

# Effiziente 2D-Navigation für Mobile Service Roboter

Dirk Holz

Bonn-Rhein-Sieg University of Applied Sciences  
dirk.holz@inf.fh-bonn-rhein-sieg.de

**Abstract:** Das Ziel dieser Arbeit ist das Design und die Implementierung eines vollständigen Systems zur robusten Navigation mobiler Roboter in häuslichen Umgebungen. Adressierte Probleme sind die Modellierung von Umgebungen, die Planung von Pfaden sowie die Steuerung eines mobilen Roboters. Das resultierende System wurde erfolgreich in der Robocup@Home-Liga eingesetzt.

## 1 Motivation und Problemstellung

Service Roboter, die z.B. im Haushalt assistieren, Besucher durch Museen führen oder in der Alten- und Krankenpflege eingesetzt werden können, bergen ein enormes Potential, den Alltag des Menschen erheblich zu vereinfachen [SHW04]. Um in der Welt, in der wir leben, sicher und effizient agieren zu können, benötigen autonome Roboter Fähigkeiten zur Wahrnehmung und Modellierung ihrer Umgebung, sowie, bis zu einem bestimmten Grad, über diese und in ihr stattfindenden Änderungen Schlussfolgerungen anzustellen. Zur effektiven Navigation und Planung von Aktionen benötigen mobile Roboter interne Modelle ihrer Umgebung. Die Fähigkeit diese selbstständig zu erzeugen sowie sich selbst und andere Objekte in ihnen lokalisieren zu können, wird als Voraussetzung für vollständig autonom agierende Roboter gesehen [Wan04]. Das Problem bei der Modellierung einer unbekannteren Umgebung mit mobilen Robotern ist, dass der Roboter seine Position kennen muss, um erfasste Umgebungsstrukturen in einer Karte eintragen zu können. Zur Bestimmung seiner Position benötigt er jedoch eine Karte der Umgebung. Dieses Zyklusproblem wird in der Fachliteratur als *Simultaneous Localization and Mapping* (kurz SLAM) bezeichnet.

Der Kontext dieser Arbeit ist die RoboCup@Home-Liga, ein Wettbewerb für mobile Service Roboter. Hier müssen Roboter in häuslichen Umgebungen navigieren, Objekte suchen, erkennen, greifen und transportieren. Die Kommunikation mit dem Menschen ist auf Sprache beschränkt und erfordert robuste Spracherkennung sowie verständliche Sprachsynthese [NDD<sup>+</sup>08].

Der hierfür verwendete Roboter (Abbildung 1) besteht aus einer mobilen Plattform, mit zwei differentiell angetriebenen und zwei nachlaufenden Rädern zur Stabilität, einem 2D-Laserscanner zur berührungslosen Messung von Distanzen zu umliegenden Umgebungsstrukturen, diversen Kamerasystemen und einem Roboterarm zur Manipulation von Objekten.



Abbildung 1: Roboter

Das Ziel dieser Arbeit ist es, basierend auf den Daten des 2D-Laserscanners, speicher-effiziente Umgebungsmodelle zu erstellen und diese zur Lokalisierung des Roboters, zur Planung von Pfaden und zur Steuerung des Roboters selbst zu verwenden. Hauptaugenmerk liegt hierbei auf der Echtzeitfähigkeit und Robustheit der verwendeten Verfahren, da die Roboter zusammen mit Menschen arbeiten und jede Art von Kollision vermieden werden sollte.

## 2 Konstruktion des Umgebungsmodells – SLAM

Eine Vielzahl an Lösungsansätzen für das SLAM-Problem wurde über die letzten Jahrzehnte vorgestellt. Diese unterscheiden sich vor allem in der Formulierung des Problems, des eingesetzten Lösungsverfahrens und der Repräsentierung des Umgebungsmodells. Unterschieden werden, unter anderem, diverse probabilistische Verfahren, Graph-basierte Verfahren und solche, die SLAM als Problem der Registrierung von Distanzbildern adressieren. Das hier vorgestellte Verfahren basiert auf der Registrierung von zwei- bzw. dreidimensionalen Punktmengen, wie sie von Laserscannern erfasst werden. Der iterative Algorithmus der nächsten Punkte (*Iterative Closest Point – ICP*) [BM92] ist ein Standard-Algorithmus zur Registrierung von Punktmengen. Der ICP-Algorithmus unterscheidet zwei Punktmengen: das *Modell*  $M$  und die *Szene* oder Datenmenge  $D$ . Die Registrierung von  $M$  und  $D$  wird als Optimierungsproblem betrachtet: Der ICP verfeinert iterativ eine Transformation  $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ , die  $D$  auf  $M$  abbildet, und somit die folgende Fehlerfunktion minimiert.

$$E(\mathbf{T}) = \sum_{i=1}^{|M|} \sum_{j=1}^{|D|} w_{ij} \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2 \quad \text{mit } w_{ij} = \begin{cases} 1, & m_i \text{ und } d_j \text{ korrespondieren} \\ 0, & \text{sonst} \end{cases} \quad (1)$$

Die Korrespondenzen  $w_{ij}$  werden durch die Suche nach dem nächsten Nachbarn  $\mathbf{m}_i \in M$  für jeden Punkt  $\mathbf{d}_j \in D$  ermittelt. Dieser rechenaufwendigste Schritt des Algorithmus wird beschleunigt durch die Verwendung einer approximierten Suche in einem  $kD$ -Baum [MA97]. Für die Minimierung von (1), im zwei- und dreidimensionalen Fall, existieren verschiedene Lösungen in geschlossener Form. Die gesuchte homogene Transformationsmatrix  $\mathbf{T}$  ist rigid, beinhaltet also nur Rotation ( $\mathbf{R}$ ) und Translation ( $\mathbf{t}$ ). Dass der Algorithmus zu einem (lokalen) Minimum konvergiert, wurde bereits in [BM92] bewiesen, allerdings unter der Annahme, dass für jeden Punkt  $\mathbf{d}_j$  wirklich ein korrespondierender Punkt in  $M$  existiert. Bei der Registrierung von Tiefenbildern im Kontext des SLAM-Problems ist dies nicht immer der Fall und die zu registrierenden Punktmengen überlappen nur partiell. Daraus resultierende falsche Korrespondenzen haben einen negativen Effekt auf die Registrierung, können aber durch Verwerfung bestimmter Punktpaare umgangen werden, z.B. wenn die Punkt-zu-Punkt Distanz einen Schwellwert überschreitet, oder der Modellpunkt  $\mathbf{m}_i$  mehrfach in den Punktkorrespondenzen auftritt [RL01].

Für die Registrierung mehrerer Laserscans und die Erzeugung eines Umgebungsmodells wird hier ein inkrementelles Verfahren verwendet. Der erste Laserscan  $D_1$  bildet hierbei die Basis für das Umgebungsmodell  $M_1$ , das als simple Punktwolke repräsentiert wird. Das Koordinatensystem von  $D_1$ , in dessen Ursprung sich der Roboter befindet, bildet das

Koordinatensystem des Modells. Neue Laserscans  $D_i$  werden mit  $M_{i-1}$  registriert und die enthaltenen Punkte  $d_{i,j}$  werden, nach Anwendung der gefundenen Transformation  $T_i$  zum Modell  $M_{i-1}$  hinzugefügt. Auf diese Weise wird das Umgebungsmodell inkrementell erweitert ( $M_i = M_{i-1} \cup \{R_i d_{i,j} + t_i | d_{i,j} \in D_i\}$ ). Um den Speicheraufwand des Umgebungsmodells zu minimieren und redundantes Speichern von Punkten zu vermeiden, wird eine weitere Punktpaarsuche durchgeführt. Nur solche Punkte in  $D_i$ , deren Distanz zum korrespondierenden Punkt in  $M$  einen Schwellwert  $\epsilon_d$  überschreiten, werden hinzugefügt. Zusätzlich wird die Punktdichte in den Scans reduziert, indem Punkte, deren Distanz zueinander unter  $\epsilon_d$  liegt, durch ihren Schwerpunkt ersetzt werden. Ein, auf diese Weise erzeugtes Umgebungsmodell ist in Abbildung 2 dargestellt. Es ist um eine Liste von vorgegebenen und erlernten Objekten erweitert.

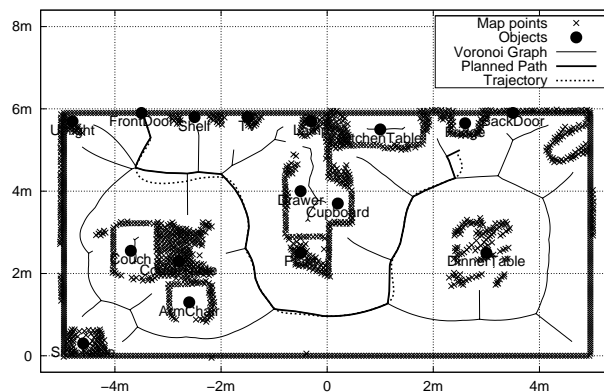


Abbildung 2: Modell einer @Home-Arena mit Objektvektor, geplantem Pfad und Trajektorie.

### 3 Pfadplanung und Steuerung

Um zu bestimmten Positionen in der Umgebung zu gelangen, muss der Roboter zunächst kollisionsfreie Pfade planen und diesen anschließend folgen. Für die Planung wird der  $A^*$  Algorithmus [HNR68] zur Berechnung des kürzesten Pfades zwischen zwei Knoten in einem Graphen verwendet. Um aus einem, wie oben beschriebenen, Punktmodell  $M$  einen Graphen für die Pfadplanung zu erzeugen, wird zunächst das Voronoi-Diagramm für die in  $M$  enthaltenen Punkte berechnet [For87]. Anschließend werden alle Kanten, die von dem Roboter nicht befahren werden können aus dem Graphen entfernt. Dazu zählen Kanten, deren Abstand zu nächsten Modellpunkten kleiner ist als die halbe breite des Roboters, und Kanten, die außerhalb der konvexen Hülle des Modells und damit in nicht modellierten Bereichen liegen. Die konvexe Hülle wird hier mit *Graham's Scan* Algorithmus [Gra72] berechnet.

Um einen geplanten Pfad abzufahren wird zunächst ein nichtlinearer Regler verwendet, um sich der Zielposition auf einer kollisionsfreien Bahn zu nähern [IC04]. Anschließend wird ein weiterer Regler [Ind99] verwendet, um Zielposition und -orientierung zu erreichen. Zusätzlich werden Verfahren zur Kollisionsvermeidung aus [HLS08] verwendet. Eine daraus resultierende Trajektorie ist in Abbildung 2 dargestellt.

## 4 Abschließende Bemerkungen

Das resultierende System wurde 2008 erfolgreich in der RoboCup@Home-Liga eingesetzt. Der Roboter belegte den 2. Platz bei den RoboCup German Open in Hannover und wurde Vize-Weltmeister bei den RoboCup Weltmeisterschaften in China. Videos, die den Einsatz der hier beschriebenen Verfahren zeigen, sind verfügbar unter:

<http://www.b-it-bots.de/media>.

Mögliche Erweiterungen des Systems mit einem 3D-Laserscanner zur räumlichen Modellierung der Umgebung und zur 3D-Kollisionsvermeidung wurden in [HLS08] vorgestellt.

## Literatur

- [BM92] P. J. Besl und N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [For87] S. J. Fortune. A Sweepline Algorithm for Voronoi Diagrams. *Algorithmica*, Seiten 153–174, 1987.
- [Gra72] R. L. Graham. An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. *Information Processing Letters*, 1(4):132–133, 1972.
- [HLS08] D. Holz, C. Lörken und H. Surmann. Continuous 3D Sensing for Navigation and SLAM in Cluttered and Dynamic Environments. In *Proceedings of the International Conference on Information Fusion (FUSION)*, 2008.
- [HNR68] P. E. Hart, N. J. Nilsson und B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [IC04] G. Indiveri und M. L. Corradini. Switching linear path following for bounded curvature car-like vehicles. In *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles (IAV04)*, 2004.
- [Ind99] G. Indiveri. Kinematic Time-invariant Control of a 2D Nonholonomic Vehicle. In *Proceedings of the 38th Conference on Decision and Control (CDC'99)*, 1999.
- [MA97] D. Mount und S. Arya. ANN: A library for approximate nearest neighbor searching. In *Proceedings of the 2nd Annual Fall Workshop on Computational Geometry*, 1997.
- [NDD<sup>+</sup>08] D. Nardi, J.-D. Dessimoz, P. F. Dominey, L. Iocchi, P. E. Rybski, J. Savage, S. Schiffer, T. Wisspeintner, T. van der Zant und A. Yazdani. RoboCup@Home – Rules and Regulations. Available online, 2008.
- [RL01] S. Rusinkiewicz und M. Levoy. Efficient Variants of the ICP Algorithm. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, 2001.
- [SHW04] R. D. Schraft, M. Hägele und K. Wegener, Hrsg. *Service - Roboter - Visionen*. Hanser Fachbuchverlag, 2004.
- [Wan04] C.-C. Wang. Simultaneous Localization, Mapping and Moving Object Tracking. PhD Thesis CMU-RI-TR-04-23, Carnegie Mellon University, Pittsburgh, PA, 2004.