensoren wahrgenommen
- Bewegung des Roboters durch die Umgebung

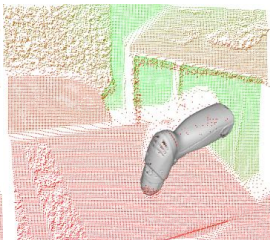
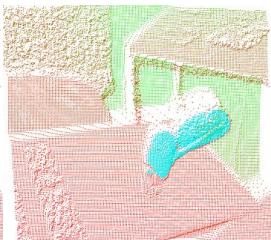
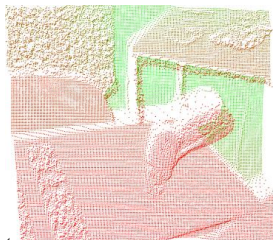
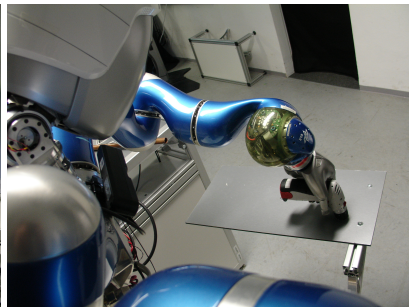
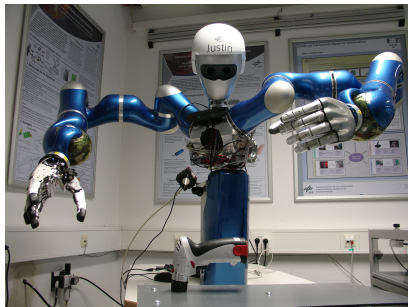
Objektlageschätzung:

- Modell eines Objektes/Gegenstandes bekannt
- Gegenstand wird durch Sensor(en) wahrgenommen
- Bewegung des Sensors um das Objekt



Objektlageschätzung

Anwendungsbeispiel: Greifen eines Akkuschrauber



Objektlageschätzung

in short

- Problem: Lageschätzung eines bekannten Objektes von dem Sensoraufnahmen vorliegen
 - Vereinfachung: Sensorlagen sind bekannt (ohne grossen Fehler)
 - Objekt wird durch ein bekanntes 3D-Modell repräsentiert (z.B. Dreiecksnetz)
- mögliche Ziele: Roboter sollen Objekte autonom greifen, automatische Modellausrichtung von Aufnahmen aus verschiedenen Perspektiven (3D-Modellierung)
- Anwendungen: teilautonome Telepräsenz (s. Anwendungsbeispiel), Objektinspektion (Qualitätskontrolle in der Fertigung), automatisiertes Sortieren von Teilen, robotergestützte Chirurgie, 3D-Modellierung
- Lösungsansätze: vieles möglich, unter anderem merkmalsbasierte Partikelfilter mit Tiefensensoren und Oberflächenmodellen



Exploration und Kartierung

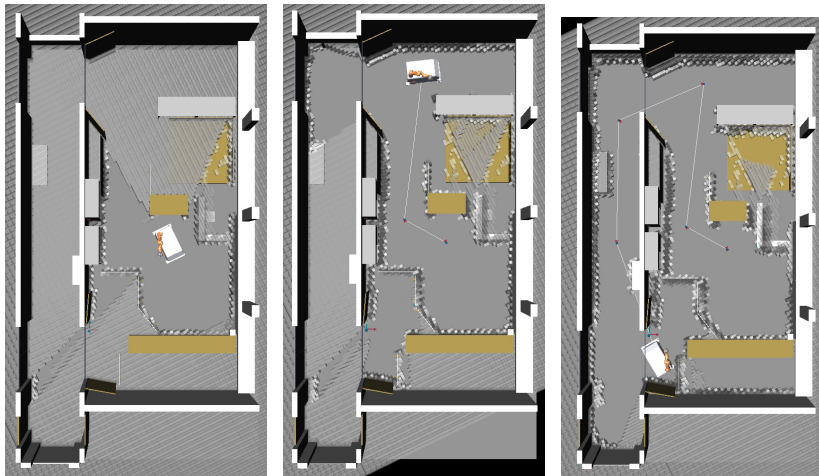
kurzgefasst

- Kartierung: erstelle ein 3D-Modell der Umgebung aus Sensordaten
- Exploration:
 - autonomes Kartieren mit (mobilem) Roboter
 - Planung nächster bester Perspektiven des Sensors
 - Pfadplanung, um diese Perspektiven/Lagen des Roboters zu erreichen
 - Abbruchkriterien



Exploration und Kartierung

Beispiel Mobiler Roboter

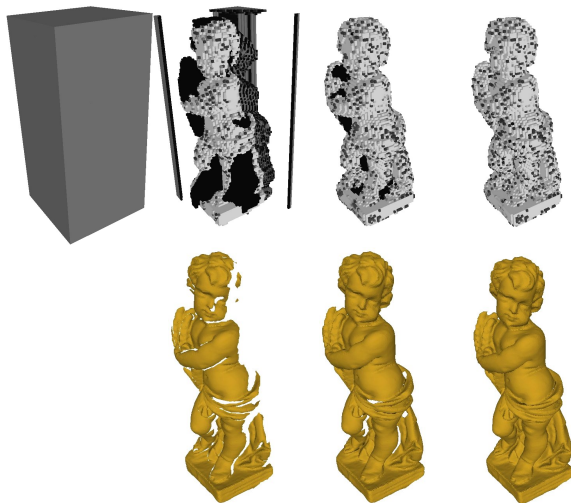


autonome Kartierung: entstehende Karte von Startlage bis zur kompletten Karte



Exploration und Kartierung

Beispiel 3D-Modellierung



autonome 3D-Modellierung: Entstehung Voxel- und Oberflächenmodell



Exploration und Kartierung

Grundlagen Kartierung

- verschiedene Typen von Karten:
 - metrisch (Oberflächennetze, Voxelkarten)
 - spärlich (Merkmale wie Ecken, Kanten, künstliche Landmarken)
 - topologisch (Verbindung von Räumen, Korridore, Erreichbarkeitsgraphen)
- Beispiel Voxelkarten (metrisch, volumetrisch), die aus Daten von Abstandssensoren generiert werden:
 - Voxel-Grauwerte: Wahrscheinlichkeit, dass Voxel 'besetzt'
 - in mobiler Robotik: inkrementelles update der Karte
 - 'probabilistische' Sensormodelle und einfaches Bayes-update
 - Alternative zu Bayes: Fuzzymengen, Dempster-Shafer (Kombinationsregel)



Exploration und Kartierung

Grundlagen Exploration

- autonomes Kartieren mit mobilem Roboter, Planung der nächsten Aufnahmeperspektiven
- Optimalitätskriterien: Entropie/ expected information gain, Erreichbarkeit, Fahrtkosten etc.
- im Suchraum wird geeignet gesampelt (z.B. Grenzregionen zum Unbekannten)
- fortgeschrittene Methoden berücksichtigen Qualität des erstellten Modells



Ausblick Robotik

Schritthaltende Lageschätzung

- Problem:
 - Lageschätzung eines bekannten Objekts mit Tiefensensoren
 - Objekt bekannt: Modell (Oberflächenmodell, Merkmale) verfügbar
- offline Algorithmus existiert in Form eines Partikelfilters
- nächster Schritt: Lageschätzung beim 'Scannen' des Objekts (während der Datenakquise).
- größte Herausforderung: schritthaltende Berechnung (in 'Echtzeit')



Ausblick Robotik

Exploration: Reduktion der Mehrdeutigkeit vs. information gain (IG)

- Problem:
 - finde nächste beste Lage des Sensors
 - gegeben: 'probabilistische' Voxelkarte (Zustände: Besetzungswahrscheinlichkeiten)
 - Maximierungskriterium für Kandidaten: expected IG (genähert durch Entropie)
- Idee: Kartierung mit Dempster-Shafer existiert, Reduktion von Mehrdeutigkeiten versuchen
- wie unterscheidet sich das Endergebnis?
- gibt es einen Unterschied in der Performance (Zeit, Qualität, Information), falls die Daten schlecht sind?



Ausblick Robotik

Next-Best-View-Planning für autonome Kamerakalibrierung mobiler Systeme

- Problem:
 - Bestimmung nächster Aufnahmeposition eines mobilen Roboters relativ zu einem Kalibrierkörper
 - gegeben: 3D-Modell des Kalibrierkörpers, kalibrierter Referenzsensor (Laserscanner)
 - Kalibrierung: mit Lageschätzungsalgorithmus
- Idee: entscheidungstheoretische Formulierung
- können grundlegende Algorithmen der Entscheidungstheorie angewendet (und implementiert) werden?



Ausblick Intervallwahrscheinlichkeiten

datenbasiertes sequentielles Entscheiden unter Unsicherheit

- klassische Entscheidungstheorie:
 - Gewinn-/Verlustfunktion auf Zuständen der Welt und den Aktionen
 - welche Aktion ist die beste?
 - oft: Wissen über Zustände durch Verteilungsfunktionen modelliert
 - Lösungen: minimax, maximin, bayes etc.
 - datenbasiertes Entscheidungsproblem (und Lösung) äquivalent zu einem datenfreien Entscheidungsproblem (nach Bayes-Update)
- Entscheiden unter Unsicherheit
 - Lösung des datenbasierten Problems nicht äquivalent zu datenfreiem Problem
 - effiziente Algorithmen fehlen, Vorarbeiten von Noubiap, Troffaes, Augustin u.a.



Ausblick Intervallwahrscheinlichkeiten

Fragen

Lösung des datenbasierten Problem nicht äquivalent zu datenfreiem Problem nach Update

- Interpretation? Sollte man Updates durchführen? Passen update und Optimalitätskriterium zusammen? Kommt die Herleitung von a priori Wissen ohne Update aus?
- Quantifizierung der Unterschiede (Fehler) möglich?
- Anwendung in Robotik (s.oben)?
- Parallelisierung von datenfreiem und datenbasierten Problem nach Update sinnvoll?



Vielen Dank für die Aufmerksamkeit !

