

Flächendeckendes Explorieren und Navigieren in a priori unbekannter Umgebung mit low-cost Robotern

Frank Peters, Michael Kasper, Mirko Eßling, Ewald von Puttkamer

Universität Kaiserslautern • Fachbereich Informatik
AG Robotik und Prozeßrechentechnik • Postfach 3049 • D-67653 Kaiserslautern
fpeters@epost.de, {kasper, puttkam}@informatik.uni-kl.de

Zusammenfassung. Autonome mobile Serviceroboter mit flächendeckender Fahrweise können für alle Arten der Untergrundbearbeitung und -inspektion eingesetzt werden. Besondere wirtschaftliche Bedeutung wird in naher Zukunft der automatischen Bodenreinigung zukommen. In diesem Aufsatz wird die Umsetzung eines Verfahrens zum flächendeckenden Fahren in a priori unbekanntem Umgebungen auf der Basis preiswerter Roboter-Hardware beschrieben, wie es beispielsweise für Reinigungsanwendungen im Heimbereich zum Einsatz kommen kann. Parallel zur Durchführung der Reinigungsaufgabe wird dabei die Umgebung exploriert und kartiert. Das zunächst simulativ getestete Verfahren wurde zum Nachweis der Praxistauglichkeit auf zwei vergleichsweise preiswerten Roboterplattformen implementiert.

1 Einleitung

Servicerobotern als "Leitprodukten der Zukunft" [Schraft und Schmierer 98] wird für die kommenden Jahre ein großes Wachstumspotential vorausgesagt. Einen relativ hohen Anteil an diesen auf den Dienstleistungs- und Sektoren spezialisierten Robotern werden dabei verschiedenste Reinigungsmaschinen einnehmen. Neben spezialisierten Robotern zur Fassaden- oder Fahrzeugreinigung kommt hier der Bodenreinigung eine Schlüsselrolle zu.

Während es bereits seit einigen Jahren (teil-)autonome Bodenreinigungsroboter für den kommerziellen Einsatz auf Flughäfen, Bahnhöfen oder in Supermärkten gibt ([Hofner 97], [Lawitzky et al. 98], [Cybernetix 00]), existieren für den Heimbereich bisher nur Prototypen wie beispielsweise die Roboter der Firmen Electrolux, Kärcher oder Dyson ([Electrolux 97], [Kärcher 99], [Dyson 00]).

Dies liegt vor allem an den speziellen Anforderungen im Heimbereich: *Autonomous Home Cleaning Robots (AHCRs)* müssen ein hinreichend starkes Reinigungsaggregat besitzen und gleichzeitig leicht und klein genug sein, um zwischen Stuhl- und Tischbeinen reinigen zu können. Sie benötigen darüberhinaus eine robuste und fehlertolerante Steuerung, welche auch den Umgang mit dynamischen Objekten beherrscht und den Roboter zielgerichtet und effizient steuert. Die existierenden Prototypen arbeiten hingegen meist rein indeterministisch (vgl. [Doty 93]) und somit nicht zielgerichtet. Beispiele hierfür sind die *AHCRs* der Firmen Electrolux und Kärcher.

Eine effiziente und zielgerichtete Vorgehensweise beim flächendeckenden Fahren erfordert zwangsläufig eine hinreichend genaue Umweltmodellierung durch den Roboter sowie dessen Positionsbestimmung innerhalb der Umgebung. Da für die sich häu-

fig ändernden Einsatzumgebungen (Veränderung innerhalb einer Umgebung, Einsatz in verschiedenen Umgebungen) der Aufwand einer a priori Kartenerstellung nicht zweckmäßig erscheint, muss ein *AHCR* parallel zur Reinigungsaufgabe seine Umgebung explorieren und kartieren. Schließlich sollten autonome Reinigungsroboter einfach in der Handhabung und zugleich preiswert sein. Letzteres stellt aus heutiger Sicht die größte Einschränkung dar. Die technologischen Anforderungen lassen sich mit entsprechender Ausstattung bereits seit geraumer Zeit lösen, wie der Einsatz autonomer Reinigungsmaschinen im kommerziellen Umfeld zeigt. Durch die stärkeren finanziellen Restriktionen im privaten Sektor und den daraus resultierenden Beschränkungen in der Sensorik und Rechenleistung, scheiden hier viele klassische Lösungsansätze von vornherein aus.

Den Autoren sind keine publizierten und unter realen Einsatzbedingungen validierten Ansätze bekannt, die diese Anforderungen - insbesondere im Hinblick auf eine zielgerichtete Arbeitsweise - im Heimbereich berücksichtigen. Vor diesem Hintergrund wurde in [Kasper et al. 98] ein verhaltensorientierter Ansatz vorgestellt, welcher die genannten Prämissen erfüllt. Der dort noch rein simulativ getestete Algorithmus wurde nun auf zwei reale Kleinroboter übertragen und validiert. Besonderes Augenmerk wurde dabei auf den Einsatz kostengünstiger Sensorik, sowohl zur Umweltperzeption, als auch zur Positionsbestimmung des Roboters gelegt.

2 Eingesetztes Verfahren

Der implementierte Algorithmus wurde bereits in [Kasper et al. 98] vorgestellt und simulativ getestet. Im folgenden soll die Grundidee nochmals in Erinnerung gerufen, sowie die leicht erweiterte Kontrollstruktur vorgestellt werden.

2.1 Verhaltensorientierte Kontrollstruktur

Verhaltensorientierte Kontrollstrukturen werden durch ihren modularen Aufbau, die kurzen Reaktionszeiten und die hohe Fehlertoleranz sowie den relativ geringen Ressourcenbedarf den genannten Anforderungen in hervorragender Weise gerecht. Sie bieten zudem den Vorteil, dass die einzelnen Module weitgehend unabhängig voneinander implementiert und getestet, sowie bei Bedarf zu einem späteren Zeitpunkt durch verbesserte Versionen ersetzt werden können.

Den Kern der in dieser Arbeit verwendeten Kontrollstruktur bilden verdeckte und aktuatorische Verhaltensweisen, welche über globale Datenstrukturen kommunizieren (siehe Abb. 1). Die Verhalten greifen im wesentlichen auf vier globale Datenstrukturen zurück: Eine Rasterkarte zur Repräsentation der bereits befahrenen Fläche, eine probabilistische Hinderniskarte nach Elfes [Elfes 89], eine Navigationskarte zur Wegeplanung sowie auf einen *LineStack*, auf dem noch abzufahrende Streckensegmente abgelegt werden.

Die flächendeckende Fahrweise ergibt sich durch sukzessives Abarbeiten der auf dem *LineStack* befindlichen Streckensegmente, was letztendlich einer *back-tracking*-Strategie entspricht.

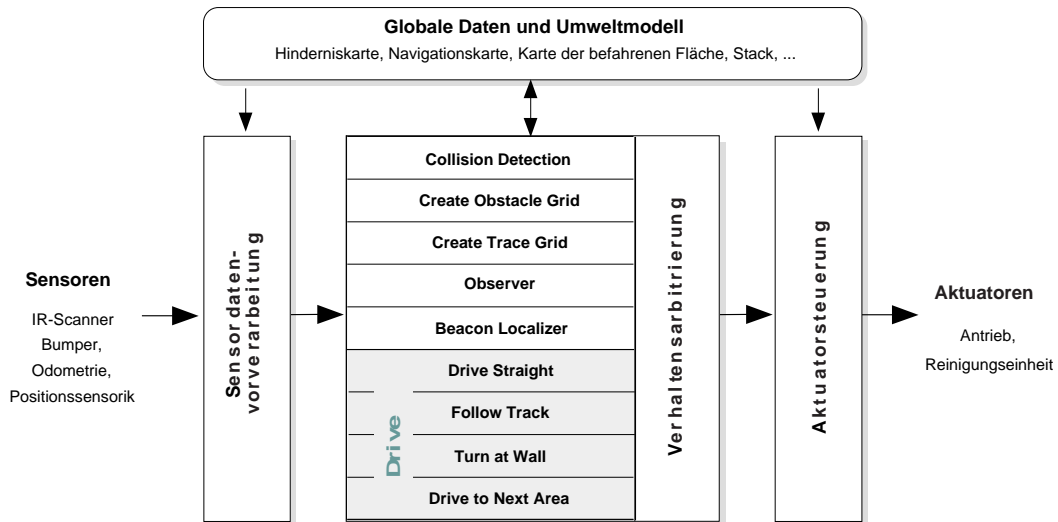


Abbildung 1: Verhaltensorientierte Kontrollstruktur

Die Grundidee des Algorithmus lässt sich wie folgt skizzieren:

1. Fahre geradeaus bzw. bündig entlang der eigenen Spur bis zum Auftreffen auf ein Hindernis. Beobachte dabei die benachbarten Bahnen und speichere freie, noch nicht befahrene Streckenabschnitte auf dem *LineStack* ab.
2. Wenn der *LineStack* noch Einträge enthält, fahre zu der entsprechenden Position und gehe zu 1.
3. Ende

2.2 Probabilistische Hinderniskarte

Um von der vorhandenen Sensorik zur Hinderniserkennung abstrahieren und Messfehler kompensieren zu können, wurde eine Hinderniskarte nach [Elfes 89] implementiert, bei der die Messwerte der einzelnen Sensoren mit Hilfe einer Wahrscheinlichkeitsverteilung in die Karte eingetragen werden. Die Hindernisdetektion basiert ausschließlich auf der aufgenommenen Wahrscheinlichkeitskarte, entkoppelt von der eigentlichen Sensordatenaufnahme.

Aufgrund der eingeschränkten Speicherkapazität der eingesetzten Roboter-Plattformen wird nur ein eingeschränkter Bereich um den jeweiligen *AHCR* herum in einer lokalen, hochauflösenden Rasterkarte repräsentiert. Größe und Auflösung der Karte können an die Abmessungen des verwendeten Roboters angepasst werden. Typische Werte sind $2\text{ m} \times 2\text{ m}$ bei einer Auflösung von 1 cm .

2.3 Karte der befahrenen Fläche

Um ein bündiges Fahren entlang bisheriger Spuren zu realisieren und eine unnötige, mehrfache Flächenabdeckung zu vermeiden, wird die bereits befahrene Fläche im Zuge der Exploration in einer globalen *Spurkarte* erfasst. Diese speichert das gesamte, potentiell durch den *AHCR* befahrbare Gebiet und muss entsprechend dimensioniert sein. Bei einer typischen Auflösung von 1 cm und einer Arealgröße von $10\text{ m} \times 10\text{ m}$

liegt der erforderliche Speicherbedarf bei ca. 122 KByte. Durch den Einsatz üblicher Kompressionstechniken (Quad-Tree, Lauflängenkodierung, usw.) kann der Speicherbedarf deutlich reduziert, bzw. die Arealgröße erhöht werden. Die Eintragung der befahrenen Fläche in die Karte erfolgt durch das Einsetzen einer Maske mit den Roboterabmessungen.

2.4 Navigationskarte

Die Punkt-zu-Punkt Navigation über größere Distanzen hinweg erfolgt ausschließlich auf bereits befahrenem Gebiet und wird mit einer vereinfachten Potentialfeldmethode realisiert. Hierbei genügt es, die anziehende Komponente des Zieles, nicht aber die abstoßenden Komponenten der Hindernisse zu berücksichtigen. Die eigens zur Navigation aufgebaute Navigationskarte (ebenfalls eine Rasterkarte, jedoch mit gröberer Auflösung als die *Spurkarte*) enthält Informationen über die befahrenen Areale sowie implizit über die Hindernissituation. Mit Hilfe des *Distance Transform* Verfahrens wird die Karte ausgehend vom anzusteuern Ziel mit ansteigenden Potentialwerten geflutet. Von jedem hindernisfreien Feld der Rasterkarte kann danach ein minimaler Pfad zum Ziel mit sinkenden Potentialwerten ermittelt und befahren werden. Dynamische Hindernisse, die sich auf bereits befahrenem Gebiet befinden, werden während der Punkt-zu-Punkt Navigation als temporäre Hindernisse in der Navigationskarte vermerkt. Versperren solche Hindernisse den Zielpfad, so erfolgt eine Neuplanung unter Berücksichtigung der veränderten Situation.

2.5 Verdeckte Verhaltensweisen

Die folgenden verdeckten Verhaltensweisen realisieren im Hintergrund ablaufende Funktionen, wie z.B. die Kollisionserkennung oder den Aufbau der Hinderniskarte.

Das Verhalten *Collision Detection* erkennt und bereinigt Kollisionen, die üblicherweise durch Bumper detektiert werden. Das kollisionsauslösende Hindernis wird in die Hinderniskarte eingetragen. *Create Obstacle Grid* trägt die Messwerte der IR-Distanzsensoren mit Hilfe eines probabilistischen Sensormodells in die lokale Hinderniskarte ein. Die vom Roboter befahrene Fläche wird von *Create Trace Grid* in die globale *Spurkarte* eingetragen.

Ein zentrales Verhalten ist der *Observer*. Er beobachtet die seitliche Umgebung des *AHCR* auf angrenzende, noch zu befahrende Flächen und legt diese in Form von Streckensegmenten auf dem *LineStack* ab. Der *Beacon Localizer* ermittelt die absolute Position des Roboters mit Hilfe einer aktiven Landmarke (siehe Abschnitt 3.3).

2.6 Aktuatorische Verhaltensweisen

Die aktuatorischen Verhaltensweisen implementieren das eigentliche, mäanderförmige Befahren einer zusammenhängenden Freifläche.

Drive Straight realisiert ein schlichtes Geradeausfahren. Es wird in Situationen eingesetzt, in denen nicht einer zuvor beobachteten Bahn gefolgt werden kann. Das Verhalten *Follow Track* folgt bündig bzw. mit vorgegebener Überlappung einer bereits befahrenen Spur. *Turn at Wall* folgt der Kontur eines Hindernisses (z.B. einer Wand), sofern dieses den Weg blockiert und eine noch nicht befahrene Nachbarbahn existiert.

Die Punkt-zu-Punkt Navigation des *AHCR* wird durch *Drive to Next Area* implementiert. Dieses Verhalten navigiert den Roboter auf Basis der Navigationskarte zu dem Anfang einer zuvor beobachteten, jedoch noch nicht befahrenen Bahn, sofern das aktuelle Teilgebiet komplett befahren wurde und keine unbefahrenen Nachbarbahnen existieren.

3 Implementierung des Verfahrens auf realen Kleinrobotern

3.1 Roboter-Plattformen

Durch die Implementierung auf zwei Roboter-Plattformen mit unterschiedlichen Antriebs- und Sensorkonfigurationen konnte der vorgestellte Ansatz real validiert werden. Der kleinere der beiden Roboter (*Doozer*, Grundfläche 10×13 cm) verfügt über fünf starr montierte IR-Entfernungssensoren (Sharp GP2D02, aktives Triangulationsprinzip, nutzbare Reichweite 40 cm) und einen Differentialantrieb.

Der größere Roboter (*Xenia*, 32 cm Durchmesser) ist mit vier rotierenden IR-Sensoren des gleichen Typs und einem Synchro-Drive Antrieb ausgestattet und besitzt zudem ein Empfangsmodul zur Positionskorrektur mittels einer aktiven Landmarke (siehe Abschnitt 3.3). Abbildung 2 a) zeigt die in Zusammenarbeit mit Kollegen der University of Western Australia in Perth entwickelten mobilen Kleinroboter [Bräunl 99].

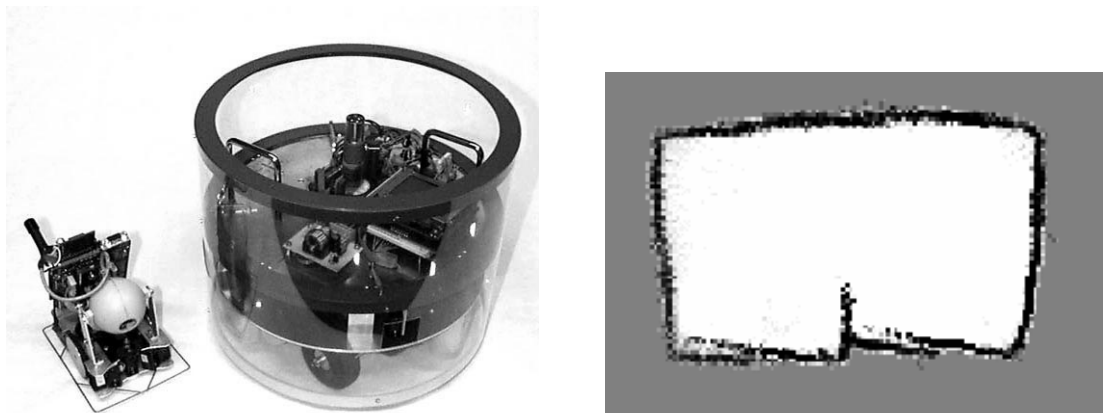


Abbildung 2: a) Eingesetzte Roboterplattformen *Doozer* und *Xenia* (von links)
b) Mit *Doozer* aufgenommene Hinderniskarte (ca. $2\text{m} \times 1\text{m}$)

3.2 Hardwareabstraktion

Da der Kernalgorithmus einerseits nicht direkt auf einzelne Sensoren, sondern ausschließlich auf die lokale Hinderniskarte zugreift, und andererseits das Antriebssystem durch eine (v, ω) -Schnittstelle kontrolliert, konnte weitgehend von der verwendeten Roboterhardware abstrahiert werden. Trotz unterschiedlicher Antriebs- und Sensorkonfigurationen der beiden Kleinroboter findet so der gleiche Programmcode Verwendung.

Abbildung 2 b) zeigt eine typische, mit dem Roboter *Doozer* aufgenommene probabilistische Hinderniskarte, die auf Basis des Sensormodells und der Odometriedaten, ohne externe Positionskorrektur, autonom erstellt wurde.

3.3 Positionsbestimmung

Von zentraler Bedeutung für das vorgestellte Verfahren ist die exakte Kenntnis der aktuellen Position des *AHCR*. Nur so können Bahnen mit optimaler Überlappung befahren und andere, die für eine spätere Bearbeitung auf dem *LineStack* abgelegt wurden, zuverlässig angefahren werden.

In den letzten Jahren haben Arbeiten wie [Weiß et al. 94] oder [Thrun et al. 00] Möglichkeiten aufgezeigt, mit denen z.B. durch Korrelation von Sensordaten die Roboterposition in der Ebene bestimmt werden kann. Diese Verfahren erfordern jedoch sowohl genaue und weitreichende Entfernungssensoren, als auch eine hohe Rechenleistung. Beide Voraussetzungen wird ein *AHCR* in absehbarer Zeit nicht erfüllen können.

Das Verhalten *Beacon Localizer* verwendet daher ein neuartiges und kostengünstiges Positionsensormodell, welches eine ohnehin vorhandene Basis- bzw. Ladestation zu einer aktiven Landmarke erweitert. Die Landmarke enthält zwei parallelliegende Laser auf einer mit bekannter Drehzahl rotierenden Scheibe und ermöglicht bei Sichtkontakt eine Positionsbestimmung in Polarkoordinaten für einen oder mehrere Roboter (Abbildung 3). Als Empfangselement dient dem *AHCR* ein kompakter, stabförmiger Photodetektor mit 360° Empfangscharakteristik, der aufgrund seiner Länge Höhenunterschiede bis zu 15 cm ausgleichen kann. Die Entfernung eines Roboters von der Basisstation wird aus der zeitlichen Differenz zwischen dem Auftreffen der beiden Laserstrahlen auf dem Photodetektor ermittelt. Zusätzlich wird die Winkelkoordinate des Roboters zu einer (gedachten) 0°-Referenzachse der Landmarke bestimmt. Dies geschieht durch einmalige Synchronisation einer Empfängeruhr auf die Rotationsgeschwindigkeit der Lasereinheit und eine anschließende, kontinuierliche Zeitmessung unter Inkaufnahme einer geringen Winkeldrift. Aufgrund der typischerweise kurzen Einsatzdauer eines *AHCR* hat sich diese Vorgehensweise als ausreichend erwiesen.

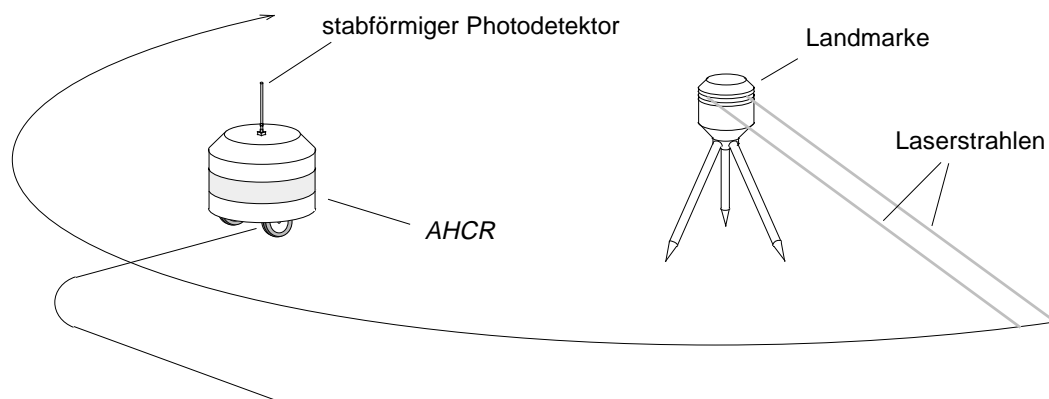


Abbildung 3: Positionsbestimmung mit Hilfe einer aktiven Landmarke

Das Verhalten *Beacon Localizer* ermittelt kontinuierlich die Position des *AHCR* relativ zur Landmarke (sofern diese sichtbar ist) und korrigiert die x- und y-Komponenten der internen Roboterodometrie. Zur Korrektur der Orientierung werden Einzelmessungen mit den dazwischenliegenden Bewegungsdaten des Roboters herangezogen. Der realisierte Prototyp erlaubt Positionsbestimmungen mit einer Genauigkeit besser als 1 cm in einem Radius von 10 m um die Landmarke, bei einer Wiederholrate von 2 Hz. Durch Mittelung mehrerer Messungen kann diese Genauigkeit auch bis zur maximalen Reichweite von 50 m aufrecht erhalten werden.

4 Experimentelle Ergebnisse

Durch das Zusammenspiel der einzelnen Verhaltensweisen wird eine lokal-optimale Aktionssteuerung realisiert. Ein übergeordneter Planer, der ein global-optimales Verhalten garantieren würde, kann aufgrund der erst während der Task-Ausführung gewonnenen Umweltinformationen prinzipiell nicht zum Einsatz kommen. Die mit den Versuchsplattformen erzielten Ergebnisse sind in Hinblick auf den Abdeckungsgrad der Gesamtfläche und den Prozentsatz der genau einfach befahrenen Fläche direkt mit den in der Simulation ermittelten Werten aus [Kasper et al. 98] vergleichbar. Beide Werte liegen bei einfach strukturierten Umgebungen zwischen 75 und 90 %.

Abbildung 4 a) zeigt ein typisches Ergebnis einer Simulationsfahrt (allerdings ohne das initiale *Wall-Following*), während Abbildung 4 b) das Ergebnis einer realen Fahrt des Roboters *Xenia* in einem Areal von 2,3 m × 3,3 m wiedergibt. Zum Nachweis der tatsächlich befahrenen Fläche liegt Abbildung 4 b) nicht die interne Rasterkarte des *AHCR*, sondern eine per Deckenkamera aufgenommene Bildsequenz zu Grunde.

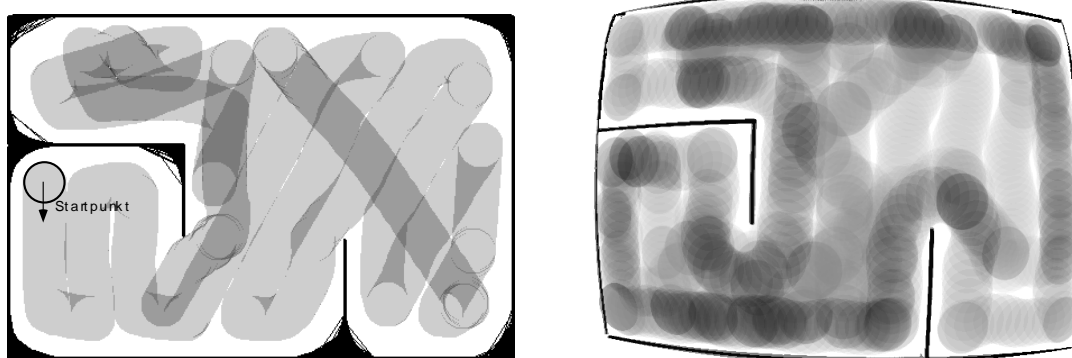


Abbildung 4: a) Umweltmodell und Flächenabdeckung in der Simulation (links)
b) Flächenabdeckung im realen Experiment (rechts)

5 Zusammenfassung und Ausblick

Es wurde die Implementierung eines Algorithmus zum flächendeckenden Fahren in a priori unbekannter Umgebung auf zwei Roboterplattformen vorgestellt. Parallel zu der flächendeckenden Fahrt wird dabei die Umgebung exploriert und kartiert.

Die Versuchsergebnisse in einer einfach strukturierten Laborumgebung belegen die prinzipielle Übertragbarkeit des zuvor simulativ untersuchten Verfahrens auf reale Maschinen mit kostengünstigen Sensoren und moderater Rechenleistung. Nach einer Optimierung des Programmcodes hinsichtlich der Speicherplatz- und Rechenzeitanforderungen werden in Kürze Tests in Umgebungen größer 10 m × 10 m durchgeführt. Zusätzliche Hindernissensoren und eine weiter verbesserte Auswertung der bisherigen Sensoren, werden den Einsatz in typischen "Heimszenarien" außerhalb des Labors ermöglichen.

Literatur

- [Bräunl 99] Bräunl, T.: *EyeBot: A Family of Autonomous Mobile Robots*; Proceedings of the 6th intl. Conf. on Neural Information Processing, (ICONIP '99), Perth, Nov. 1999,
- [Cybernetix 00] Produktionformation der Firma Cybernetix, Marseille, Frankreich zu der autonomen Reinigungsmaschine Acromatic 750; <http://www.cybernetix.fr>; Januar 2000
- [Doty 93] Doty, K. L.; Harrison, R. R.: *Sweep Strategies For a Sensory-Driven, Behavior-Based Vacuum Cleaning Agent*, AAAI Fall Symposium Series "Instantiating Real-World Agents", Raleigh, October 1993
- [Dyson 00] Produktinformationen der Firma Dyson, Malmesbury, Großbritannien, zu dem autonomen Staubsauger DC06; <http://www.dyson.com>; Januar 2000
- [Electrolux 97] Pressemitteilung der Firma Electrolux zur Vorstellung des Prototyps eines autonomen Bodenreinigungsroboters; <http://www3.electrolux.se/robot/>; Dezember 1997; siehe auch <http://www.eureka.com/whatsnew/robotvacupdate.htm>
- [Elfes 89] Elfes, A.: *Using Occupancy Grids for Mobile Robot Perception and Navigation*; IEEE Computer 6/89
- [Hofner 97] Hofner, C.: *Automatische Kursplanung und Fahrzeugführung für mobile Roboter bei flächendeckenden Bearbeitungsaufgaben*; Dissertation am Lehrstuhl für Steuerungs- und Regelungstechnik, TU München, 1997
- [Kärcher 99] Pressemitteilung der Firma Kärcher, Winnenden, zur Vorstellung des Prototypen eines Saugroboters für Privathaushalte; <http://www.karcher.com/deutsch/press.shtml#roboter>; Februar 1999
- [Kasper et al. 98] Kasper, M.; Müller, D.; von Puttkamer, E.: *Ein verhaltensorientierter Ansatz zum flächendeckenden Fahren in a priori unbekannter Umgebung*; 14. Fachgespräch Autonome Mobile Systeme (AMS '98); Karlsruhe, Dezember 1998
- [Lawitzky et al. 98] Endres, H.; Feiten, W., Lawitzky, G.: *Field Test of a Navigation System: Autonomous Cleaning in Supermarkets*; Proceedings of the 1998 IEEE International Conference on Robotics & Automation (ICRA); Leuven, Belgium, May 1998
- [Schraft und Schmierer 98] Schraft, R. D.; Schmierer, G.: *Serviceroboter: Produkte, Szenarien, Visionen*; Springer, 1998
- [Thrun et al. 00] Thrun, S.; Fox, D.; Burgard, W.; Dellaert F.: *Robust monte carlo localization for mobile robots*; Technical Report CMU-CS-00-125, Carnegie Mellon University, Pittsburgh, 2000
- [Weiß et al. 94] Weiß, G.; Wetzler, C.; von Puttkamer, E.: *Positions- und Orientierungsbestimmung von bewegten Systemen in Gebäuden durch Korrelation von Laserradaraten*; Autonome Mobile Systeme, 10. Fachgespräch Stuttgart, 13.-14. Oktober 1994