

Wenn Roboter nach dem Weg fragen

Navigation durch natürliche Mensch-Maschine Interaktion

Navigation by natural human-Robot Interaction

Quirin Mühlbauer, Tingting Xu, Andrea Bauer, Klaas Klasing, Georgios Lidoris, Florian Rohrmüller, Stefan Sosnowski, Kolja Kühnlenz, Dirk Wollherr, Martin Buss, Technische Universität München

Zusammenfassung Im Rahmen des „Autonomous City Explorer (ACE)“ Projektes wurde ein Roboter entwickelt, der sich ohne die Verwendung von GPS oder vorherigem Kartenwissen in unbekanntem Umgebungen zurechtfindet, indem er ähnlich wie ein Tourist Passanten nach dem Weg fragt. Während eines Experimentes legte ACE erfolgreich die 1,5 km lange Strecke vom Campus der Technische Universität München zum Marienplatz im Zentrum von München zurück. Dieses Pilotprojekt ist ein erster Schritt in Richtung Wissensbildung durch Mensch-Roboter-Interaktion zukünftiger humanoider Roboter.

▶▶▶ Summary The “Autonomous City Explorer (ACE)” project provides a robot capable of navigating in unknown urban environments without the use of GPS data or prior map knowledge. The ACE robot finds its way by interacting with pedestrians in a natural and intuitive way. In a recent experiment the robot managed to successfully travel a distance of 1.5 km from the campus of the Technische Universität München to Marienplatz, the central square of Munich. This pilot project is a first step towards knowledge acquisition through interaction between humans and future humanoid robots.

Schlagwörter Mensch-Roboter Interaktion, Wissensdarstellung, soziale Robotik, angewandte Robotik, autonome Navigation
▶▶▶ Keywords Human-robot interaction, knowledge representation, social robotics, field robotics, autonomous outdoor navigation

1 Einleitung

Durch die technische Entwicklung der letzten Jahre kommt Robotern in immer mehr Bereichen unseres täglichen Lebens eine immer größere Bedeutung zu. Bereits seit vielen Jahren wird daran gearbeitet humanoide Roboter zu entwickeln, die den Menschen auf intelligente und vielseitige Weise unterstützen [1]. Um diese Vision umzusetzen sind kognitive Fähigkeiten nötig, insbesondere bei der Wahrnehmung der Umgebung, der Entscheidungsfindung und der Interaktion mit Menschen. Bei autonomen Fahrzeugen, die sich in städtischen Umgebungen fortbewegen, hat die Forschung in den letzten Jahren große Fortschritte erzielt [2]. Auch auf

dem Gebiet der Mensch-Roboter-Interaktion wurden beeindruckende Ergebnisse erzielt, beispielsweise existieren einige autonome Roboter, die Besuchern von Museen [3] oder Einkaufszentren [4] beratend zur Seite stehen. Zum Navigieren setzen all diese Systeme jedoch entweder einen GPS-Empfänger und vordefinierte globale Wegpunkte oder eine strukturierte und vollständig bekannte Umgebung voraus. In komplexen, dynamischen Umgebungen jedoch, werden Robotern, die Menschen in ihren alltäglichen Aufgaben unterstützen sollen, nicht immer alle benötigten Informationen und kein umfassendes Wissen über ihre Umgebung zur Verfügung stehen. Somit ist es ein zentraler Aspekt autonomen Verhaltens, durch In-

teraktion mit Menschen an dieses fehlende Wissen zu gelangen. Bisher gibt es nur wenige Roboter, die in der Lage sind, benötigte Informationen von Menschen zu erfragen [5; 6].

In diesem Artikel wird das *Autonomous City Explorer (ACE)* Projekt vorgestellt [7; 8], wobei im Besonderen auf Algorithmen für die Bildverarbeitung eingegangen werden soll. Ziel des Projektes war es, einen Roboter zu entwickeln, der sich in unbekanntem städtischen Umgebungen zurechtfindet, indem er mit Fußgängern interagiert. Dem Roboter kann dabei ein bestimmtes Ziel in der Stadt vorgegeben werden, das er dann ohne GPS oder vorheriges Kartenwissen findet, indem er Passanten nach dem Weg fragt. Hierfür ist eine kognitive Architektur nötig, die Module für Navigation, Bildverarbeitung und Interaktion miteinander verknüpft. Die bei ACE eingesetzte Architektur verwendet unter anderem einen dualen Pfadplaner, eine Strategie zur Erkennung der Umgebung, Gestenerkennung als Eingabemodalität und eine semantische Darstellung des gewonnenen Wissens über den Weg. Da die Passanten zufällig ausgewählt werden und kein Vorwissen über Roboter zu erwarten ist, muss die Interaktion auf möglichst natürliche und intuitive Weise erfolgen.

Der entwickelte Roboter hat im Rahmen eines umfangreichen Experimentes erfolgreich den Weg vom Campus der Technischen Universität München bis zum Marienplatz, im Zentrum der Stadt München, zurückgelegt. Für die Strecke von 1,5 km benötigte der Roboter 5 Stunden, wobei ACE mehrere Straßen überquerte, dem Gehsteig folgte und mit 38 Passanten interagierte. Bild 1 zeigt einige Stationen auf dem Weg zum Marienplatz.

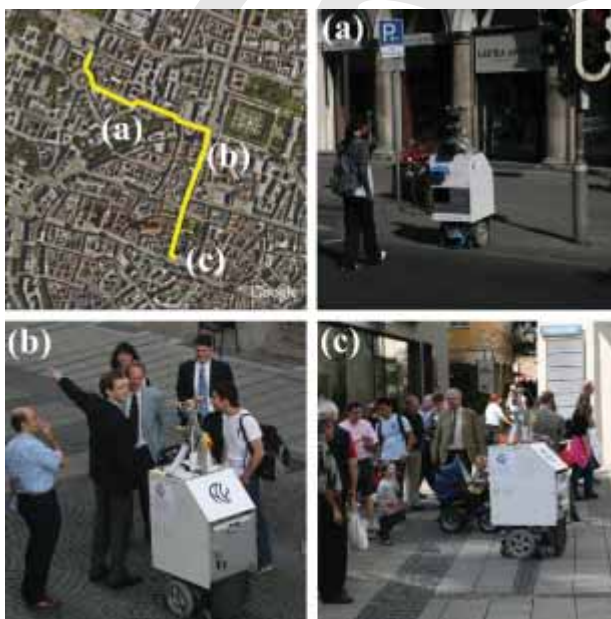


Bild 1 Weg vom Campus der TUM zum Marienplatz: (a) Überqueren einer Straße, (b) Interaktion in der Fußgängerzone und (c) ACE erreicht sein Ziel, den Marienplatz.

Dieser Artikel ist wie folgt strukturiert: in Kapitel 2 wird der Aufbau von ACE beschrieben, die einzelnen Subsysteme für die Navigation, Bildverarbeitung und Interaktion werden in Kapitel 3 vorgestellt, gefolgt von Ergebnissen des Experimentes in Kapitel 4. Kapitel 5 schließt mit einer Zusammenfassung und einem Ausblick.

2 Systemarchitektur

ACE besteht aus einer von BlueBotics SA entwickelten Plattform mit Differentialgetriebe, zwei Laserscanner für die Navigation und das Erkennen des Gehsteigs, einem Lautsprecher, einem Touchscreen, einem animierten Mund und einem multifokalen Kamerakopf. Bild 2 zeigt den Roboter und seine Hauptkomponenten. Der Roboter ist 78 cm lang, 56 cm breit, 178 cm hoch und wiegt dabei ca. 160 kg. Die Maximalgeschwindigkeit beträgt 1,4 m/s, die maximale Beschleunigung ist $1,35 \text{ m/s}^2$. Da der Roboter keine großen Stufen und keine großen Steigungen überwinden kann, müssen Gehsteigkanten und unebene Untergründe vermieden werden. Der Kamerakopf besteht aus einer Bumblebee X3 Kamera, die auf einer Schwenk-Neige-Plattform montiert ist und mit einer Brennweite von 3,8 mm für die Erkennung von Menschen verwendet wird, sowie einer fest montierten Bumblebee X2 Kamera, die mit einer Brennweite von 2 mm für die Erkennung von Gesten und Verkehrsschildern verwendet wird.

Die Software läuft auf zwei Quad-Core Linux-PCs, von denen einer für Navigation und Interaktion und einer für die Bildverarbeitung verwendet wird. Der Bildverarbeitungs-PC ist mit zwei CUDA-fähigen Grafikkarten (Geforce 8800) für aufwändige Berechnungen ausgerüstet. Durch neuartige Lithium-Polymer Akkus können diese PCs bis zu acht Stunden lang betrieben werden. Ein dritter PowerPC läuft unabhängig davon und übernimmt die low-level-Regelung der Plattform.

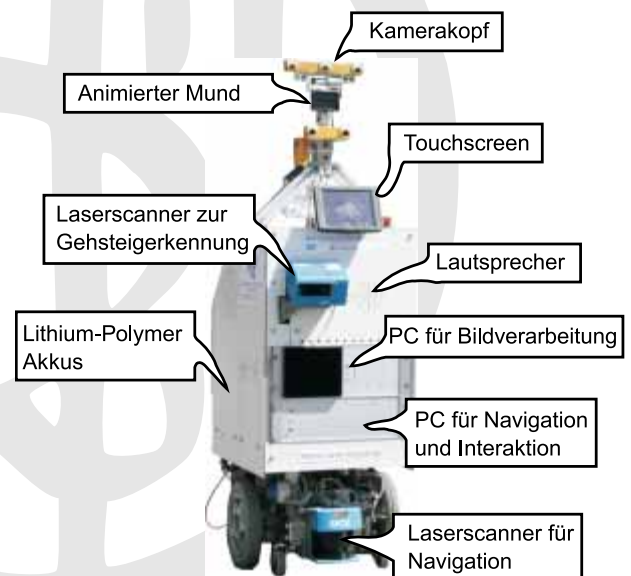


Bild 2 Der ACE-Roboter mit seinen Hauptkomponenten.

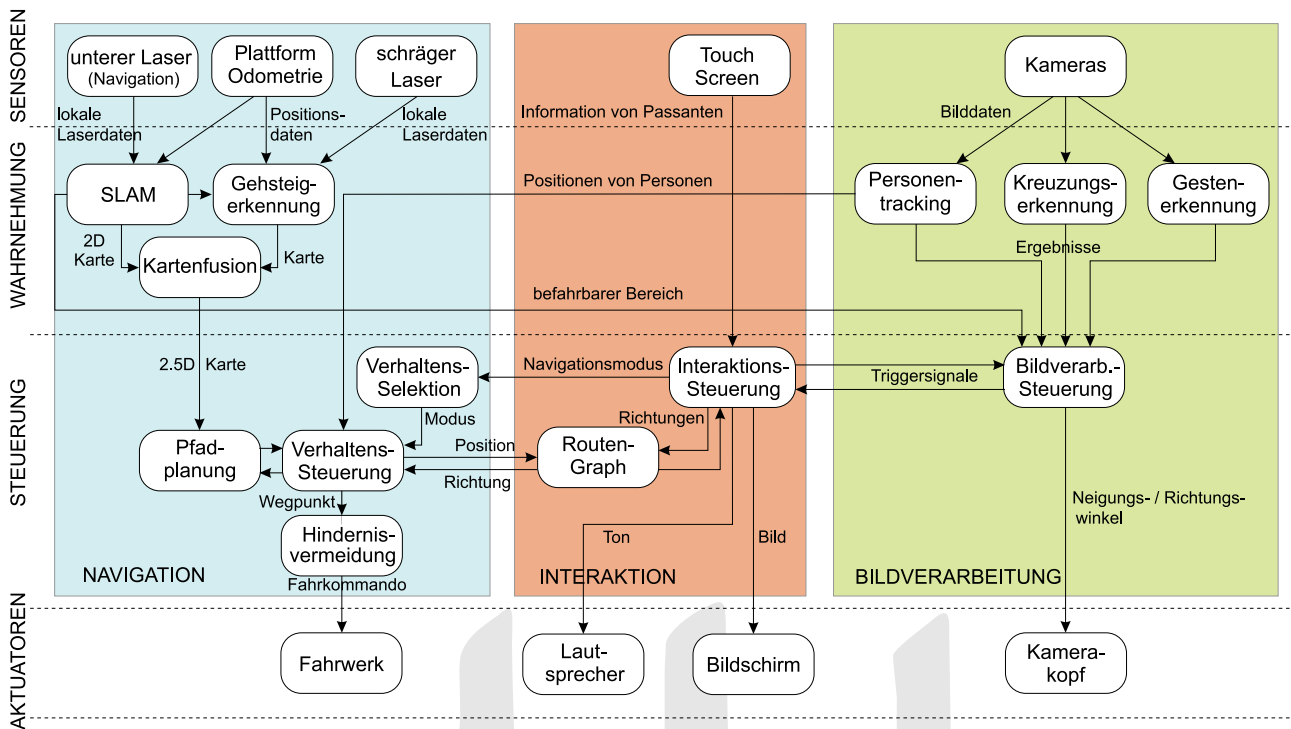


Bild 3 Die Software Architektur von ACE. Module sind eingerahmt dargestellt, während Subsysteme als Regionen farbig hervorgehoben sind. Die verschiedenen Schichten sind durch gestrichelte Linien horizontal getrennt.

Die Software besteht aus drei Subsystemen, je eines für Navigation, Bildverarbeitung und Interaktion, die sich in verschiedenen Schichten unterteilen lassen. Bild 3 zeigt die Aufteilung in Module, Subsysteme und Schichten sowie die Zusammenhänge zwischen den einzelnen Teilen.

3 Beschreibung der Subsysteme

In diesem Abschnitt werden die einzelnen Subsysteme, angefangen bei dem Navigationssystem und dem Bildverarbeitungssystem, beschrieben. Diese bilden die Grundlage für das abschließend beschriebene Interaktionssystem.

3.1 Navigation

Damit der ACE-Roboter ohne GPS und vorheriges Kartenwissen sicher zu einem vorgegebenen Ziel navigieren kann, muss er sich selbst lokalisieren, eine Repräsentation seiner Umgebung aufbauen und einen fahrbaren Pfad durch diese planen können. Dieser Abschnitt beschreibt die verwendeten Ansätze für die simultane Lokalisierung und Kartenerstellung (SLAM, *Simultaneous Localization and Mapping*) für die Generierung einer 2.5D Umgebungsrepräsentation und die Pfadplanung. Weiterhin werden die verschiedenen Navigationsverhaltensmuster des Roboters beschrieben.

Eine Grundvoraussetzung, damit autonome Systeme in einer nicht vorhersagbaren dynamischen Umgebung agieren können, ist zunächst die Fähigkeit aus verrauschten Sensordaten ein Modell der Umgebung zu erstellen. Solch ein Modell kann mittels SLAM bestimmt werden. SLAM zählt zu den grundlegenden Problemen der

Robotik und wurde in den letzten Jahren intensiv erforscht. Der erfolgreichste Ansatz [9] basiert auf dem Erweiterten Kalman Filter (EKF). Dieser Ansatz ist sehr effektiv da seine resultierende Schätzung voll mit den verwendeten Landmarken-Karten und den Roboter-Posen (Position und Orientierung) korreliert. Nachteile dieses Ansatzes sind allerdings, dass das Bewegungsmodell und das Sensorrauschen als gaußverteilt angenommen werden und dass er weiterhin schlecht mit der Kartengröße skaliert. Letzteres ist auf die Beibehaltung der vollständigen Korrelationsmatrix zurückzuführen. Ein weiterer sehr bekannter Ansatz [10] korrigiert die Schätzung der Pose mittels der inversen Kovarianzmatrix. Diese ist spärlich besetzt was die Prädiktion und Korrektur in konstanter Zeit ermöglicht. Weiterhin wurden Partikelfilter als effiziente Methode zur Zustandsschätzung [11] eingesetzt. In [12] wurden sie für das SLAM Problem auf Basis von Landmarken-Karten erweitert. In [13] wird eine Technik zur Verbesserung von raster-basiertem Rao-Blackwellized SLAM vorgestellt. Dabei wird ein ähnlicher Ansatz verwendet, mit dem Unterschied dass der Scan-Abgleich durch den Filter nicht für jedes Partikel, sondern lediglich vor der Benutzung neuer Odometriedaten durchgeführt wird. Dadurch lässt sich eine bessere Performance erzielen.

Partikelfilter ermöglichen die Approximation beliebiger Wahrscheinlichkeitsverteilungen. Dies macht sie robuster gegen unvorhersehbare Ereignisse wie beispielsweise kleine Kollisionen, welche sich gerade in Außenbereichen des Öfteren ereignen und nicht modelliert werden können. Der einzige Nachteil von

Partikelfiltern ist der höhere Rechenaufwand, da mehr Partikel benötigt werden um die Genauigkeit der Approximation zu verbessern. Falls jedoch eine geeignete initiale Verteilung gewählt wird, kann die Approximation auch bei kleiner Partikelanzahl ausreichend genau gehalten werden. Weiterhin stützen sich raster-basierte SLAM Verfahren nicht auf festgelegte Merkmalsextraktoren, welche auf der Annahme beruhen, dass die Umgebung eine bestimmte Struktur aufweist. Vor allem in unstrukturierten Umgebungen liefern daher raster-basierte Methoden ein wesentlich robusteres und genaueres Kartenwissen.

Das SLAM Modul von *ACE* verwendet ausschließlich Informationen von den Radencodern sowie Entfernungsmessdaten eines Laserscanners, um eine Schätzung der Roboterposition sowie eine Karte zu extrahieren. Hierfür wird ein Belegungs raster (Occupancy Grid [14]) zusammen mit Partikelfiltern [15] verwendet. Somit werden die oben erwähnten Vorteile genutzt, dass beliebige Fehlermodelle approximiert werden können und der Ansatz nicht von vordefinierten Umgebungsmerkmalen abhängig ist. Da sich *ACE* häufig in unstrukturiertem Gelände bewegt, verhilft dies zu einem robusten und genauen Ergebnis.

Das SLAM Modul verwendet die Daten des unteren Laserscanners, welcher in einer Ebene parallel zum Boden misst. Um allerdings negative Hindernisse, wie beispielsweise Gehsteige oder Stufen, erkennen zu können, ist der *ACE*-Roboter mit einem zweiten schräg nach unten gerichteten Laserscanner ausgestattet. Dieser liefert 3D-Messpunkte, welche als befahrbar klassifiziert werden, wenn entweder die Höhe des Punktes innerhalb von zwei statischen Schwellwerten liegt oder die Distanz zum nächstliegenden vorangegangenen Messwert innerhalb eines gewissen Bereiches liegt und der vorangegangene Messwert befahrbar war. Auf diese Weise werden stufenlose Gefälle als befahrbar erkannt sowie Messrauschen bewältigt. Anhand der klassifizierten Punkte wird ein zweites Raster erstellt, welches in das Koordinatensystem des ersten Rasters transformiert und mit diesem zu einem 2.5D-Raster fusioniert wird. Hierbei wird immer die höchste Belegungswahrscheinlichkeit der Zelle verwendet. Eine ausführlichere Beschreibung der Befahrbarkeitserkennung ist in [16] zu finden. Die 2.5D Karte wird schließlich zur Pfadplanung des Roboters genutzt.

Zur Pfadplanung wird ein dualer Ansatz verwendet, um zunächst einen Sichtbarkeitsgraphen zu erstellen. Hierbei werden Begrenzungsboxen um jedes Hindernis gelegt, und diejenigen Eckpunkte der Boxen, die im freien Raum liegen, dem Graphen hinzugefügt. Weiterhin werden Knoten von einem erweiterten Voronoi-Graphen extrahiert und ebenfalls hinzugefügt. Eine A*-Suche liefert letztlich den kürzesten Pfad auf dem resultierenden Sichtbarkeitsgraphen. Der duale Ansatz ermöglicht dem Roboter sowohl effizientes Navigieren in weitläufigen Umgebungen als auch sicheres Navigieren mit größtmöglichem Hindernisabstand an Engstellen.

Die oben beschriebenen Methoden stellen die Grundfertigkeiten eines Roboters dar, um von A nach B zu kommen. Um allerdings die verschiedensten Situationen bewältigen zu können, mit denen ein Roboter in einer städtischen Umgebung konfrontiert wird, muss er sein Navigationsverhalten adäquat anpassen. Der *ACE*-Roboter verfügt insgesamt über fünf Navigationsverhalten: Das Erkunden seiner Umgebung, das Folgen einer Person, das Fahren in eine bestimmte Richtung, das Anfahren eines bestimmten Zielpunktes und – beispielsweise während einer Interaktion – das Verharren im Ruhezustand. Das für eine Situation geeignetste Verhalten wird anhand eines Zustandsautomaten ausgewählt. Weiterhin wird während der anschließenden Ausführung permanent überwacht ob gewisse verhaltensabhängige Bedingungen erfüllt sind, oder ob aufgrund von zu großen Abweichungen ein Verhaltenswechsel erforderlich ist.

Eine wesentlich detailliertere Beschreibung der Navigationsmodule ist in [17] zu finden.

3.2 Bildverarbeitung

Das Bildverarbeitungs-Subsystem besteht aus drei verschiedenen Modulen. Das Modul *Personentracking* ist aktiv, sobald der Roboter eine Interaktion initiieren will oder einem Menschen über eine Kreuzung folgt. Während der Interaktion ist das Modul *Gestenerkennung* aktiv, um die Körperhaltung des Menschen zu interpretieren. Während des Fahrens verwendet *ACE* das Modul *Kreuzungserkennung*. Die Übergänge zwischen den Modulen werden dabei jeweils von dem Subsystem *Interaktion* getriggert.

Menschen finden und folgen

Damit eine Interaktion mit Passanten initiiert werden kann muss *ACE* die Fähigkeit haben, Menschen zu erkennen. Die meisten bisherigen Algorithmen suchen bestimmte Merkmale in Bildern und versuchen diese zu klassifizieren [18]. Diese Algorithmen sind allerdings rechenaufwendig und auf hohe Auflösungen angewiesen, weswegen sie für Anwendungen im Freien nicht geeignet sind. Algorithmen, die nach menschlicher Hautfarbe suchen sind zwar echtzeitfähig, aber nicht robust genug. Deswegen werden im *ACE*-Projekt verschiedene Algorithmen nach [19; 20] kombiniert.

Finden von Menschen

Bild 4 veranschaulicht das Modell für die Erkennung von Menschen. Aus zwei nacheinander folgenden Eingangsbildern werden, unter der Annahme dass Menschen normalerweise in Bewegung bleiben, Hautfarbe im Gesicht oder Körper besitzen und näherungsweise vertikal auf dem Boden stehen, drei Merkmalskarten berechnet, nämlich eine Bewegungskarte, eine Hautfarbentkarte und eine Orientierungskarte. Die drei Merkmalskarten werden zu einer aufgabenorientierten Salienzkarte kombiniert, in der helle Pixel die Positionen der potentiellen

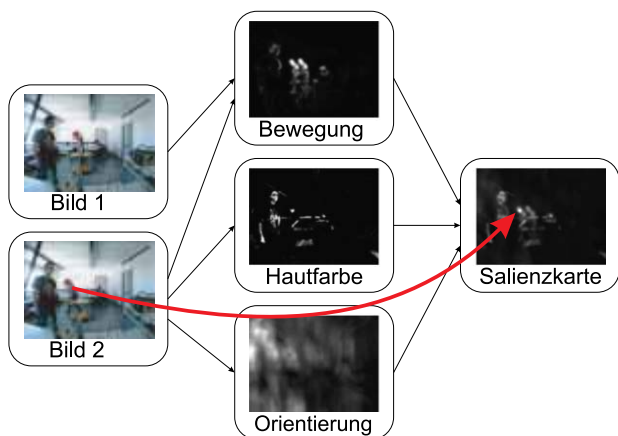


Bild 4 Merkmals- und Aufmerksamkeitskarten von zwei aufeinander folgenden Bildern.

Passanten im Eingangsbild zeigen. Um den Einfluss unterschiedlicher Lichtverhältnisse zu reduzieren wird eine Normierung im roten, grünen und blauen Kanal des Eingangsbildes durchgeführt. Um eine höhere Robustheit zu erreichen wird die Hautfarbentkarte aus einer gewichteten Summe dreier verschiedener Karten berechnet, die im normierten RGB-, HSV- und YCrCb-Farbraum vorliegen. Dabei wird in jedem Farbraum jedes Pixel in Hautfarbe oder Nicht-Hautfarbe klassifiziert.

Für die Bewegungskarte wird aus zwei direkt aufeinanderfolgenden Eingangsbildern die absolute Differenz der Graustufen von jedem Pixel berechnet. Um kleine Bewegungen zu kompensieren, die durch das Wackeln der Kamera während der Bewegung auftreten, werden die Pixeloffsets in x - und y -Richtung, k und l , zwischen zwei aufeinanderfolgenden Bildern geschätzt. Dabei wird ein Optimierungsproblem wie folgt gelöst:

$$Q = \min_{k,l} \sum_{i=1}^N (I_1(x_i, y_i) - I_2(x_i - k, y_i - l)) \quad (1)$$

wobei $k \in (-k_{\max}, k_{\max})$ und $l \in (-l_{\max}, l_{\max})$. N bezeichnet dabei die Anzahl von Pixeln (x_i, y_i) im Bild. Die Orientierungskarte wird mit Hilfe eines Gaborfilters berechnet, das vertikale Merkmale im Bild lokalisiert.

Die hellste Stelle in der berechneten Salienzkarte markiert die Region, in der sich am wahrscheinlichsten ein Mensch befindet und wird zum Fokus der Aufmerksamkeit des Roboters. Der Kamerakopf wird so gesteuert, dass diese Stelle in die Bildmitte gebracht wird. Somit zeigt der Roboter sein Interesse, mit Menschen zu interagieren. Um die Rechenkosten zu reduzieren, wird der Algorithmus in einem Suchfenster ausgeführt. Die Größe des Suchfensters ändert sich ständig zwischen 250×220 und 640×480 , abhängig von der Position des Fokus im vorhanden Bild (siehe Bild 5). Auf diese Weise kann der Fokus des Roboters auch relativ konstant auf einem ausgewählten Passant bleiben.

Da eine maximale saliente Stelle in einer Salienzkarte permanent existiert, tritt bei diesem einfachen und

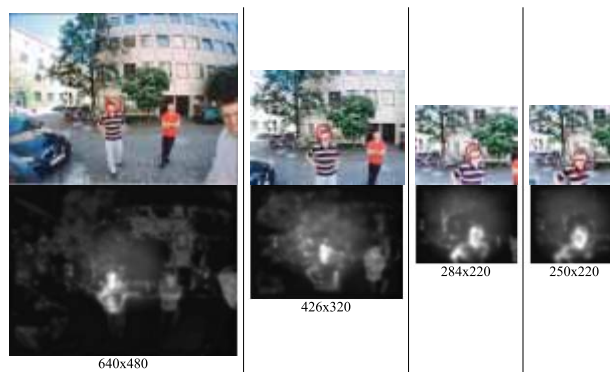


Bild 5 Dynamische Anpassung der Suchfenstergröße während der Bildverarbeitung.

realzeitfähigen Ansatz False-Positive-Ergebnis häufig auf, die teilweise durch 3D Lageschätzung untergedrückt ist. Mehrere Ergebnisse finden Sie im Kap. 4.2.

Erkennung der menschlichen Körperhaltung

Eine der natürlichsten und intuitivsten Eingabemöglichkeit neben Sprache, ist das Verwenden von Gesten. Der bei ACE gewählte Algorithmus verwendet ein Stereokamerasystem, das eine dreidimensionale Repräsentation der Umgebung in Form einer Punktwolke übergibt [21].

Sobald ACE Informationen über den Weg benötigt, wird ein Passant mit den zuvor beschriebenen Algorithmen gesucht, angesprochen und gebeten, in die Richtung zu zeigen, in die der Roboter fahren soll. Von dem Menschen wird mit der Stereokamera ein Bild aufgenommen aus dem eine 3-dimensionale Punktwolke berechnet wird. Anschließend wird diese Punktwolke segmentiert, d.h. Strukturen, die als Menschen identifiziert werden, werden von der Umgebung isoliert. Hierfür wird ein Algorithmus verwendet, der beginnend beim Kopf lokal zusammenhängende Strukturen sucht. Anschließend wird für jeden Menschen ausgehend vom Kopf ein Modell des menschlichen Körpers mit 15 Gelenken und insgesamt 28 Freiheitsgraden iterativ eingepasst. Da die Auflösung einer Stereokamera zu gering ist um Hände und Füße ausreichend hoch aufzulösen, werden sie im Modell nicht beachtet. Bild 6 veranschaulicht, wie dieser Algorithmus funktioniert. Ausgehend von einer Referenzstellung wird jedes Gelenk dabei so bewegt, dass es optimal in die Punktwolke passt. Hierbei werden nur solche Bewegungen betrachtet, die von einem Menschen auch ausgeführt werden können. Danach werden die mit dem Gelenk verbundenen Gelenke ebenso eingepasst.

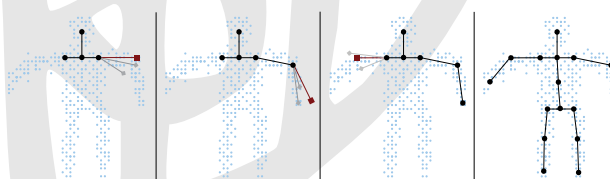


Bild 6 Iteratives einpassen eines Modells des menschlichen Körpers in die 3-dimensionale Punktwolke.

Sollte für ein Gelenk keine sinnvolle Lösung gefunden werden können, so wird dieses und alle nachfolgenden Gelenke verworfen. Dadurch kann der Algorithmus auch teilweise verdeckte Körperhaltungen erkennen.

Bei einer Gruppe von dicht aneinander stehenden Menschen können bei der Segmentierung Fehler auftreten, bei denen Körperteile falsch zugeordnet werden, wodurch falsche Körperhaltungen erkannt werden.

Erkennen von Kreuzungen und Folgen des Gehsteigs

Sicherheit ist ein wichtiger Aspekt beim Navigieren eines Roboters durch eine Großstadt. ACE muss sich gemeinsam mit Fußgängern auf dem Gehsteig bewegen und darf Kreuzungen nicht alleine überqueren. Somit muss der Roboter zum Einen in der Lage sein, Gehsteige sowie Kreuzungen zu erkennen. Dafür wird ein dualer Ansatz verwendet, der zum Einen über die Form des Gehsteigs versucht, Kreuzungen zu erkennen. Weiterhin wird davon ausgegangen, dass an jeder Kreuzung mindestens ein Verkehrsschild oder eine Ampel vorhanden ist, was ebenfalls zum Erkennen von Kreuzungen berücksichtigt wird.

Für die Erkennung des Gehsteigs wird ein texturbasierter Ansatz verwendet. Hierbei wird angenommen, dass das Bodensegment direkt vor dem Roboter zum Gehsteig gehört, was mit den erwähnten Laserscannern überprüft werden kann. Dieser Bereich ist in Bild 7 mit einem Viereck markiert. Nun wird diese Referenztextur über Histogramme mit dem Rest des Bildes verglichen. Um wechselnden Texturen erkennen zu können, werden bestehende Algorithmen [22; 23] mit einem Gedächtnis erweitert. Wenn die Texturen übereinstimmen, so wird davon ausgegangen dass es sich um einen Teil des Gehsteigs handelt. Das so entstandene Binärbild wird nun in eine Karte transformiert und kann mit einer Tex-

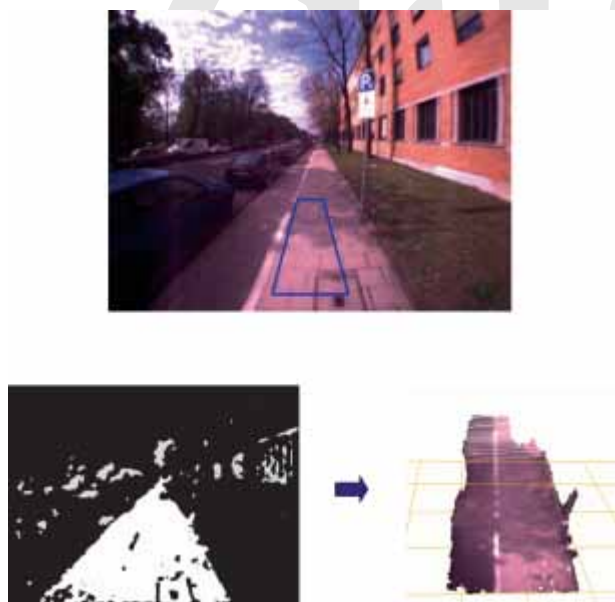


Bild 7 Veranschaulichung des zur Erkennung des Gehsteigs eingesetzten Algorithmus.

tur überzogen werden. Der untere Teil von Bild 7 zeigt das Ergebnis dieses Vergleichs. Um mit verschiedenen Beschaffenheiten des Gehsteigs zurechtzukommen, baut sich der Algorithmus eine Bibliothek mit verschiedenen Referenztexturen über einen bestimmten Zeitraum hinweg auf.

In einigen Fällen ist der vorgestellte Algorithmus jedoch nicht geeignet. Bei starker Sonneneinstrahlung entstehen zu einem Blooming-Effekte auf der Kamera, die zu falschen Ergebnissen führen, und zum anderen abrupte Übergänge zwischen in der Sonne und im Schatten liegenden Teilen des Gehsteigs. Solche Übergänge werden zunächst als Hindernis erkannt. Wenn sich viele Menschen direkt vor dem Roboter befinden, so werden diese als Hindernisse erkannt. Da sie jedoch den Gehsteig hinter sich verdecken, wird ein Großteil des gesamten Gehsteigs als besetzt angenommen.

Für die Erkennung der Verkehrsschilder und Ampeln wird der Rapid Object Detection Algorithmus der OPENCV-Bibliothek verwendet, der mit über 10 000 Bildern trainiert wurde. Insgesamt wurden 12 verschiedene Verkehrszeichen gelernt, wobei für jedes ein eigener Klassifikator trainiert wurde. Für jeden Klassifikator ist eine Berechnungszeit von ca. 100 ms nötig, was bei einer Durchschnittsgeschwindigkeit von 0,5 m/s und dem großen Blickwinkel der Kamera zu robusten Ergebnissen führt, da Schilder über mehrere Bilder gesehen und Treffer dadurch verifiziert werden können.

Um Straßen sicher überqueren zu können, folgt ACE einem T-Shirt mit aufgedruckten Schachbrettmuster. Hierfür wird der Algorithmus der OPENCV-Bibliothek verwendet.

3.3 Interaktion

Die Mensch-Roboter-Interaktion ist ein wichtiger Bestandteil des Gesamtsystems, da der Roboter weder über vorgegebenes Kartenwissen noch über einen GPS-Sensor verfügt und Informationen über den globalen Weg von Passanten erfragen werden muss. Diese Weginformation wird zur globalen Pfadplanung innerhalb der durch

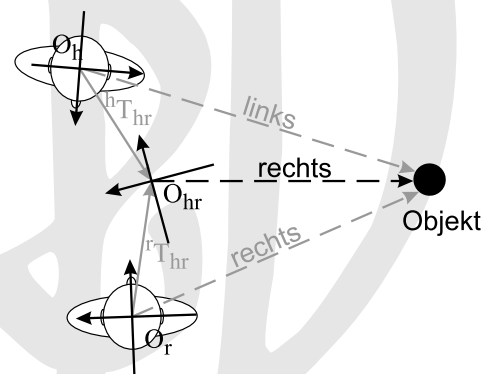


Bild 8 Veranschaulichung des Problems der verschiedenen Referenzsysteme O_h und O_r . Erst nach den Koordinatentransformationen ${}^rT_{hr}$ bzw. ${}^hT_{hr}$ lässt sich die Lage eines Objekts eindeutig über ein gemeinsames Koordinatensystem O_{hr} beschreiben.

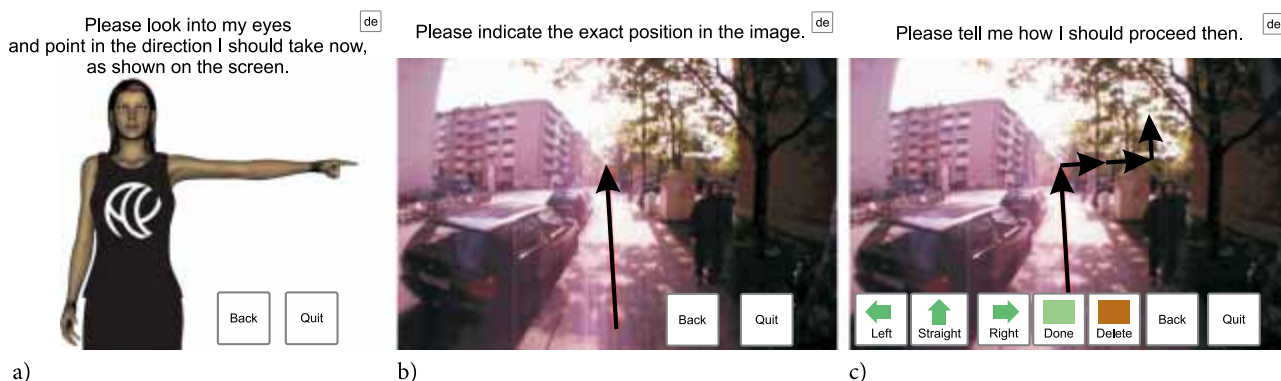


Bild 9 Beispiele des Touchscreens, während des Angleichens der Referenzsysteme (a, b) und der Eingabe von Richtungsangaben (b, c).

SLAM erworbenen lokale Karte zur Navigation benutzt. Im Folgenden wird ein kurzer Überblick über das *Interactions* Subsystem gegeben, eine ausführliche Beschreibung ist in [8] zu finden.

Die Interaktion wird von einem Zustandsautomaten gesteuert, der den Menschen durch den Dialog führt und Signale zu den beiden anderen Modulen sendet, bzw. Signale von ihnen empfängt. Hat *ACE* einen Menschen als Interaktionspartner gefunden, wird dieser aufgefordert den Weg zum Ziel zu beschreiben. Es könnte dabei allerdings zu Doppeldeutigkeiten der Informationen kommen, wenn nicht geklärt wäre, aus wessen Sicht der Weg beschrieben wird, wie in Bild 8 veranschaulicht. Daher wird der Mensch aufgefordert, die erste Richtung mit einer Geste anzugeben, was Koordinatentransformationen ${}^rT_{hr}$ bzw. ${}^hT_{hr}$ der Referenzsysteme O_h bzw. O_r , in ein gemeinsames Referenzsystem O_{hr} bewirkt. Von dem gemeinsamen Referenzsystem aus lassen sich alle weiteren Richtungen eindeutig beschreiben, daher werden diese dann über Pfeiltasten auf dem Touchscreen eingegeben, wie in Bild 9 zu sehen.

Die Wegbeschreibungen von Passanten werden als topologischer Routen-Graph $G_T \langle N_i, E_i \rangle$ intern repräsentiert, ähnlich wie in [24]. Der Routen-Graph besteht aus Knoten $N_i(t_j)$, die Landmarken vom Typ t_j entsprechen, und Kanten $E_i(a_i, N_k, N_l)$, die Aktionen a_i entsprechen, die zwei Landmarken N_k und N_l verbinden. Weiterhin wird den Kanten ein Wert $c_j \in [0, 1]$ zugewiesen der besagt, wie plausibel die Information ist. Dieser Wert hängt von der Anzahl der Bestätigungen n_c^j und Ablehnungen n_r^j der Information ab. Der jeweils nächste Knoten im topologischen Routen-Graphen wird als globaler Wegpunkt an das *Navigations*-Modul geschickt. Hat *ACE* die nächste Landmarke erreicht, so wird der Routen-Graph mit den metrischen Werten von den Sensoren aktualisiert. So entsteht nach und nach ein metrischer Routen-Graph, der genutzt werden könnte, um den Weg auch wieder zurück zum Ausgangspunkt zu finden.

4 Experimentelle Ergebnisse

Am 31. August 2008 wurde ein Experiment durchgeführt, bei dem *ACE* erfolgreich den Weg vom Campus

der TU München zum Marienplatz zurücklegte. Der zurückgelegte Weg lässt sich in Bild 1 erkennen. Für die Strecke von 1,5 km benötigte der Roboter ca. 5 Stunden und interagierte mit 38 Passanten. Diese große Zahl ergibt sich insbesondere durch neugierige Passanten, die viele Interaktionen selbst initiierten. Bis zum Erreichen der Fußgängerzone gab es insgesamt 12 Interaktionen. In der Fußgängerzone wurde *ACE* 21-mal von Passanten unterbrochen und initiierte lediglich 5 Interaktion selbst.

Während des Experiments wurde *ACE* aus Sicherheitsgründen von den Entwicklern begleitet. Dabei wurden ständig wichtige Daten von dem Roboter zu einem externen Laptop geschickt, um die korrekte Funktion überwachen zu können.

4.1 Navigation

Bild 13 zeigt auf der linken Seite ein $200\text{ m} \times 200\text{ m}$ Occupancy-Grid des SLAM-Moduls. Während des Experiments lief das SLAM-Modul mit 2 Hz, verwendete 200 Partikel und generierte Karten von $50\text{ m} \times 50\text{ m}$ bei einer Auflösung von 15 cm. Die Pfadplanung lief ebenfalls mit 2 Hz.

4.2 Bildverarbeitung

Innerhalb eines 11-minütigen Tests unter Einsatzbedingungen wurde für Passanten, die bis zu 8 Meter von der Kamera entfernt waren, eine Detektionsrate von 92,6% und eine False-Positive-Rate von 9,1% erreicht. Die maximale Rechenzeit für ein Bild mit 640×480 betrug 0,7 s, während die minimale Rechenzeit für Bilder mit 250×220 nur 0,2 s betrug. Mit Hilfe der GPU-Implementierung wurde eine 30%tige Beschleunigung der Berechnung der Hautfarbenkarte ermöglicht.

Bild 10 veranschaulicht die Ergebnisse der Erkennung von Menschen. In den Bildern der oberen Reihe ist der als Aufmerksamkeitsfokus ausgewählte Mensch mit einem Kreis markiert. Im nächsten Schritt bewegt sich der Kamerakopf, um den ausgewählten Menschen in die Bildmitte zu bringen. Die Bilder auf der unteren Reihe sind die Eingangsbilder nach der Kamerakopfbewegung. Bei Personengruppen wird die Person, die die maximalen Salienz auf der Salienzkarte aufweist, als Interaktionspartner



Bild 10 Kamerabewegung während der Erkennung von Menschen.



Bild 11 Mögliche Fehler bei der Erkennung von Menschen.

ausgewählt. Bei mehreren gleichwertigen Salienzwerten wird die Person in der Bildmitte ausgewählt. Durch dieses Verhalten signalisiert ACE Interesse an dem gegenwärtig ausgewählten Interaktionspartner.

Bild 11 zeigt zwei mögliche Fehlerfälle. In der linken Spalte wurde ein Teil eines Gebäudes im Hintergrund ausgewählt, da es eine ähnliche Farbe wie die menschliche Hautfarbe aufweist. In der rechten Spalte wurden sich bewegende Blätter als Kandidaten für Menschen ausgewählt. Um die beiden typischen Fehler zu vermeiden, wird die 3D Lage eines Menschen begrenzt. Außerdem sollte ein optimales Gewicht zwischen den verschiedenen Merkmalskarten festgelegt werden.

Bild 12 zeigt das Ergebnis der Erkennung der Körperhaltung in fünf verschiedenen Fällen unter Laborbedingungen. Der vorgestellte Algorithmus wurde während der Fahrt verwendet, allerdings wurden aus Performance-Gründen keine Daten aufgenommen. Die erste Reihe zeigt die Kamerabilder, gefolgt von den ermittelten Körperhaltungen aus verschiedenen Blickrichtungen. Wie sich erkennen lässt, berechnet der Algorithmus auch bei mehreren Menschen in einem Bild die Körperposen. Bei einem Laborexperiment mit

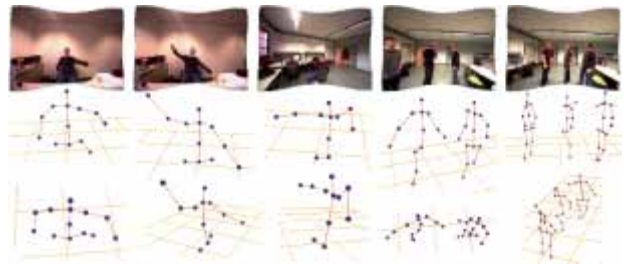


Bild 12 Ergebnisse der Detektion der Körperhaltung.

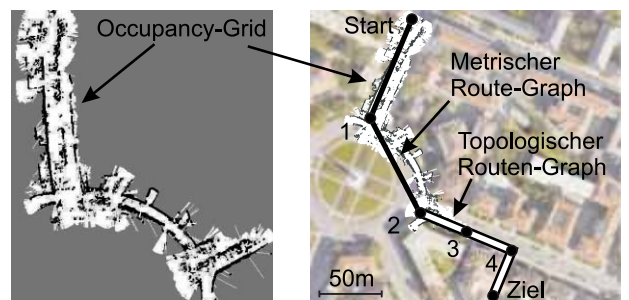


Bild 13 Links: Beispiel eines vom Navigationsmodul erstellten Occupancy-Grids. Rechts: ein Routen-Graph vom Interaktionsmodul aus Wegbeschreibungen von Passanten. Occupancy-Grid und Routen-Graph im Vergleich zum Satellitenbild.

150 Messungen war er dabei in der Lage, 80% der Gesten richtig zu erkennen. Dabei wurde die Richtung gemessen, in der die Testperson gezeigt hat. Diese wurde mit dem Detektionsergebnis verglichen, wobei ein mittlerer Fehler von 7° erreicht wurde. Ab einem Fehler von 25° wurde die Messung als falsch angenommen. Bei der Erkennung von Kreuzungen konnten bei einer Testsequenz von 485 Bildern zwischen 79% und 88% der Schilder richtig erkannt werden. Bei 2% bis 6% der Bilder wurden fälschlicherweise nicht vorhandene Schilder gefunden.

4.3 Interaktion

Während des Experiments interagierte ACE mit 38 Menschen, wie z. B. in Bild 1 und Bild 10 zu sehen ist. Die Passanten wurden nicht vorher eingewiesen, wie sie mit dem Roboter interagieren können und hatten größtenteils kein technisches Vorwissen. Die Angaben der Passanten waren korrekt und der Roboter erreichte sein Ziel aufgrund der Wegbeschreibungen von Passanten. Nur in einem Fall wurde eine falsche Richtung angegeben, die dann aber bei der nächsten Interaktion von einem anderen Menschen korrigiert wurde. ACE bildete aus den Wegbeschreibungen einen Routen-Graphen, wie in Bild 13 dargestellt.

5 Zusammenfassung und Ausblick

In diesem Artikel wurde das *Autonomous City Explorer*-Projekt vorgestellt, dessen Schwerpunkte auf Aspekten der Wahrnehmung, Navigation und Mensch-Roboter-Interaktion in städtischen Umgebungen liegt. Dabei wurde gezeigt, dass sich Roboter auch ohne Kartenvorwissen oder GPS in ihnen unbekanntem Umgebungen

zurechtfinden können, indem Sie durch Interaktion mit Menschen die nötigen Informationen gewinnen. Die Ergebnisse dieses Pilotprojektes lassen sich auf zukünftige humanoide Roboter übertragen.

Der Roboter ACE hat es dabei erfolgreich geschafft, den Weg vom Campus der Technischen Universität München zum Marienplatz, dem Zentrum der Stadt München, zu finden. Hierfür wurden Algorithmen und Strategien aus verschiedenen Forschungsbereichen in einem Framework zusammengefasst. Diese dienen sowohl zum sicheren Navigieren in einer Großstadt, zum Erkennen und Folgen von Menschen, zur Gestenerkennung, zur Detektion von Gehsteigen und Verkehrszeichen als auch zur Interaktion mit zufällig ausgewählten Passanten. Die Reaktionen der Passanten haben gezeigt, dass Menschen Roboter in ihrer Umgebung akzeptieren und gewillt sind, Robotern bei der Ausführung ihrer Aufgaben zu helfen.

Das ACE-Projekt hat mannigfaltige Möglichkeiten für die weitere Entwicklung aufgezeigt. Wünschenswert sind beispielsweise weitere Eingabemöglichkeiten wie natürliche Spracherkennung. Die kognitive Architektur kann mit komplexeren Methoden zum Verstehen der Umgebung und einer intelligenteren Entscheidungsfindung erweitert werden. Auf lange Sicht existiert die Vision, eine Ebene semantischen Verstehens zu erreichen, auf der Roboter Namen und Eigenschaften von Orten und Objekten lernen und daraus logische Schlüsse ziehen können.

Danksagung

Die Autoren danken BlueBotics SA (www.bluebotics.com) für die Unterstützung mit der mobilen Plattform. Weiterhin wurde das Projekt zum Teil durch das DFG Exzellenzcluster *Cognition for Technical Systems – CoTeSys* (www.cotesys.org), durch die Europäische Kommission als Teil des *Robot@CWE-Projekts* (www.robot-at-cwe.eu) und durch das BMBF Nationale Netzwerk, *Bernstein-Zentrum für Computational Neuroscience München* (www.bccn-munich.de), unterstützt.

Literatur

- [1] R. Brooks, „New approaches to robotics“, *Science*, vol. 253, 1991.
- [2] Y. Morales, E. Takeuchi, A. Carballo, W. Tokunaga, H. Kuniyoshi, A. Aburadani, A. Hirose, Y. Nagasaka, Y. Suzuki, and T. Tsubouchi, „1 km autonomous robot navigation on outdoor pedestrian paths“, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.
- [3] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, „Probabilistic algorithms and the interactive museum tour-guide robot minerva“, *Int. J. Robotics Research*, vol. 19, (11), pp. 972–999, 2000.
- [4] H.-M. Gross, H.-J. Böhme, C. Schröter, S. Müller, A. König, C. Martin, M. Merten, and A. Bley, „Shopbot: Progress in developing an interactive mobile shopping assistant for everyday use“, in *Proc. IEEE Internat. Conf. on Systems, Man and Cybernetics (IEEE-SMC)*, 2008.
- [5] S. Lauria, G. Bugmann, T. Kyriacou, and E. Klein, „Instruction based learning: how to instruct a personal robot to find hal.“, in *Proc. European Workshop on Learning Robots*, 2001.
- [6] C. B. Ferrell and J. Velsquez, „Toward teaching a robot ‚infant‘ using emotive communication acts“, in *Proceedings of Simulation of Adaptive Behavior, workshop on Socially Situated Intelligence*, 1998.
- [7] G. Lidoris, K. Klasing, A. Bauer, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss, „The autonomous city explorer project: Aims and system overview“, in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 29–Nov 2 2007.
- [8] A. Bauer, K. Klasing, G. Lidoris, Q. Mühlbauer, F. Rohrmüller, S. Sosnowski, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss, „The autonomous city explorer: Towards natural human-robot interaction in urban environments“, *International Journal of Social Robotics*, pp. 127–140, 2009.
- [9] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, „A solution to the simultaneous localization and map building (SLAM) problem“, *IEEE Transactions on Robotics and Automation*, vol. 17, (3), pp. 229–241, 2001.
- [10] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, „Simultaneous localization and mapping with sparse extended information filters“, *Int. J. Robotics Research*, vol. 23, (7–8), pp. 693–716, 2004.
- [11] K. Murphy, „Bayesian map learning in dynamic environments“, in *Advances in Neural Information Processing Systems (NIPS)*, 1999.
- [12] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, „Fastslam: A factored solution to simultaneous localization and mapping“, in *National Conf. on Artificial Intelligence (AAAI), Edmonton, Canada*, 2002.
- [13] G. Grisetti, C. Stachniss, and W. Burgard, „Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling“, in *IEEE Int. Conf. of Robotics and Automation (ICRA), Barcelona, Spain*, 2005.
- [14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [15] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2000.
- [16] K. Klasing, G. Lidoris, A. Bauer, F. Rohrmüller, D. Wollherr, and M. Buss, „The autonomous city explorer: Towards semantic navigation in urban environments“, in *1st Int. Workshop on Cognition for Technical Systems*, 2008.
- [17] G. Lidoris, F. Rohrmüller, D. Wollherr, and M. Buss, „The autonomous city explorer (ace) project – mobile robot navigation in highly populated urban environments“, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.
- [18] B. B. Leibe and A. Leonardis, „Robust object detection with interleaved categorization and segmentation“, *IJCV Special Issue on Learning for Vision and Vision for Learning*, vol. 77, (1–3), pp. 259–289, 2008.
- [19] S. K. Singh, D. S. Chauhan, M. Vasta, and R. Singh, „A robust skin color based face detection algorithm“, *Tamkang Journal of Science and Engineering*, vol. 6, (4), pp. 227–234, 2003.
- [20] A. A. V. Vezhnevets and V. Sazonov, „A survey on pixel-based skin color detection techniques“, in *Proc. of Conf. Graphicon*, 2003.
- [21] Q. Mühlbauer, K. Kühnlenz, and M. Buss, „A model-based algorithm to estimate body poses using stereo vision“, in *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication*, 2008.
- [22] I. Ulrich and I. Nourbakhsh, „Appearance-based obstacle detection with monocular color vision“, in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 866–871, 2000.
- [23] J. S. Seng and T. J. Norrie, „Sidewalk following using color histograms“, *Journal of Computing Sciences in College*, vol. 23, (6), pp. 172–180, 2008.



[24] S. Werner, B. Krieg-Brückner, and T. Herrmann, „Modelling navigational knowledge by route graphs“, in *Ch. Freksa et al. (Eds): Spatial Cognition II 1849*, pp. 295–316, Springer, 2000.

Manuskripteingang: 15. Juli 2009

Dipl.-Ing. Quirin Mühlbauer ist seit Mai 2007 Doktorand am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: Bildverarbeitung, kognitives Verstehen der Umgebung.

Adresse: s. u., Tel.: +49 (0)89 28923444, E-Mail: qm@tum.de

Dipl.-Ing. Tingting Xu ist seit Oktober 2006 Doktorandin am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: Bildverarbeitung, visuelle Aufmerksamkeitssteuerung.

Adresse: s. u., Tel.: +49 (0)89 28923441, E-Mail: xu@tum.de

Dipl.-Ing. Andrea Bauer ist seit Dezember 2006 Doktorandin am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: Informationsgewinnung durch Mensch-Roboter-Kommunikation.

Adresse: s. u., Tel.: +49 (0)89 28923415, E-Mail: ab@tum.de

Klaas Klasing, M.Sc. M.Sc ist seit August 2006 Doktorand am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: Samplingbasiert Pfadplanung, 3D Perzeption.

Adresse: s. u., Tel.: +49 (0)89 28923409, E-Mail: kk@tum.de

Dipl.-Ing. Georgios Lidoris ist seit Oktober 2004 Doktorand am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: SLAM, Autonome Navigation.

Adresse: s. u., Tel.: +49 (0)89 28923417, E-Mail: georgios.lidoris@tum.de

Dipl.-Ing. Florian Rohrmüller ist seit Juni 2007 Doktorand am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: Autonome Navigation, Entscheidungsfindung in kooperativen Multi-Roboter Systemen.

Adresse: s. u., Tel.: +49 (0)89 28925775, E-Mail: rohrmueller@tum.de

Dipl.-Ing. Stefan Sosnowski ist seit November 2007 Doktorand am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: Non-verbale Mensch-Roboter-Interaktion, Emotionsmodellierung.

Adresse: s. u., Tel.: +49 (0)89 28923406, E-Mail: sosnowski@tum.de

Dr.-Ing. Kolja Kühnlenz ist seit 2007 Forschungsgruppenleiter am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: Mobile Roboter, Bildverarbeitung in der Robotik, Mensch-Roboter-Interaktion.

Adresse: s. u., Tel.: +49 (0)89 28923445, E-Mail: koku@tum.de

Dr.-Ing. Dirk Wollherr ist seit 2005 Forschungsgruppenleiter am Lehrstuhl für Steuerungs- und Regelungstechnik. Wissenschaftliche Schwerpunkte: Mobile Roboter, Mensch-Roboter-Interaktion, Objekterkennung und Verstehen der Umgebung, Schnelle Manipulation.

Adresse: s. u., Tel.: +49 (0)89 28923401, E-Mail: dw@tum.de

Univ. Prof. Dr.-Ing./Univ. Tokio Martin Buss ist Ordinarius des Lehrstuhls für Steuerungs- und Regelungstechnik der Technischen Universität München. Hauptarbeitsgebiete: Automatisierungstechnik, Mechanik und Robotik, multimodale Mensch-Maschine-Systeme, Optimierungsverfahren, nichtlineare und hybride diskret-kontinuierliche Regelungssysteme.

Adresse: Lehrstuhl für Steuerungs- und Regelungstechnik, Technische Universität München, Theresienstrasse 90, 80290 München, Tel.: +49 (0)89 28928395, E-Mail: mb@tum.de

Vorschau auf Heft 12/2010

Unsere nächste Ausgabe ist ein Schwerpunktheft zum Thema „Industrierobotik“ (Gastherausgeber: J. Hofschulte, M. Suppa) und enthält unter anderem folgende Beiträge:

- *Haddadin, S., Suppa, M., Fuchs, S., Bodenmüller, T., Albu-Schäffer, A., Hirzinger, G.:* Konzepte für den Roboterassistenten der Zukunft
- *Bischoff, R. et al.:* Aus der Forschung zum Industrieprodukt: Die Entwicklung des KUKA Leichtbauroboters
- *Som, F.:* Intuitives Bedieninterface für effiziente Mensch-Roboter-Kooperation

Weitere Informationen über geplante Hefte, ausführliche Informationen über die in den letzten Heften der **at** erschienenen Beiträge sowie Hinweise für Autoren finden Sie im Internet unter <http://www.at-automatisierungstechnik.de>.