

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN
INSTITUT FÜR INFORMATIK
ABTEILUNG III

Robuste Roboterlokalisierung mit omnidirektionaler Bildsensorik

Diplomarbeit

SIMONE FRINTROP
31. Oktober 2001

Erstgutachter: PD DR. JOACHIM HERTZBERG

Zusammenfassung

Robotik ist ein Bereich der Informatik, der in letzter Zeit zunehmend an Bedeutung gewinnt. Fortschritte werden insbesondere in dem Bereich der autonomen mobilen Systeme erwartet. Diese sollen in der Lage sein, nicht nur in speziell für sie gefertigten oder angepassten Umgebungen zu agieren, sondern auch in natürlichen oder für den Menschen geschaffenen Umgebungen. Eine fundamentale Aufgabe autonomer mobiler Roboter ist die Navigation. Sie basiert auf robusten Lokalisierungsmethoden, die auch mit gestörten Sensordaten erfolgreich arbeiten sollten.

Diese Arbeit befaßt sich mit der Lokalisierung von Indoor-Robotern. Als Sensor wird dabei eine omnidirektionale Kamera verwendet, mit der 360°-Panoramabilder aufgenommen werden. Eine signifikante Datenreduktion der Bilddaten wird durch Projektion in den Eigenraum erreicht; Appearance-based Matching und geeignete Klassifikationsverfahren ermöglichen die Zuordnung von Bildern zu Positionen in der Umgebung. Es werden zwei Ansätze zur Lokalisierung störbehafteter Bilddaten vorgestellt. Während die erste Methode mit den kompletten Panoramabildern arbeitet, werden die Bilder bei der zweiten Methode in einzelne Sektoren aufgeteilt, die dann zur Klassifikation genutzt werden. Dabei wird der visuelle Kontext berücksichtigt, der durch die Reihenfolge der Sektoren gegeben ist.

Diese Arbeit legt einen Schwerpunkt auf die Generierung robuster Methoden. So wird getestet, wie die vorgestellten Ansätze mit Schwierigkeiten umgehen, beispielsweise mit Rotationen des Roboters, Beleuchtungsvarianzen oder Okklusionen von Teilbereichen der Bilder durch Personen oder Gegenstände. Dabei werden die Möglichkeiten und die Grenzen der einzelnen Methoden erläutert, sowie Verbesserungsmöglichkeiten vorgestellt.

Danksagung

Allen voran möchte ich meinem Erstgutachter PD Dr. Joachim Hertzberg danken, der mir einen Einblick in dieses interessante Forschungsgebiet ermöglicht hat. Er hat diese Arbeit mit wertvollen Hinweisen unterstützt und hat sich stets Zeit für Fragen und Diskussionen genommen. Ein besonderer Dank gilt auch Lucas Paletta, der diese Diplomarbeit betreut hat. Mit Unterstützung durch und in Zusammenarbeit mit ihm ist ein Großteil der vorgestellten Methoden und Experimente entwickelt worden. Er hat mich in das wissenschaftliche Arbeiten eingeführt und stand mir jederzeit mit Rat und Tat zur Seite.

Ganz besonders möchte ich auch Henrik danken, der sich stets an meinem Thema interessiert gezeigt, Probleme mit mir durchdiskutiert und einen Großteil der Arbeit Korrektur gelesen hat. Nicht zuletzt hat er mich ermutigt und aufgemuntert und mir geholfen, meine Freizeit zu genießen.

Außerdem einen schönen Dank allen Korrekturlesern und -leserinnen für die Fehlersuche, dem Mensatisch für die mittägliche Gesellschaft und der Römerlagerbar für den Nachschub an entspannenden Getränken. Schließlich möchte ich noch besonders meinen Eltern dafür danken, daß sie mich während meines gesamten Studiums in jeder Hinsicht unterstützt und ermutigt haben und mir diesen Ausbildungsweg ermöglicht haben.

Inhaltsverzeichnis

Zusammenfassung	iii
Danksagung	v
1 Einführung	1
1.1 Roboterlokalisierung	1
1.2 Wissenschaftlicher Beitrag	6
1.3 Übersicht	6
I Methoden	9
2 Datenkompression mit PCA	11
2.1 PCA	11
2.2 Der Eigenraum	14
2.3 Komplexität	21
3 Bildklassifikation	23
3.1 Appearance-based Matching	23
3.2 Nächster-Nachbar Klassifikator	25
3.3 Bayes'sche Klassifikation	28
3.4 Neuronale Netze	29
4 Vollbild-Lokalisierung	31
4.1 Vorbereitung	31
4.2 PCA - Lernen der Umgebung	33
4.3 Lokalisierung	33
4.4 Problemfaktoren	34
5 Sektoren-Lokalisierung	37
5.1 Motivation	38
5.2 Multisektorklassifikation	38
5.3 Visueller Kontext	43
5.4 Rotationsunabhängige Lokalisierung	46

5.5	Komplexität	47
II	Experimente	49
6	Das Arbeitsszenario	53
6.1	Roboter und omnidirektionale Kamera	53
6.2	Die Panoramabilder	55
6.3	Die Büroumgebung	56
6.4	Fehlerauswertung und Präsentation	57
7	Ergebnisse der Vollbild-Lokalisierung	61
7.1	Allgemeine Tests	61
7.2	Beleuchtungsvarianzen	68
7.3	Okklusionen	81
8	Ergebnisse der Sektoren-Lokalisierung	87
8.1	Anzahl von Sektoren	87
8.2	Rotationen	90
8.3	Okklusionen	91
9	Vergleich der Lokalisierungsmethoden	99
9.1	Normale Bilder	99
9.2	Rotationen	101
9.3	Okklusionen	103
9.4	Diskussion	105
10	Zusammenfassung und Ausblick	109
10.1	Zusammenfassung	109
10.2	Ausblick	110

Kapitel 1

Einführung

Lokalisierung ist eine fundamentale Aufgabe autonomer mobiler Roboter. Diese Arbeit stellt einen Ansatz zur Lokalisierung solcher Roboter mittels omnidirektionaler Bildsensoren vor. Dabei werden robuste Lokalisierungsmethoden entwickelt, die auch unter realen Bedingungen gute Ergebnisse liefern. Diesbezüglich werden zwei Schwerpunkte gesetzt: Zum einen wird die Robustheit gegenüber Beleuchtungsvarianzen behandelt; zum anderen wird ein vielversprechender Ansatz zum Umgang mit starken Okklusionen durch Berücksichtigung des lokalen visuellen Kontextes vorgestellt.

Dieses Kapitel motiviert in Abschnitt 1.1 zunächst die Lokalisierung von Robotern im allgemeinen, sowie die Verwendung eines omnidirektionalen Bildsensors im besonderen. Desweiteren wird die Komprimierung der Daten mit einem speziellen Kompressionsverfahren erläutert sowie verschiedene Lokalisierungsmethoden vorgestellt. Abschnitt 1.2 beschreibt den wissenschaftlichen Beitrag der vorliegenden Arbeit und Abschnitt 1.3 gibt schließlich einen Überblick über den Aufbau der Arbeit.

1.1 Roboterlokalisierung

Robotik ist ein Bereich der Informatik, der in letzter Zeit zunehmend an Interesse gewinnt. Große Fortschritte werden insbesondere in dem Bereich der autonomen mobilen Systeme erwartet (Dudek & Jenkin [DJ00], Pfeiffer & Scheier [PS99], Arkin [Ark98] und Murphy [Mur00]). Diese sollen in der Lage sein, nicht nur in speziell für sie gefertigten oder angepaßten Umgebungen zu agieren, sondern auch in einer Umwelt, in der solche Anpassungen nicht möglich oder nicht gewünscht sind. Dies können natürliche Umgebungen wie der Mars oder Vulkane sein (Dante Roboter [WPB95]) oder auch für den Menschen geschaffene Umgebungen, wie etwa öffentliche Einrichtungen, z.B. Büroumgebungen oder Museen (Burgard *et al.* [BCF⁺00]).

Eine wichtige Aufgabe autonomer mobiler Roboter ist es, sich in ihrer Umgebung selbstständig zurechtfinden und in ihr navigieren zu können. Eine Voraussetzung erfolgreicher Navigation ist aber zunächst, daß der Roboter seine eigene Position kennt. Damit die dafür benötigten Lokalisierungsmethoden auch unter realen Bedingungen gute Ergebnisse liefern,

sollen diese Methoden möglichst robust sein. So muß der Roboter beispielsweise mit Beleuchtungsvariationen umgehen können, die durch Veränderungen des Lichteinfalls durch Wetterveränderungen oder Tageszeit hervorgerufen werden, sowie mit räumlichen Veränderungen der Umgebung durch Personen oder veränderte Positionen von Gegenstände.

Sensorik

Damit ein autonomer mobiler Roboter seine Position bestimmen kann, muß er seine Umgebung zunächst wahrnehmen können. Dazu benötigt er Sensoren, die ihm Informationen über seine Umwelt liefern. Als Sensoren verwendet man üblicherweise Laser, Sonarsensoren, taktile und visuelle Sensoren [Nou98]. Im folgenden soll auf einige Vor- und Nachteile verschiedener Sensoren eingegangen werden. Man beachte aber, daß sich meist die besten Ergebnisse erzielen lassen, wenn verschiedene Sensoren kombiniert werden. Diese Methodik nennt man *Sensor fusion* (Dasarathy [Das94]). Ein repräsentatives Beispiel einer Architektur, die dieses gewährleistet, ist die Rhino-Architektur (Burgard *et. al* [BBC⁺95]).

Zur Lokalisierung werden häufig Abstandssensoren wie Laser oder Sonarsensoren verwendet. Diese haben den Vorteil, daß sie eine nicht sehr große und damit leicht zu speichernde und zu verarbeitende Informationsmenge liefern. Ein Nachteil dieser Sensoren ist, daß sie nicht den gesamten dreidimensionalen Raum um den Roboter abdecken, sondern üblicherweise nur einen Scan in einer bestimmten Höhe durchführen. So können Gegenstände mit anderer Höhe, beispielsweise Tischkanten, leicht übersehen werden.

Visuelle Sensoren

Eine andere Möglichkeit ist die biologisch motivierte Verwendung visueller Sensoren. Nicht nur der Mensch, auch viele Tiere nutzen vorwiegend visuelle Information, um sich zu orientieren und sich in ihrer Umgebung zurechtzufinden. Siehe dazu auch die Arbeiten von Judd & Collett [JC98], Dill, Wolf & Heisenberger [DWM93], sowie Epstain & Kanwisher [EK98].

Zur Lokalisierung von Robotern werden visuelle Sensoren allerdings seltener verwendet. Der Hauptgrund dafür ist die sehr große Informationsmenge, aus der sowohl Speicherprobleme als auch Schwierigkeiten bei der Verarbeitung und Interpretation der Daten resultieren. Andererseits liegt gerade in dieser Informationsfülle ein großes Potential. Geeignet interpretiert lassen sich Kamerabilder sinnvoll zur Lokalisierung nutzen und können darüber hinaus zur Objekterkennung wiederverwendet werden und beispielsweise als visuelle Landmarken dienen (Thrun [Thr98]).

Zudem handelt es sich bei visuellen Sensoren meistens um Flächensensoren, so daß die gesamte Höhe des Raumes abgedeckt wird. Eine Ausnahme stellt Röfer in [Rö98] vor. Er verwendet eindimensionale farbige Panoramabilder, die mit einem speziellen optischen Sensor aufgenommen wurden. Dieser führt, ähnlich wie Laser- oder Sonarsensoren, einen Rundum-Scan der Umgebung in einer bestimmten Höhe durch.

Omnidirektionale Kameras

Ein Nachteil von Kamerabildern ist, daß sie meist nur einen Ausschnitt der Umgebung zeigen. So steht entweder nur Information aus einer bestimmten Richtung zur Verfügung oder aber es müssen mehrere Kamerabilder in verschiedenen Richtungen aufgenommen werden. Um diesen Nachteil zu umgehen, ohne auf die Vorteile eines visuellen Sensors verzichten zu müssen, wird in dieser Arbeit eine omnidirektionale Kamera verwendet. So muß an einer Position nur ein Kamerabild aufgenommen werden. Dadurch wird der Speicherplatz reduziert, eine Rotation von Roboter oder Kamera überflüssig gemacht und Mehrdeutigkeiten in der Interpretation der sich überlappenden Teilbilder vermieden.

Abbildung 1.1 zeigt links ein mit der omnidirektionalen Kamera aufgenommenes Bild. Es wurde im Flur einer Büroumgebung aufgenommen. In der Mitte zeigt sich ein kreisförmiger Ausschnitt der Decke, der durch eine zentrale Öffnung aufgenommen wurde. Dieses Kamerabild wird in ein rechteckiges Panoramabild transformiert, wie auf der rechten Seite der Abbildung zu sehen ist. Mit diesen Panoramabildern soll im folgenden gearbeitet werden. Omnidirektionale Sensoren wurden in letzter Zeit häufig zur Roboternavigation eingesetzt, siehe z.B. die Arbeiten von Jogan & Leonardis [JL00], Aihara *et al.* [AIYT98] und Winters *et al.* [WGLSV00].

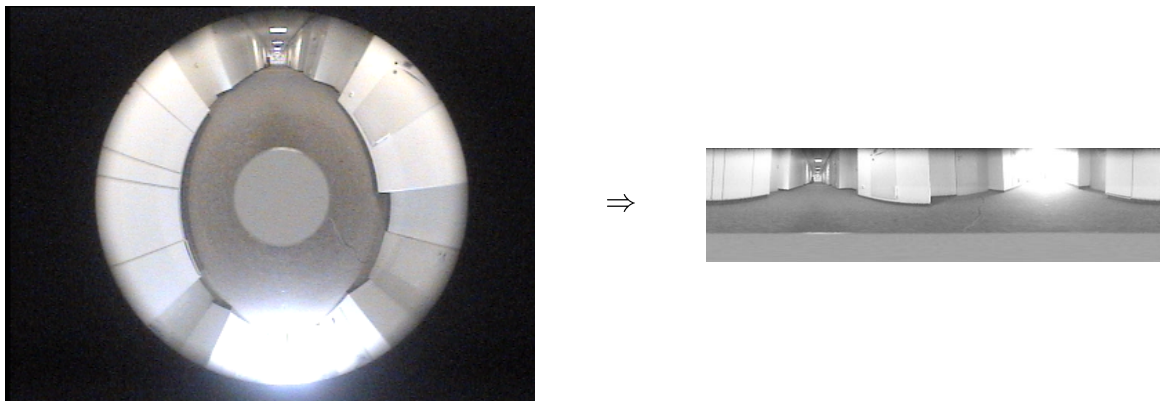


Abbildung 1.1: Ein mit dem omnidirektionalen Sensor aufgenommenes Kamerabild links und das transformierte Panoramabild rechts.

Lokalisierungsmethoden

Weltmodelle

Damit der Roboter sich lokalisieren kann, muß die durch die Sensoren erhaltene Information geeignet interpretiert werden. Die meisten Lokalisierungsmethoden verwenden dazu ein Modell der Umgebung. Sie werden eingeteilt in geometrische und topologische Ansätze. Geometrische Ansätze besitzen eine maßstabsgetreue Karte ihrer Umgebung. Die aus den

Sensordaten extrahierte Information wird dabei mit dem zuvor erstellten Modell verglichen. Typische Beispiele für geometrische Modelle sind occupancy grids für Sonar- oder Laser-basierte Lokalisierung (Moravec [Mor88] und Burgard *et al.* [BFHS96]). Bei topologischen Ansätzen wird die Karte durch einen Nachbarschaftsgraphen realisiert. Hierbei werden Positionen durch Knoten repräsentiert und Kanten zwischen zwei Knoten stellen die Nachbarschaft zweier Positionen dar.

Ein Ansatz, der ohne explizites Umgebungsmodell auskommt, ist das sogenannte Memory-based Matching bzw. Appearance-based Matching. Diese Herangehensweise vergleicht die Sensorinformation direkt mit den im Speicher abgelegten Daten. Arbeiten dazu finden sich bei Maeda *et al.* [MKS97], Matsumo *et al.* [MIII99] und Kröse *et al.* [KBVM99].

In dieser Arbeit wird Appearance-based Matching mit einem topologischen Umgebungsmodell kombiniert (siehe Abbildung 1.2). Die Büroumgebung, in der der Roboter agiert, wird in drei Abschnitte eingeteilt, die aus Flurteilen bzw. einem Büro bestehen. Diese Abschnitte entsprechen den drei Knoten eines topologischen Modells, die durch Kanten miteinander verbunden sind. Innerhalb jedes Abschnitts werden die Bilder dann mittels Memory-based Matching direkt mit gespeicherten Referenzdaten verglichen.

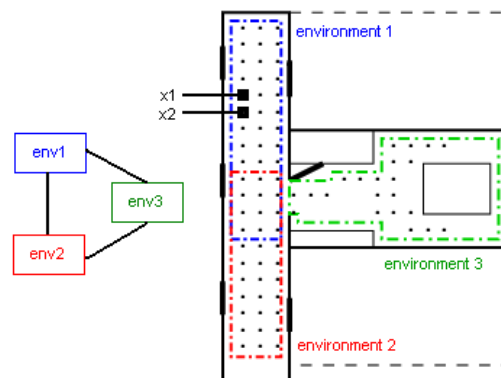


Abbildung 1.2: Maßstabsgetreue Kartierung der dieser Arbeit zugrundeliegenden Büroumgebung (rechts), und zugehöriges topologisches Modell dieser Umgebung (links)

Diese Arbeit legt den Schwerpunkt auf die Lokalisierung durch Memory-based Matching innerhalb eines Umgebungsabschnitts und geht davon aus, daß zuvor durch globale Lokalisierung (z.B. mit Hilfe der Odometrie) festgestellt wurde, in welchem Abschnitt sich der Roboter befindet.

Datenkompression

Da die Panoramabilder eine große Informationsmenge darstellen, ist es zunächst notwendig, die Datenmenge stark zu reduzieren. Die in dieser Arbeit verwendete Datenkompressionsmethode basiert auf der PCA (Principal Component Analysis), auch bekannt als Karhunen-Loève Transformation. Dabei wird der Eigenraum der Kovarianzmatrix der Bilddaten er-

zeugt. Die Panoramabilder werden dann in diesen Eigenraum projiziert. Dieser besitzt die Eigenschaft, daß sich seine Dimension unter maximalem Informationserhalt drastisch verkleinern läßt. So erhält man an Stelle der Panoramabilder sehr viel kleinere Datenvektoren.

PCA wird auch häufig zur Objekterkennung eingesetzt (Murase & Nayar [MN95], Turk & Pentland [TP91] und Paletta [Pal99]). Im Bereich der Roboternavigation finden sich Arbeiten mit PCA bei Kröse *et al.* [KBVM99], Jogan & Leonardis [JL00] und Maeda *et al.* [MKS97].

Klassifikation

Um die Lokalisierung zu vollenden, muß den Sensordaten eine Roboterposition zugeordnet werden. Dazu benötigt man Klassifikationsverfahren. Bekannte Klassifikationsverfahren sind Klassifikatoren mit Trennfunktion (beispielsweise lineare Klassifikatoren), Distanzklassifikatoren sowie statistische Klassifikatoren (Jain *et al.* [JKS95], Duda & Hart [DHS01]). Desweiteren gibt es die Möglichkeit, neuronale Netze zur Klassifikation zu nutzen (Bishop [Bis95]).

Diese Arbeit verwendet einen Nächster-Nachbar Klassifikator, ein typischer Distanzklassifikator, sowie statistische Klassifikatoren. Die in einer Trainingsphase aufgenommenen Kamerabilder dienen hierbei als Prototypen. Einem zu klassifizierenden Bild wird dann mit Hilfe der Klassifikationsmethoden ein Trainingsbild zugeordnet. Die Position dieses Trainingsbildes ist dann die gesuchte Roboterposition, womit die Lokalisierung abgeschlossen ist.

Problemfaktoren

Die beschriebene Lokalisierung mit visueller Sensorik beruht darauf, daß sich Bilder benachbarter Positionen ähnlich sehen. Dies ist aus verschiedenen Gründen oft nicht der Fall. So treten beispielsweise Schwierigkeiten auf, wenn der Roboter rotiert ist, wenn die Beleuchtung variiert oder wenn Teilbereiche der Bilder durch Personen oder Gegenstände verdeckt sind. In dieser Arbeit werden einige dieser Schwierigkeiten beschrieben sowie mögliche Lösungswege präsentiert. Ein besonderer Schwerpunkt liegt dabei auf der Behandlung von Beleuchtungsvarianzen.

Sektoren-Lokalisierung

Viele Lokalisierungsmethoden werden unzuverlässig, wenn die Sensordaten durch Hindernisse okkludiert sind. In einer natürlichen Umgebung sind aber Veränderungen der Umgebung durch Gegenstände oder umherlaufende Personen an der Tagesordnung.

In dieser Arbeit wird eine Methode präsentiert, die die Robustheit der Lokalisierung unter solchen Bedingungen vergrößert. Sie basiert auf der Aufspaltung der Sensordaten in viele kleine Datenmengen. Dazu werden die Panoramabilder in sich überlappende Sektorbilder unterteilt. Diese Sektorbilder werden einzeln zur Lokalisierung genutzt. Das hat den Vorteil, daß nicht-verdeckte Sektoren sehr gute Ergebnisse liefern. Allerdings können hierbei leicht Mehrdeutigkeiten auftreten, da ein Sektor deutlich weniger Information enthält

als ein globales Panoramabild. So kommt es häufig vor, daß ein Sektor einen Teil der Umgebung beinhaltet, dessen Erscheinungsform häufig auftritt.

Diese Ambiguitäten können durch Betrachten des visuellen Kontextes des Sektorbildes verringert werden, welcher durch die Reihenfolge weiterer Sektoren desselben Panoramabildes repräsentiert wird, vorausgesetzt, einige dieser Sektoren sind ebenfalls nicht-okkludiert (siehe auch Paletta, Frintrop & Hertzberg [PFH01b]). Einen anderen Ansatz zur robusten Lokalisierung bezüglich Okklusionen präsentieren Jogan und Leonardis in [JL00].

1.2 Wissenschaftlicher Beitrag

Omnidirektionale Kameras haben in letzter Zeit zunehmend an Interesse gewonnen. Auch im Bereich der Roboterlokalisierung wurden sie bereits erfolgreich eingesetzt ([JL00], [AIYT98]). Diese Arbeit legt einen Schwerpunkt darauf, die Robustheit dieses Ansatzes gegenüber verschiedenen Störquellen des Sensors zu untersuchen und durch geeignete Methoden zu verbessern. Insbesondere wurde die Robustheit gegenüber Beleuchtungsvarianzen getestet und verbessert. Weiterhin wurde das Verfahren der Sektor-Lokalisierung entwickelt, eine Methode, die einen vielversprechenden Ansatz zur Behandlung starker Okklusionen darstellt. Auszüge dieser Arbeit wurden bereits veröffentlicht in ([PFH01a, PFH01b, PFH01c]).

1.3 Übersicht

Diese Arbeit gliedert sich in zwei Teile. Im ersten Teil werden die verwendeten Methoden vorgestellt, im zweiten Teil werden Experimente präsentiert anhand derer diese Methoden erprobt wurden.

Teil I: Methoden

Kapitel 2 befaßt sich mit einer bekannten Datenkompressionsmethode, genannt PCA (Principal Component Analysis). Es wird beschrieben, wie der Eigenraum der Bilddaten erzeugt wird, wie die Bilddaten in diesen Eigenraum projiziert werden und welche Komplexität dieses Verfahren hat.

Kapitel 3 beschreibt verschiedene Klassifikationsverfahren. Zunächst wird Appearance-based Matching als Grundlage der Klassifikation beschrieben, dann wird auf einige Klassifikatoren näher eingegangen.

Kapitel 4 faßt das gesamte Vorgehen bei der Lokalisierung mit den kompletten Panoramabildern zusammen. Diese Methode wird Vollbild-Lokalisierung genannt, da sie im Gegensatz zu der Sektoren-Lokalisierung des folgenden Kapitels steht.

Kapitel 5 stellt die Sektoren-Lokalisierung vor, ein Verfahren zur Behandlung von starken Okklusionen in den Bilddaten. Es beschreibt die Aufteilung der Bilder in einzelne Sektoren, die Klassifikation dieser Sektoren und die Berücksichtigung des lokalen visuellen Kontextes.

Teil II: Experimente

Kapitel 6 beschreibt das Arbeitsszenario der Experimente. Der Roboter, die Kamera, die Panoramabilder sowie die Büroumgebung werden vorgestellt; außerdem wird beschrieben, wie die Ergebnisse der folgenden Kapitel zu interpretieren sind.

Kapitel 7 zeigt die Ergebnisse der Vollbild-Lokalisierung. In verschiedenen Testszenarien werden die Möglichkeiten und die Grenzen dieser Methode aufgezeigt. Ein besonderer Schwerpunkt liegt dabei auf der Behandlung von Beleuchtungsvarianzen.

Kapitel 8 präsentiert die Ergebnisse der Sektoren-Lokalisierung. Ein besonderer Schwerpunkt liegt hier auf der Behandlung von starken Okklusionen in den Bilddaten.

Kapitel 9 vergleicht die beiden Lokalisierungsmethoden und stellt Vor- und Nachteile der Ansätze gegenüber.

Kapitel 10 liefert schließlich eine kurze Zusammenfassung sowie einen Ausblick über mögliche Verbesserungen und Erweiterungen der vorgestellten Verfahren.

Teil I

Methoden

Kapitel 2

Datenkompression mit PCA

Zur Roboterlokalisierung werden meist keine visuellen Sensoren verwendet, sondern beispielsweise Laser- oder Sonarsensoren. Der vielleicht wichtigste Grund dafür ist die große Informationsmenge der Bilddaten. Der Umgang mit diesen erfordert gewöhnlich große Speicher- und Zeitressourcen, so daß Echtzeitberechnungen während des Roboterbetriebs nicht möglich sind. Wenn Bilddaten unter Echtzeitanforderungen eingesetzt werden sollen, ist es zunächst dringend notwendig, die Datenmenge stark zu reduzieren. Ziel einer guten Datenkompression ist eine hohe Kompressionsrate bei gleichzeitig hohem Informationserhalt. Eine bekannte Methode zur Bilddatenkompression basiert auf der PCA (Principal Component Analysis), auch bekannt als Karhunen-Loève Transformation.

Das folgende Kapitel erklärt zunächst in Abschnitt 2.1 das generelle Vorgehen bei PCA. Abschnitt 2.2 beschreibt im einzelnen die mathematischen Berechnungen, die dem Verfahren zugrunde liegen. In Abschnitt 2.3 wird schließlich die Komplexität von PCA aufgeschlüsselt, um zu zeigen, daß das Verfahren sich zum Einsatz für Echtzeitanwendungen eignet.

2.1 PCA

Principal Component Analysis ist ein bekanntes Verfahren zur Bilddatenkompression, das erstmalig 1901 von Pearson [Pea01] in biologischem Kontext eingeführt wurde und später von Hotelling [Hot33], Karhunen [Kar47] und Loève [Loe63] im Rahmen der Wahrscheinlichkeitstheorie wiederaufgegriffen wurde. Im Bereich der Bildverarbeitung wird PCA häufig zur Objekterkennung (z.B. Buchstaben- und Gesichtserkennung) eingesetzt (Murase & Nayar [MN95], Turk & Pentland [TP91] und Paletta [Pal99]). In der Literatur finden sich weitere Kompressionsverfahren als Alternative zu PCA, wie z.B. Independent Components Analysis (ICA) [Com94, BS95], Fisher's Linear Discriminant [Fis36, BHK97] oder Evolutionary Projection Pursuit [LW98].

Zum Verständnis von PCA betrachten wir zunächst, wie ein Grauwert-Bild mathematisch dargestellt werden kann. Ein Bild bestehe aus n Pixeln, jedes repräsentiert durch eine Zahl, die den Helligkeitswert des Pixels angibt. Indem das Bild nun zeilenweise ausgele-

sen wird, ergibt sich ein n -dimensionaler Vektor (x_1, \dots, x_n) . Somit kann man das Bild als Punkt in einem n -dimensionalen Raum betrachten. Da n gewöhnlich sehr groß ist, wird eine Transformation in einen Raum mit weniger Dimensionen durchgeführt. Angenommen, man würde dazu beliebige Komponenten des Bildvektors entfernen, beispielsweise den Vektor hinter dem k -ten Element abschneiden. Das würde zu einem quadratischen Fehler führen, der der Summe der Varianzen der entfernten Komponenten entspricht [Pal99].

Bei Principal Component Analysis (PCA) wird eine Koordinatentransformation vom Bildraum in einen Untervektorraum durchgeführt, in dem der quadratische Fehler bei Abschneiden des resultierenden Vektors minimal ist. Diese Eigenschaft bietet der Eigenraum der Kovarianzmatrix der Bilddaten. Dieser wird von den Eigenvektoren der Kovarianzmatrix aufgespannt. Als Bilddaten dienen dabei eine Menge von Trainingsbildern, die in einer Vorbereitungsphase an repräsentativen Stellen der zu lernenden Umgebung aufgenommen werden (siehe Kapitel 4: Vollbild-Lokalisierung). Der Eigenraum entspricht somit einem neuen rotierten Koordinatensystem, das bestmöglichst an die Eigenschaften der Datenmenge angepaßt ist. Abbildung 2.1 verdeutlicht die Vorgehensweise mit PCA. Ein Bild $x = (x_1, \dots, x_n)$ wird durch PCA auf den Vektor im Eigenraum $g = (g_1, \dots, g_k)$ reduziert, wobei k die Dimension des Eigenraums darstellt.

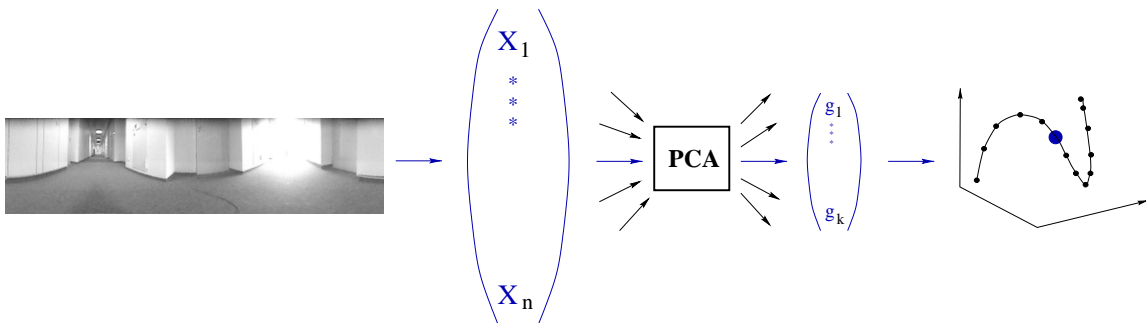


Abbildung 2.1: Datenkompression mittels PCA

2.1.1 Eigenschaften von Eigenvektoren

Um zu verdeutlichen, warum bei Abschneiden eines Vektors im Eigenraum der quadratische Fehler minimal wird, betrachten wir die Eigenvektoren genauer. Die Eigenvektoren sind in Richtung der höchsten Varianzen der Datenmenge ausgerichtet. Jedem Eigenvektor ist ein Eigenwert zugeordnet. Die Größe dieses Eigenwertes entspricht der Varianz der Daten in der Richtung des zugehörigen Eigenvektors. Wie bereits erwähnt, ist der quadratische Fehler des neuen Datenvektors im Eigenraum geringer, wenn die Varianzen der entfernten Komponenten klein sind. Die Signifikanz der Eigenvektoren nimmt also mit der Größe der zugehörigen Eigenwerte zu, so daß lediglich die Eigenvektoren mit den größten Eigenwerten gewählt werden müssen. PCA liefert als Ergebnis die Eigenvektoren geordnet bezüglich der

Größe ihrer Eigenwerte, so daß durch ein Abschneiden eines Vektors im Eigenraum genau die relevantesten Daten erhalten werden.

Die mathematische Berechnung der Eigenvektoren folgt in Abschnitt 2.2.1, hier sei jetzt nur eine Veranschaulichung des Verfahrens gegeben. In Abbildung 2.2 sieht man eine Datenmenge, deren Elemente den Klassen C und D zugeordnet sind, die Eigenvektoren sind e_1 und e_2 . Auf der rechten Seite werden die Daten nur bezüglich e_1 dargestellt.

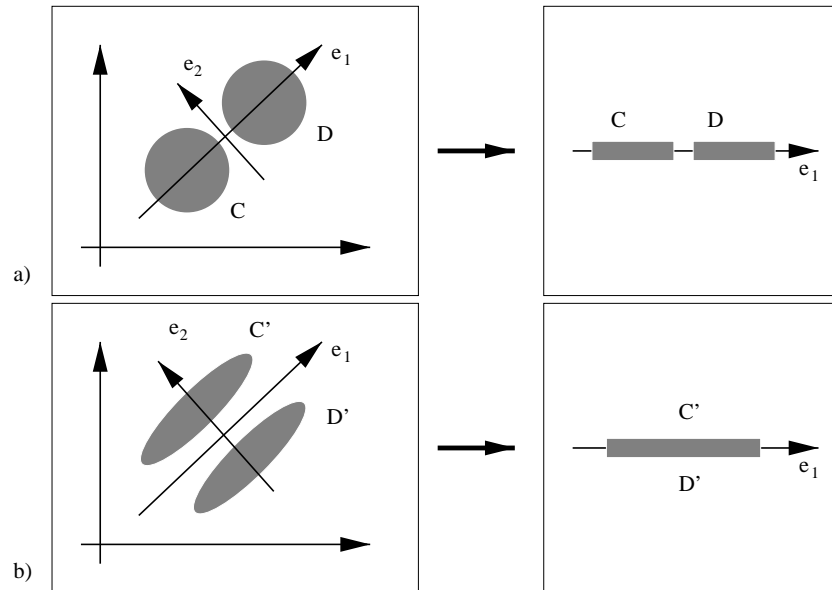


Abbildung 2.2: Die Eigenvektoren e_1 und e_2 der Datenmengen C und D , sowie C' und D' . In (a) wird die Struktur der Daten erfaßt, so daß Klassifikation möglich ist, in (b) dagegen nicht.

Wie man sieht, sind die Eigenvektoren bezüglich der größten Varianzen der Datenmenge ausgerichtet. Die Größe der Varianz entspricht dabei der Größe des zugehörigen Eigenwertes. Dies sei hier graphisch durch die Länge der Vektoren e_1 und e_2 veranschaulicht. Man beachte jedoch, daß sich tatsächlich nicht die Länge der Eigenvektoren unterscheidet, sondern die Größe der Eigenwerte.

Auf der rechten Seite von Abbildung 2.2 sieht man nun eine Darstellung der Datenmengen C und D nur bezüglich des Eigenvektors e_1 . Offensichtlich reicht in a) dieser eine Vektor aus, um die Struktur der Daten zu erfassen. Eine Zuordnung eines Testelementes zu C und D ist leicht möglich. In b) dagegen ist dies wegen einer Verzerrung der Datencluster nicht gegeben. Obwohl der Eigenvektor e_1 den größeren Eigenwert hat, kann er die Struktur der Daten nicht ausreichend erfassen. Eine erfolgreiche Zuordnung zu C und D ist so nicht möglich. Natürlich wäre durch Hinzunehmen des zweiten Eigenvektors e_2 dieser Fehler wieder behoben, allerdings ginge damit auch der Vorteil der Datenreduktion verloren. Das bringt uns zu der Frage, wieviele Eigenvektoren gebraucht werden, um eine erfolgreiche Klassifikation zu gewährleisten.

2.1.2 Auswahl der Eigenvektoren

Wie schon angedeutet, erhielte man durch Verwendung aller Eigenvektoren eine verlustfreie Darstellung der Daten im Eigenraum. Damit wäre aber die Datenmenge noch so groß wie zuvor, d.h. es wäre keine Datenkompression erreicht worden. Um eine Datenkompression zu erhalten, werden also nur k Eigenvektoren verwendet.

Wie groß die Anzahl k der zu verwendenden Eigenvektoren tatsächlich ist, kommt auf die Aufgabenstellung an. Zur Kompression der Bilder mit anschließender Rekonstruktion, so daß keine visuell sichtbare Verschlechterung der Bildqualität auftritt, wird eine große Menge an Eigenvektoren benötigt. Je genauer die Rekonstruktion ausfallen soll, desto mehr Eigenvektoren werden benötigt. Beispielsweise wurden in [KBVM99] Experimente gemacht, bei denen 90 Eigenvektoren benötigt wurden, um bei einer Datenbank aus 2000 Bildern eine 80%ige Rekonstruktion der Bilder zu erreichen.

Nun sind wir allerdings nicht an einer Rekonstruktion der Bilder interessiert, sondern nur an einer Erhaltung der wichtigsten Bildcharakteristika, die zur Klassifikation notwendig sind. Dafür ist es ausreichend, eine kleine Anzahl von Eigenvektoren ($k \ll n$) zu wählen [MN95]. Ein Ansatz ist es, k so zu bestimmen, daß die ersten k Eigenvektoren die wesentlichen Variationen der Bildmenge erfassen, so daß gilt:

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^n \lambda_i} \geq T, \quad (2.1)$$

wobei λ_i der zu dem i -ten Eigenvektor gehörige Eigenwert ist und T ein Schwellwert, der nahe bei 1 liegt. Experimente haben gezeigt, daß in unserem Anwendungsbereich $5 \leq k \leq 20$ ausreichend ist. So haben z.B. Murase & Nayar in [MN95] 20 Objekte aus verschiedenen Blickwinkeln aufgenommen, was eine Bildmenge von 1440 Bildern von je $128 \times 128 = 16\,384$ Pixeln ergab. Es zeigte sich, daß hierbei ein 20-dimensionaler Eigenraum ausreichte, um die Bilder zu klassifizieren. In unseren Experimenten wurden Bilder der Größe $360 \times 200 = 72\,000$ Pixel in einen 10-dimensionalen Eigenraum projiziert (siehe Teil II: Experimente). Da k der Größe eines Bildvektors im Eigenraum entspricht, lassen sich also Bilder mit 72 000 Pixeln auf Vektoren mit 10 Komponenten reduzieren.

2.2 Der Eigenraum

2.2.1 Erzeugen des Eigenraums

Um den Eigenraum der Trainingsbildmenge zu erzeugen, müssen nun also die Eigenvektoren berechnet werden. Dazu gehen wir folgendermaßen vor:

- Normalisierung der Bilder,

- Subtraktion des Durchschnittsbildes c und
- Auswahl der Eigenvektoren.

Normalisierung

Um zu gewährleisten, daß die Bilderkennung nicht durch Variationen der Beleuchtungsintensität oder durch Ungenauigkeiten beim Aufnehmen der Bilder beeinflusst wird, werden die Bilder zunächst normalisiert. Ein aufgenommenes Kamerabild $\hat{x} = (\hat{x}_1, \dots, \hat{x}_n)$ wird zu dem normalisierten Bild $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)$ transformiert:

$$\bar{x}_i = \frac{\hat{x}_i}{\|\hat{x}\|} \quad \text{mit} \quad i = 1, \dots, n, \quad (2.2)$$

dabei ist mit $\|\hat{x}\|$ die Norm von \hat{x} gemeint, d.h.

$$\|\hat{x}\| = \sqrt{\sum_{i=1}^n \hat{x}_i^2}. \quad (2.3)$$

Die normalisierten Bildvektoren sind Einheitsvektoren der Länge $\|\bar{x}\| = 1$. Man beachte, daß die Normalisierung nur solche Beleuchtungsvarianzen betrifft, die die gesamte Bildmenge betreffen. Beleuchtungsunterschiede bei einzelnen Bildern können so nicht ausgeglichen werden.

Subtraktion des Durchschnittsbildes

Wie in Abschnitt 2.1 beschrieben, richtet PCA die Achsen des neuen Koordinatensystems nach den höchsten Varianzen der Datenmenge aus. Damit der Eigenvektor mit dem größten Eigenwert auch wirklich die Dimension der größten Datenvarianz repräsentiert, muß der Ursprung des Koordinatensystems im Mittelpunkt der Datenmenge plziert sein. Um das Koordinatensystem dorthin zu verschieben, wird das Durchschnittsbild der Bildmenge berechnet und von jedem einzelnen Bild abgezogen. Das Durchschnittsbild c berechnet sich durch

$$c_i = \frac{\sum_{j=1}^m \bar{x}_{ij}}{m}, \quad (2.4)$$

mit $m =$ Anzahl der Bilder und $i = 1, \dots, n$. Nachdem das Durchschnittsbild von jedem Bild subtrahiert wurde, erhält man eine neue Bildmenge, die in Spaltenschreibweise die Bildmatrix X darstellt. Hierbei handelt es sich um eine $(n \times m)$ Matrix, mit $n =$ Anzahl der Pixel pro Bild und $m =$ Anzahl der Bilder:

$$X = \begin{pmatrix} \bar{x}_{11} - c_1 & \dots & \bar{x}_{1m} - c_1 \\ \vdots & & \vdots \\ \bar{x}_{n1} - c_n & \dots & \bar{x}_{nm} - c_n \end{pmatrix} \quad (2.5)$$

Bestimmung der Eigenvektoren

Im folgenden werden die Eigenvektoren der Kovarianzmatrix von X berechnet. Die Kovarianzmatrix Q ist definiert durch:

$$Q = XX^T. \quad (2.6)$$

Bei Q handelt es sich um eine symmetrische $(n \times n)$ -Matrix. In der Diagonalen von Q befinden sich die Varianzen der einzelnen Pixel, die restlichen Stellen enthalten die Kovarianzen zwischen je zwei Pixeln. Die Varianz q_{ii} gibt an, wie sehr der Wert von Pixelposition i in den einzelnen Bildern der Bildmenge variiert. Die Kovarianz q_{ij} verdeutlicht, wie die Pixelpositionen i und j in der Bildmenge in Verbindung stehen. Zum besseren Verständnis betrachten wir genauer, wie sich Q aus den Spalten und Zeilen von X und X^T berechnet¹:

$$\begin{pmatrix} x_{11} & \dots & \mathbf{x}_{1j} & \dots & x_{1n} \\ \vdots & & \vdots & & \vdots \\ x_{m1} & \dots & \mathbf{x}_{mj} & \dots & x_{mn} \end{pmatrix} \begin{pmatrix} x_{11} & \dots & x_{1m} \\ \vdots & & \vdots \\ \mathbf{x}_{i1} & \dots & \mathbf{x}_{im} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{nm} \end{pmatrix} \begin{pmatrix} q_{11} & \dots & \dots & q_{1n} \\ \vdots & & & \vdots \\ & & \mathbf{q}_{ij} & \\ \vdots & & & \vdots \\ q_{n1} & \dots & \dots & q_{nn} \end{pmatrix}$$

Um q_{ij} zu berechnen, werden also Zeile i von Matrix X und Spalte j von Matrix X^T benötigt. Spalte j von X^T entspricht dabei der Zeile j von X , d.h. es werden Zeile i und Zeile j von X benötigt. Da die Zeilen von X die Pixelposition angeben, fließen also Pixel i und Pixel j in die Berechnung von q_{ij} ein. Zu dieser Kovarianzmatrix Q werden nun die Eigenvektoren gesucht. Die Beziehung zwischen einer Matrix Q und deren Eigenwerten λ_i und Eigenvektoren e_i ist gegeben durch die Eigenwertgleichung

$$\lambda_i e_i = Q e_i \quad \text{oder} \quad (Q - \lambda_i I) e_i = 0, \quad (2.7)$$

¹Darstellung zur Multiplikation von Matrizen nach [Fis86]

wobei I die Einheitsmatrix ist. Die Lösungen zu diesem linearen Gleichungssystem aus n Gleichungen erhält man durch Bestimmung des charakteristischen Polynoms P_Q :

$$P_Q = \det(Q - \lambda I) = \begin{vmatrix} q_{11} - \lambda & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} - \lambda & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ q_{n1} & \dots & \dots & q_{nn} - \lambda \end{vmatrix} \quad (2.8)$$

Die Nullstellen λ_i des charakteristischen Polynoms sind jetzt genau die Eigenwerte von Q und die zugehörigen Vektoren e_i die Eigenvektoren [Kow79]. Damit ist die Bestimmung der Eigenwerte und Eigenvektoren auf das fundamentale algebraische Problem der Bestimmung der Nullstellen eines Polynoms vom Grad n zurückgeführt. Dieses Problem erweist sich für große n , wie in unserem Fall, als sehr kompliziert. In der Praxis verwendet man stattdessen andere Methoden. Ein häufig verwendeter Ansatz ist die "Singular Value Decomposition" (SVD) [PFTV88, Web99]. Andere Methoden sind z.B. die "Conjugate Gradient Methode" und der "Spatial Temporal Adaptive (STA) Algorithmus"; sie sind in [MN95] beschrieben.

SVD beruht darauf, daß eine Matrix X geschrieben werden kann als $X = E\Sigma E^T$. Dabei stehen in E die Eigenvektoren der Kovarianzmatrix $Q = XX^T$ von X und $\Sigma = [\text{diag}(\sqrt{\lambda_i})]$ ist eine Diagonalmatrix, die in der Diagonalen die Singulärwerte $\sqrt{\lambda_i}$ enthält. Die λ_i sind hierbei genau die Eigenwerte der Kovarianzmatrix Q . Abbildung 2.3 zeigt Input und Output der Singular Value Decomposition.

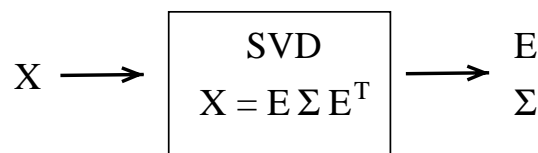


Abbildung 2.3: Die Singular Value Decomposition bildet die Eingangsmatrix X auf die Eigenvektormatrix E und die Matrix Σ der Singulärwerte ab.

Das Verfahren liefert also sowohl die Eigenwerte λ_i als auch die zugehörigen Eigenvektoren e_i von Q und hat den großen Vorteil, daß die Kovarianzmatrix Q nicht explizit berechnet werden muß. Da Q eine sehr große Matrix ist ($(n \times n)$ mit $n = \text{Anzahl der Pixel}$), wird so viel Zeit und Speicherplatz gespart. Als Ergebnis von SVD sind die Eigenvektoren in E jetzt schon in absteigender Reihenfolge bezüglich ihrer Eigenwerte sortiert, so daß nur die ersten k Eigenvektoren gewählt werden müssen.

2.2.2 Projektion in den Eigenraum

Nachdem die Eigenvektoren bestimmt sind, werden die Bilder in den Eigenraum projiziert. Dazu müssen sie bezüglich der neuen Basisvektoren dargestellt werden. Das geschieht durch

Multiplikation mit den Eigenvektoren, die in der Matrix E dargestellt werden. Zuvor muß aber wieder das Durchschnittsbild subtrahiert werden. Das ergibt für das i -te Trainingsbild die Formel:

$$g_i = E^T(X_i - c) \quad (2.9)$$

das ist in Matrixschreibweise:

$$\begin{pmatrix} e_{11} & \dots & e_{1n} \\ \vdots & & \vdots \\ e_{k1} & \dots & e_{kn} \end{pmatrix} \begin{pmatrix} x_{i1} - c \\ \vdots \\ x_{in} - c \end{pmatrix} = \begin{pmatrix} g_1 \\ \vdots \\ g_k \end{pmatrix}$$

PCA liefert also als Ergebnis:

- k Eigenvektoren mit je n Komponenten und
- m Bildvektoren im Eigenraum mit je k Komponenten

Man beachte, daß die ursprünglichen Bilder nun nicht mehr benötigt werden und gelöscht werden können.

2.2.3 Korrelation im Eigenraum

In diesem Abschnitt wird gezeigt, daß der Abstand zweier Punkte im Eigenraum ein Maß für die Korrelation der zugehörigen Bilder ist. Mit anderen Worten: Ähnliche Bilder entsprechen im Eigenraum benachbarten Punkten.

Zunächst eine Veranschaulichung dieses Sachverhalts. Abbildung 2.4 zeigt 14 Momentaufnahmen eines Übergangs einer schwarzen zu einer weißen Fläche. Offensichtlich sind zwei aufeinanderfolgende Bilder sehr ähnlich. Abbildung 2.5 zeigt die Projektionen der Bilder im Eigenraum. Es ergibt sich eine Kurve, auf der die Positionen der Aufnahmen durch Punkte markiert sind. Benachbarte (und damit ähnliche) Bilder entsprechen benachbarten Punkten auf der Kurve. Aufgrund der Einfachheit der Bilder sind hier sogar drei Dimensionen (drei Eigenvektoren) ausreichend, um die Struktur der Bilder zu erfassen.

Da der Abstand im Eigenraum ein Maß für die Ähnlichkeit der Bilder ist, läßt sich ein Testbild klassifizieren, indem es in den Eigenraum projiziert wird und dem nächsten Nachbarn zugeordnet wird (siehe Kapitel 3: Klassifikation). Abbildung 2.6 zeigt ein solches Testbild und seine Darstellung im Eigenraum. Das Testbild wurde so gewählt, daß die

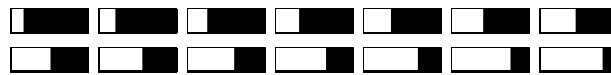


Abbildung 2.4: Momentaufnahmen eines Übergangs einer schwarzen zu einer weißen Fläche

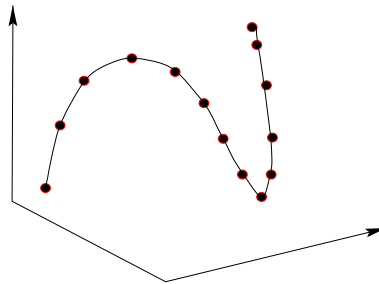


Abbildung 2.5: Darstellung der Bilder im Eigenraum

Grenze zwischen weißer und schwarzer Fläche an einer Stelle verläuft, die bei den Trainingsbildern nicht vorkam. Der dicke Punkt in der Eigenraumgrafik rechts bezeichnet die Projektion des Testbildes. Er liegt zwischen zwei Trainingsbildprojektionen. Im Klassifikationsschritt würde das Testbild dem nähergelegenen dieser Punkte zugeordnet.

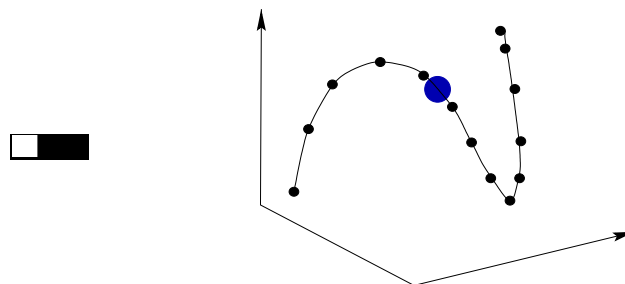


Abbildung 2.6: Ein Testbild und seine Projektion im Eigenraum

Formalisierung

Nun zur formalen Beschreibung dieser Eigenschaften. Seien x und y zwei Trainingsbilder, die zur Erzeugung des Eigenraums verwendet werden. Seien die Punkte g und h die Projektionen der Bilder in den Eigenraum. Jedes der Bilder kann nun durch seine Projektion ausgedrückt werden:

$$x = \sum_{i=1}^n g_i e_i + c. \quad (2.10)$$

Da unser Eigenraum nur k -dimensional ist, kann x durch die ersten k Summanden der obigen Gleichung approximiert werden:

$$x \approx \sum_{i=1}^k g_i e_i + c. \quad (2.11)$$

Dies ist möglich, da die genutzten Eigenvektoren genau die mit den größten Eigenwerten sind, welche wiederum die signifikantesten Variationen in der Bildmenge beschreiben. Als Resultat der Normalisierung (Abschnitt 2.2.1), sind x und y Einheitsvektoren. Die Ähnlichkeit der beiden Bilder kann nun durch Bestimmung des quadratischen Abstands (sum-of-squared-difference (SSD)) bestimmt werden:

$$\|x - y\|^2 = \sum_{i=1}^n (x_i - y_i)^2. \quad (2.12)$$

Diese Methode wird häufig im Bereich der Bildverarbeitung angewandt, z.B. zur Berechnung der Korrelation zweier Bildmuster [GW92]. Der SSD-Abstand steht in inverser Beziehung zur Korrelation, da:

$$\|x - y\|^2 = (x - y)^T (x - y) \quad (2.13)$$

$$= 2 - 2x^T y. \quad (2.14)$$

Dabei bezeichnet $x^T y$ die Korrelation. Folglich bedeutet ein Minimieren des Abstands ein Maximieren der Korrelation, d.h. ein Maximieren der Ähnlichkeit.

Nun wieder zum Ziel dieses Abschnitts. Wir wollten zeigen, daß das Abstandsmaß auch im Eigenraum ein adäquates Maß für die Ähnlichkeit der Bilder ist. Dazu beschreiben wir die Bilder durch ihre Darstellungen im Eigenraum g und h :

$$\|x - y\|^2 \approx \left\| \sum_{i=1}^k g_i e_i - \sum_{i=1}^k h_i e_i \right\|^2 \quad (2.15)$$

$$= \left\| \sum_{i=1}^k (g_i - h_i) e_i \right\|^2 \quad (2.16)$$

$$= \sum_{i=1}^k \sum_{j=1}^k e_i^T e_j \times (g_i - h_j)^2 \quad (2.17)$$

$$= \|g - h\|^2. \quad (2.18)$$

Die letzte Umformung gilt, da die Eigenvektoren orthogonal zueinander sind, d.h. es gilt $e_i^T e_j = 1$ wenn $i = j$ und 0 sonst. Es ergibt sich also:

$$\|x - y\|^2 \approx \|g - h\|^2 \quad (2.19)$$

Diese Formel impliziert, daß das Quadrat des Euklid'schen Abstands zwischen den Punkten g und h eine Approximation der Ähnlichkeit der Bilder x und y ist. Mit anderen Worten, je näher zwei Punkte im Eigenraum beieinander liegen, desto ähnlicher sind sich die zugehörigen Bilder.

2.3 Komplexität

Am Anfang dieses Kapitels wurde erwähnt, daß Bilddaten aufgrund der hohen Komplexität bei ihrer Verarbeitung oft nicht zur Roboterlokalisierung verwendet werden. In diesem Abschnitt soll nun verdeutlicht werden, wie die Verwendung von PCA dazu beiträgt, die Komplexität einzuschränken. So wird sowohl das Speichern einer großen Menge an Bilddaten ermöglicht als auch Echtzeitberechnung während des Roboterbetriebs. Zur Analyse der Komplexität von PCA müssen zwei Bereiche unterschieden werden:

- Die Vorbereitungsphase
- Die Aktionsphase

In der Vorbereitungsphase wird der Eigenraum erzeugt und die Trainingsbilder hineinprojiziert. Besonders die Erzeugung des Eigenraums ist sehr zeitaufwendig, da sie Berechnungen mit sehr großen Matrizen beinhaltet. Da diese Berechnungen aber als Vorbereitung stattfinden und nur einmal durchgeführt werden müssen, um die Umgebung zu "lernen", beeinträchtigt dies das Verhalten während des Roboterbetriebs nicht.

In der Aktionsphase wird nur ein Bild in den Eigenraum projiziert sowie die Bildklassifikation durchgeführt. Diese Berechnungen können sehr schnell ausgeführt werden, wie im folgenden gezeigt wird.

2.3.1 Die Vorbereitungsphase

Während der Vorbereitungsphase "lernt" der Roboter seine Umgebung. Zuerst werden die Trainingsbilder aufgenommen, dann werden sie vorverarbeitet, beispielsweise in rechteckige Panoramabilder transformiert, und in den Eigenraum projiziert. Um die Komplexität dieser Phase zu berechnen, sind also folgende Schritte zu berücksichtigen:

- Vorverarbeitung
- Normalisierung
- Berechnung und Subtraktion des Durchschnittsbildes
- Erzeugung des Eigenraums mit SVD

- Projektion der Trainingsbilder

Die Vorverarbeitung der Bilder, die Normalisierung und die Berechnung sowie Subtraktion des Durchschnittsbildes sind jeweils in Zeit $O(mn)$ berechenbar, da jeweils n Pixel aus m Bildern nacheinander in Betracht gezogen werden. Die Eigenwerte und Eigenvektoren werden mit SVD in $O(n^3)$ berechnet [Dem98]. Ein Bild in den Eigenraum zu projizieren, benötigt $O(kn)$ Zeit, also ist die benötigte Zeit für die Projektion von m Trainingsbildern $O(mkn)$. Da gewöhnlich $m, k \ll n$ gilt, wird die gesamte Komplexität der Vorbereitungsphase demnach von der Eigenvektorbestimmung mit SVD dominiert und beträgt folglich $O(n^3)$.

2.3.2 Die Aktionsphase

In der Aktionsphase soll der Roboter ein aufgenommenes Bild klassifizieren. Dazu sind folgende Schritte vorzunehmen:

- Vorverarbeitung
- Normalisierung
- Subtraktion des Durchschnittsbildes
- Projektion des Testbildes
- Klassifikation des Testbildes

Es sind also fast genau dieselben Schritte wie bei der Vorbereitungsphase durchzuführen, nur ist diesmal lediglich ein Bild zu berücksichtigen, und die Berechnung des Durchschnittsbildes sowie die zeitaufwendige Eigenvektorbestimmung mit SVD fallen weg.

Vorverarbeitung, Normalisierung und Subtraktion des Durchschnittsbildes benötigen jeweils $O(n)$ Zeit. Die Projektion des Bildes in den Eigenraum benötigt $O(kn)$ Zeit. Die Klassifikation des Bildes mit einem Nächster-Nachbar Klassifikator (siehe Kapitel 3.2) benötigt für m Trainingsbilder $O(km)$ Zeit. Da gewöhnlich $m \ll n$ gilt, wird die Aktionsphase demnach von $O(kn)$ dominiert. In der vorliegenden Arbeit gilt $k = 10$ und $n = 72\,000$, woraus sich $kn = 720\,000$ ergibt. In der Praxis benötigt die gesamte Aktionsphase mit diesen Werten auf einem 333 MHz PC 0.15 Sekunden. Damit ist die Methode für Echtzeitberechnungen während des Roboterbetriebs geeignet.

Kapitel 3

Bildklassifikation

Nachdem im vorherigen Kapitel eine Bildkompressionsmethode präsentiert wurde, widmet sich dieses Kapitel der Bildklassifikation. Im Bezug auf Lokalisierung bedeutet dies: Zu klassifizierende Bilder stellen Punkte im Merkmalsraum dar, die einer von mehreren Klassen zugeordnet werden, die je eine Roboterposition repräsentieren. Kontinuierliche Positionswerte zwischen diesen diskreten Klassen können durch Interpolation gewonnen werden.

Bekannte Klassifikationsverfahren sind Klassifikatoren mit Trennfunktion (beispielsweise lineare Klassifikatoren), Distanzklassifikatoren, sowie statistische Klassifikatoren. Desweiteren gibt es die Möglichkeit, neuronale Netze zur Klassifikation zu nutzen. Für die Beschreibung weiterer Klassifikationsmethoden siehe Jain *et al.* [JKS95], Bishop [Bis95] oder Duda & Hart [DHS01].

Dieses Kapitel widmet sich in Abschnitt 3.1 zunächst der Methode des Appearance-based Matchings, welches es ermöglicht, die visuell verfügbare Information sinnvoll zur Klassifikation zu nutzen. Die Abschnitte 3.2 – 3.4 stellen dann verschiedene Klassifikationsverfahren vor, mit welchen die Klassenzuordnung basierend auf Appearance-based Matching vorgenommen werden kann. Abschnitt 3.2 präsentiert zunächst den Nächste Nachbar Klassifikator, ein typischer Distanzklassifikator. Abschnitt 3.3 stellt den Bayes'schen Klassifikator als grundlegendes Verfahren wahrscheinlichkeitbasierter Klassifikation vor und in 3.4 wird kurz auf die Verwendung Neuronaler Netze zur Klassifikation eingegangen. Eine ausführliche Beschreibung dieses Themas findet sich bei Bishop [Bis95].

3.1 Appearance-based Matching

Im Bereich der Bildverarbeitung gibt es eine Vielzahl von Methoden, um die gegebenen Bilddaten zur Klassifikation zu nutzen (Jain *et al.* [JKS95]). Viele davon basieren auf Merkmalsextraktion, d.h. es werden bestimmte Merkmale aus den Bilddaten herausgefiltert. So können z.B. Linien extrahiert werden, welche dann zum weiteren Gebrauch interpretiert werden.

Bei *Appearance-based Matching* (Murase & Nayar [MN95]) werden dagegen keine Merk-

male aus den Bildern extrahiert, sondern die "Erscheinung" (appearance) eines Bildes als Ganzes betrachtet. Repräsentative Bilder werden gespeichert, und die Erkennung eines Bildes geschieht durch Bild-Matching. Diese Methode eignet sich besonders für Anwendungen, in denen es schwierig ist, repräsentative Merkmale aus den Bilddaten zu extrahieren und adäquate Weltmodelle zu erzeugen. Dies ist beispielsweise bei komplexen Bildern wie solchen aus Büroumgebungen der Fall.

Murase und Nayar beschreiben in [MN95] wie Appearance-based Matching zur Objekterkennung verwendet wird und in [JL99, MKS97, MIII99, KBVM99] wird das Verfahren zur Roboterlokalisierung genutzt. Dabei wird eine direkte Relation zwischen den Beobachtungen und der Roboterposition hergestellt.

Biologische Motivation

Die Natur dient als Vorbild für diesen Ansatz: Verschiedene biologische Systeme wenden Appearance-based Matching an. So zeigen z.B. die Arbeiten von Möller *et al.* [MLR⁺01] sowie von Judd & Collett [JC98], daß Ameisen viele einzelne Schnappschüsse ihrer Umgebung speichern, um einen Weg von einem Start- zu einem Zielpunkt zu lernen. Auch Bienen (Dill, Wolf & Heisenberg [DWM93]) und sogar Säugetiere (Epstein & Kanwisher [EK98]) speichern einzelne Bilder ihrer Umgebung, um sie später wiederzuerkennen.

Appearance

Bei Appearance-based Matching geht es also nicht darum, welche Eigenschaften ein oder mehrere dreidimensionale Objekte haben, sondern wie sie in einem zweidimensionalen Bild in Erscheinung treten. Diese Erscheinung setzt sich zusammen aus der Form der Objekte, den Reflexionseigenschaften, der Position und Ausrichtung im Bild sowie der Beleuchtung. Form und Reflexion sind dabei sogenannte intrinsische Eigenschaften, d.h. sie sind unveränderlich für ein gegebenes Objekt. Die sogenannten extrinsischen Eigenschaften umfassen die variablen Eigenschaften, also die Beleuchtung, die Position und die Ausrichtung der Objekte im Bild.

Modellerstellung

Um ein umfassendes Modell eines Objekts oder einer Roboterposition zu erhalten, müssen nun die extrinsischen Eigenschaften variiert werden und je ein Bild für jede Variation aufgenommen werden. Diese Phase kann häufig automatisiert werden, indem beispielsweise ein Objekt auf einem Drehteller automatisch rotiert und in bestimmten Abständen aufgenommen wird. Ebenso kann die Beleuchtung Schritt für Schritt variiert werden (siehe Murase & Nayar [MN95]).

Bei Lokalisierung werden in der Regel die Position und Ausrichtung der Objekte im Bild nicht variiert, da davon ausgegangen wird, daß die Gegenstände einer Büroumgebung fest sind. Es wäre allerdings auch möglich, Gegenstände, die häufig bewegt werden (beispielsweise Stühle oder Türen) nicht als fest zu betrachten und ihre verschiedenen möglichen

Positionen zu berücksichtigen. Das wäre in der Praxis allerdings sehr zeit- und speicher-aufwendig, da von allen repräsentativen Positionen der beweglichen Gegenstände Bilder aufgenommen und gespeichert werden müßten.

Folglich ist es im Falle der Lokalisierung sinnvoller, nur die Beleuchtung zu variieren. Es wird also für jede Roboterposition ein Modell erzeugt. Diese Modelle der Roboterpositionen dienen später als Referenzpunkte bei der Klassifikation (siehe Kapitel 3.2 und Kapitel 3.3). Da wir eine omnidirektionale Kamera verwenden, müssen an einer Position allerdings nicht verschiedene Bilder in verschiedenen Richtungen aufgenommen werden, da ein Bild die gesamte Umgebung erfaßt. Dabei ergibt sich allerdings das Problem, daß die Kamera immer in der gleichen Richtung ausgerichtet sein muß und der Ansatz somit nicht rotationsinvariant ist. Eine Lösung zu dieser Problematik bietet die Sektoren-Lokalisierung, die in Kapitel 5 vorgestellt wird. Weitere Lösungsansätze präsentieren Jogan und Leonardis in [JL99].

3.2 Nächster-Nachbar Klassifikator

In diesem Kapitel wird der einfache aber dennoch häufig sehr effektive Nächster-Nachbar Klassifikator (Nearest Neighbor Classifier (Jain *et al.* [JKS95]) beschrieben. Klassifikation behandelt das Problem, Testdaten einer von mehreren klar definierten Klassen zuzuordnen. Jede dieser Klassen wird durch einen oder mehrere Prototypen repräsentiert. In unserem Fall sind die Klassen die Roboterpositionen und die Prototypen sind Vektoren im Eigenraum.

Angenommen, man habe einen k -dimensionalen Merkmalsraum. Die einzelnen Dimensionen können für Merkmale stehen, die in einer vorangehenden Merkmalsextraktion gewonnen werden. In unserem Fall aber entspricht der Merkmalsraum dem in Kapitel 2.2 erzeugten Eigenraum, der durch die Eigenvektoren der k größten Eigenwerte aufgespannt wird. Ein Bild kann also als Punkt in einem k -dimensionalen Merkmalsraum betrachtet werden.

Gegeben sei nun die Menge $\{\phi_1, \dots, \phi_m\}$ von m möglichen Klassen. In unserem Fall sind dies die möglichen Roboterpositionen. Es wird nun für jede Klasse ein Prototyp bestimmt. Ein Prototyp für Klasse ϕ_i werde beschrieben durch $P_i = (P_{i1}, \dots, P_{ik})$. Desweiteren sei $y = (y_1, \dots, y_k)$ die mathematische Darstellung eines zu klassifizierenden Testbildes.

Um das Testbild zu klassifizieren, wird nun der Abstand von y zu allen Prototypen berechnet. Dann wird es dem nächstgelegenen Prototypen zugeordnet. Die Bestimmung der Abstände zu Prototypen wird in Abbildung 3.1 an einem Beispiel eines zweidimensionalen Merkmalsraums verdeutlicht. Die Prototypen $P_1 - P_4$ und der Testvektor y werden bezüglich der Merkmalsvektoren e_1 und e_2 dargestellt. Wenn zuvor PCA zur Datenkompression verwendet wurde, sind e_1 und e_2 Eigenvektoren. Als Abstandsmaß wird hier der Euklid'sche Abstand verwendet. Es ist auch möglich, die einzelnen Merkmale verschieden zu gewichten. Das ist allerdings nur sinnvoll, wenn bestimmte Merkmale extrahiert wurden, denen unterschiedliche Wichtigkeit zugeordnet wird, was bei Appearance-based Matching nicht der Fall ist. Der Abstand von Testbild y zu Prototyp i wird also berechnet durch:

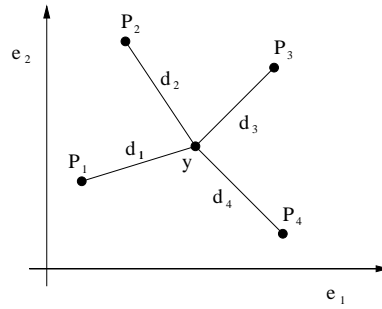


Abbildung 3.1: Berechnung der Abstände des Testvektors y zu den 4 Prototypen $P_1 - P_4$

$$d_i = \sqrt{\sum_{j=1}^k (y_j - P_{ij})^2}. \quad (3.1)$$

Das Testbild wird nun der Klasse ϕ_r zugeordnet, zu deren Prototyp es den geringsten Abstand hat:

$$d_r = \min_{i=1}^m [d_i], \quad (3.2)$$

wobei m die Anzahl der Prototypen ist. Da die Klasse r die Roboterposition ϕ_r repräsentiert, wird das Testbild y letztendlich der Position ϕ_r zugeordnet.

Bestimmung des Prototyps

In einfachen Fällen mag es einen offensichtlichen Prototypen einer Klasse geben, aber häufig gehören in der Praxis viele Objekte zu einer Klasse. In solchen Fällen gibt es zwei allgemeine Ansätze, Prototypen zu bestimmen:

1. Wähle den Mittelpunkt jedes Datenclusters als Prototypen dieser Klasse.
2. Wähle mehrere oder alle Objekte der Klasse als Prototypen.

In Fall 2. gehören zu einer Klasse jeweils mehrere Prototypen. Welcher dieser Fälle angewandt werden sollte, hängt davon ab, wie die Daten im Merkmalsraum verteilt sind. Wenn die Daten einer Klasse gebündelt auftreten, kann diese durch nur einen Prototypen repräsentiert werden (Fall 1.). Wenn die Daten anders verteilt sind, müssen mehrere Prototypen gewählt werden, um eine erfolgreiche Klassifikation zu gewährleisten (Fall 2.). Abbildung 3.2 zeigt Beispiele für diese beiden Fälle. In (a) sind Daten dargestellt, die sich zur Repräsentation durch einen Prototypen in der Mitte jedes Datenclusters eignen. In

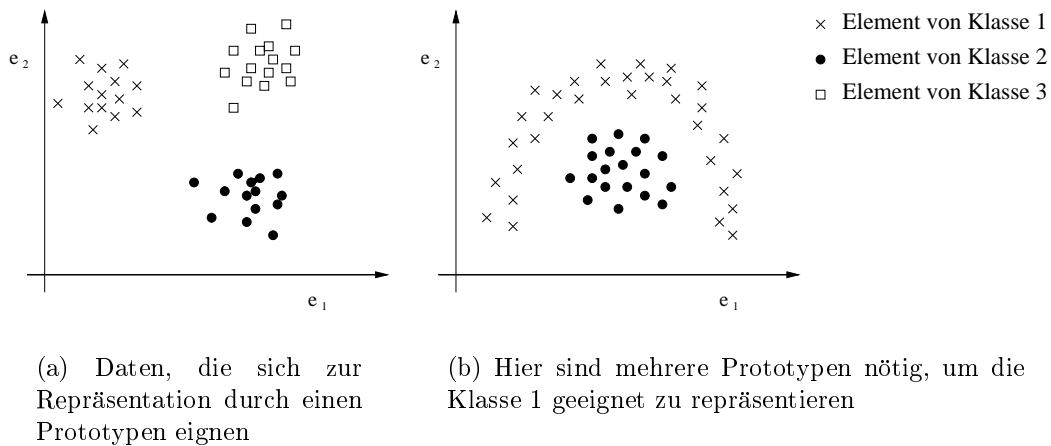


Abbildung 3.2: Verschiedene Verteilungen der Daten im Merkmalsraum

(b) treten die Elemente der Klasse 1 nicht gebündelt auf; hier sind mehrere Prototypen notwendig, um die Klasse geeignet zu repräsentieren.

Im Fall der Lokalisierung werden Projektionen von Trainingsbildern im Eigenraum als Prototypen gewählt. Wenn mehrere Trainingsbilder für eine Position verwendet werden, tritt obige Problematik auf. Dies ist z.B. dann der Fall, wenn sowohl Tag- als auch Nachtbilder als Trainingsdaten verwendet werden, um eine robustere Lokalisierung zu gewährleisten (siehe Kapitel 4.4). Hier ist es sinnvoll, Fall 2. anzuwenden und mehrere Prototypen zu bestimmen. Jede Roboterposition wird dann sowohl durch ein Tag- als auch durch ein Nachtbild als Prototyp repräsentiert. Einen Mittelwert der beiden Bilder zu wählen, ist hier nicht sinnvoll, weil ein Tagbild an Position a) einem Tagbild an Position b) in den meisten Fällen ähnlicher ist als einem Nachtbild an Position a). Wie in Kapitel 2.2.3 beschrieben, entspricht eine hohe Ähnlichkeit zweier Bilder einem geringen Abstand der zugehörigen Projektionen im Eigenraum. Es liegt also nahe, die Tag- und die Nachtbilder jeweils durch eigene Prototypen zu repräsentieren.

Mehrere Prototypen

Als Erweiterung des Verfahrens ist es möglich, zur Bestimmung der Klassenzugehörigkeit eines Testbilds nicht nur den nächstgelegenen Prototypen zu betrachten, sondern die d nächstgelegenen Prototypen (in der Literatur findet man dieses Verfahren unter dem Namen *k-Nearest Neighbor* (Fukunaga [Fuk90])). Dieser Ansatz kann zwei Vorteile bieten:

Der erste Vorteil besteht darin, daß ein Mehrheitsentscheid getroffen werden kann. Dies ist nur dann möglich, wenn eine Klasse durch mehrere Prototypen repräsentiert wird. Wenn also von den d nächsten Prototypen jeweils mehrere zu einer Klasse gehören, wird die Klasse gewählt, zu der die meisten Prototypen gehören. Hier ist es sinnvoll, einen gewichteten Mehrheitsentscheid zu treffen, so daß der Prototyp mit dem geringsten Abstand das höchste Gewicht erhält. Abbildung 3.3 zeigt ein Beispiel, bei der ein solches Vorgehen

vorteilhaft ist. Das Testelement y wird hier nicht der Klasse 2 zugeordnet, obwohl der nächstgelegene Prototyp zu Klasse 2 gehört. Da aber die drei Prototypen mit den nächstgeringsten Abständen zu Klasse 1 gehören, wird das Testelement per Mehrheitsentscheid Klasse 1 zugewiesen.

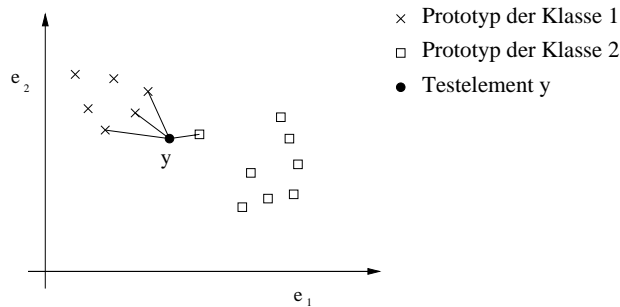


Abbildung 3.3: Klassifikation durch Mehrheitsentscheid

Der zweite Vorteil bezieht sich auf die Verwechslungsgefahr von Testdaten bei der Klassenzuordnung. Er kann auch dann genutzt werden, wenn es nur einen Prototypen pro Klasse gibt. Wenn der nächste und der zweitnächste Prototyp fast denselben Abstand zu einem Testbild haben, ist die Gefahr der Verwechslung sehr groß – vorausgesetzt, die Prototypen gehören verschiedenen Klassen an. Es kann deshalb sinnvoll sein, weitere Prototypen zu betrachten. Wenn die Abstände der ersten Prototypen sich kaum unterscheiden, kann folgendermaßen verfahren werden:

Entweder kann dieses Ergebnis verworfen werden, da keine eindeutige Klassifizierung möglich ist, oder es können beide Ergebnisse weitergeleitet werden, in der Hoffnung, die Zweideutigkeit später aufzulösen (beispielsweise durch Bayes'sches Update oder durch aktive Lokalisierung (siehe Maeda, Kuno & Shirai [MKS97])).

3.3 Bayes'sche Klassifikation

Da die Klassenzugehörigkeit meist nicht eindeutig feststellbar ist, kann stattdessen die Wahrscheinlichkeit der Zugehörigkeit zu einer Klasse bestimmt werden. Ergebnis der Klassifikation ist dann die Klasse mit der höchsten Wahrscheinlichkeit unter gegebenen Sensordaten. Der Bayes'sche Klassifikator stellt einen Idealfall statistischer Klassifikation dar, da die den Kategorien zugrundeliegende Wahrscheinlichkeitsstruktur als bekannt angenommen wird (Duda & Hart [DHS01]). Sie wird im Fall der Lokalisierung aus Trainingsdaten bestimmt.

Wiederum sei $\{\phi_1, \dots, \phi_m\}$ eine Menge von möglichen Klassen, die die möglichen Roboterpositionen repräsentieren, und y sei ein zu klassifizierendes Testbild. Die Wahrschein-

lichkeit, daß y zu einer Klasse ϕ_i gehört, ist die a-posteriori-Wahrscheinlichkeit $P(\phi_i|y)$ ¹. Um diese zu bestimmen, wende die Bayes'sche Regel an:

$$P(\phi_i|y) = \frac{p(y|\phi_i)P(\phi_i)}{p(y)} \quad (3.3)$$

mit

$$p(y) = \sum_{i=1}^m p(y|\phi_i)P(\phi_i). \quad (3.4)$$

Die Wahrscheinlichkeitsdichtefunktion $p(y|\phi_i)$ wird hierbei aus einer Menge von Beispielen mit festem ϕ_i geschätzt, die sich durch Beleuchtungs- und Positionsvarianzen geringen Ausmaßes unterscheiden. Als Modell für diese Funktion kann als erste Annäherung die Gauss'sche Dichtefunktion angenommen werden. Die a-priori-Wahrscheinlichkeit $P(\phi_i)$ wird als gleichverteilt angenommen, da jede Roboterposition als gleichwahrscheinlich angesehen wird. Es wäre auch möglich, häufiger besuchte Positionen mit höheren Wahrscheinlichkeiten zu belegen, was aber in dieser Arbeit nicht angewandt wurde. Der Normalisierungsfaktor $p(y)$ sorgt dafür, daß die Summe der einzelnen a-posteriori-Wahrscheinlichkeiten 1 ergibt. Ergebnis der Klassifikation des Testbildes y ist die Klasse ϕ_r , deren a-posteriori-Wahrscheinlichkeit $P(\phi_r|y)$ maximal ist:

$$\phi_r = \arg \max_{\phi_i} P(\phi_i|y). \quad (3.5)$$

Dieses Verfahren wird in Kapitel 5 zur Klassifikation bei der Sektoren-Lokalisierung angewendet. Weitere Beispiele von Bayes'scher Klassifikation zur Roboterlokalisierung finden sich bei Ishikawa *et al.* [IKH98] und bei Kröse *et al.* [KBVM99].

3.4 Neuronale Netze

Eine weitere häufig verwendete Klassifikationsmöglichkeit bieten Neuronale Netze (Bishop [Bis95]). Ihre Vorteile liegen in der Fähigkeit, den Merkmalsraum durch nicht-lineare Trennfunktionen in Klassen aufzuteilen. Diese Trennfunktionen werden durch Training des Netzes gewonnen. Abbildung 3.4 zeigt ein Beispiel eines Neuronalen Netzes mit einer verdeckten Schicht. Es enthält k Input-Knoten, in die das zu klassifizierende Bild $y = (y_1, \dots, y_k)$ eingegeben wird, und m Output-Knoten für die m möglichen Klassen $\{\phi_1, \dots, \phi_m\}$, die die Roboterpositionen repräsentieren.

Es gibt viele Variationen Neuronaler Netze. Zu den bekanntesten zählen Multilayer-Perceptrons (Rumelhart *et al.* [RHW86]). Diese teilen die Daten in Klassen ein, indem

¹Nach üblicher Notation (vgl. [DHS01]) bezeichne $P(\cdot)$ diskrete Wahrscheinlichkeiten und $p(\cdot)$ eine Wahrscheinlichkeitsverteilung

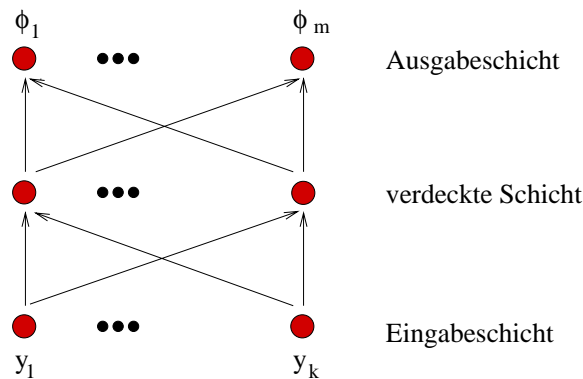


Abbildung 3.4: Neuronales Netz mit einer verdeckten Schicht

sie Hyperebenen durch den Eingaberaum legen. Abbildung 3.5 (a) stellt die Klasseneinteilung mit einem Multilayer-Perceptron in einem zweidimensionalen Merkmalsraum dar. Abbildung 3.5 (b) zeigt eine andere Möglichkeit der Klassenzuordnung durch die Verwendung von RBF-Netzen (Radial-Basis-Funktion-Netzwerke, Broomhead & Lowe [BL88]). Jede Klasse wird hier durch eine Kernfunktion repräsentiert, deren Zentrum im Mittelpunkt des Datenclusters liegt.

Repräsentative Anwendungen Neuronaler Netze zur Bildklassifikation finden sich bei Cassinis *et al.* [CGR95, CGR96] und Ishikawa *et al.* [IKH98].

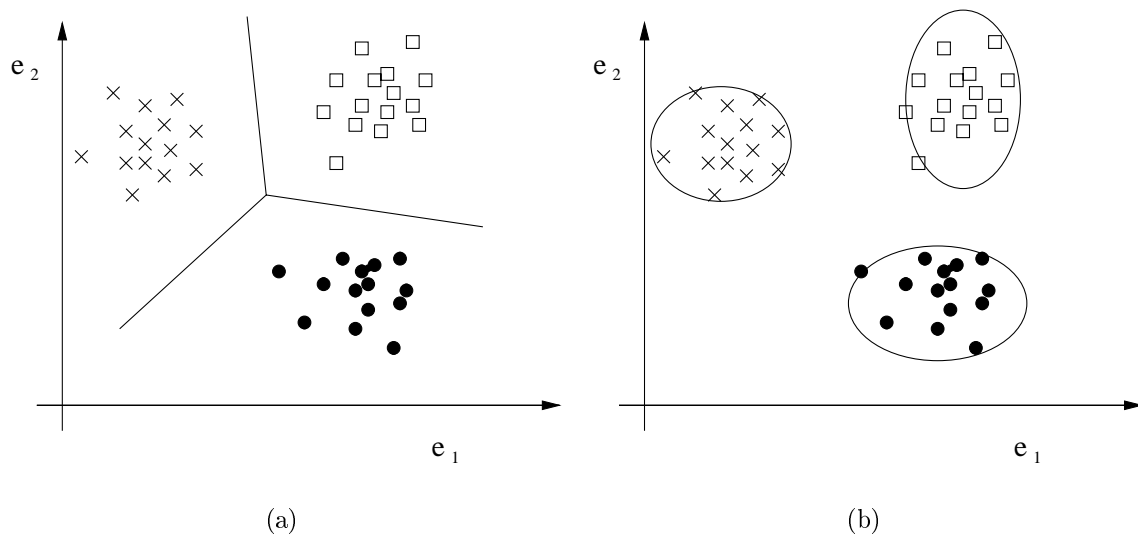


Abbildung 3.5: Einteilung von Datenpunkten in Klassen mit einem Multilayer-Perceptron [RHW86](a) und einem RBF-Netz [BL88](b)

Kapitel 4

Vollbild-Lokalisierung

In diesem Kapitel soll beschrieben werden, wie die in den vorherigen Kapiteln vorgestellten Verfahren wie PCA und Bildklassifikation zur Lokalisierung verwendet werden können. Dieses Verfahren wurde erstmals präsentiert von Jogan & Leonardis [JL00]. Da wir hier mit den kompletten Bildern arbeiten, wird diese Lokalisierungsmethode Vollbild-Lokalisierung genannt. Dies steht im Gegensatz zu der Sektoren-Lokalisierung, die in Kapitel 5 beschrieben wird und mit Teilen der Bilder arbeitet.

4.1 Vorbereitung

Bildaufnahmen

Um die Umgebung zu lernen, in der der Roboter agieren soll, müssen zunächst Trainingsbilder in dieser Umgebung aufgenommen werden. Die Positionen dieser Bilder sollten nicht zu weit auseinanderliegen, so daß zwischen Bildern der Trainingspositionen interpoliert werden kann, und sie sollten den gesamten Bereich abdecken, in dem der Roboter sich später aufhalten wird. In dieser Arbeit wurden die Trainingsbilder in Abständen von 50 cm in einer Büroumgebung des Fraunhofer Instituts für Autonome Intelligente Systeme (AIS) aufgenommen. Abbildung 4.1 zeigt diese Umgebung. Die Punkte in Flur und Raum markieren die Positionen, an denen die Trainingsbilder aufgenommen wurden.

Aufteilen der Umgebung

In unseren Experimenten haben wir die Umgebung in drei Abschnitte aufgeteilt, welche in Abbildung 4.2 zu sehen sind. Der erste Abschnitt (Environment 1) zeigt den Teil des Flurs, in dem sich das Fenster befindet. Aufgrund des Lichteinfalls an dieser Stelle unterscheiden sich die Bilder dieses Abschnitts von denen, die nicht in der Nähe des Fensters aufgenommen wurden. Der zweite Umgebungsabschnitt (Environment 2) zeigt den Teil des Flurs, der sich innerhalb des Gangs befindet. Man beachte, daß die Abschnitte a) und b) sich überlappen. Dies gewährleistet einen sicheren Übergang von einer Umgebung in die nächste. Der dritte Abschnitt (Environment 3) zeigt den Raum.

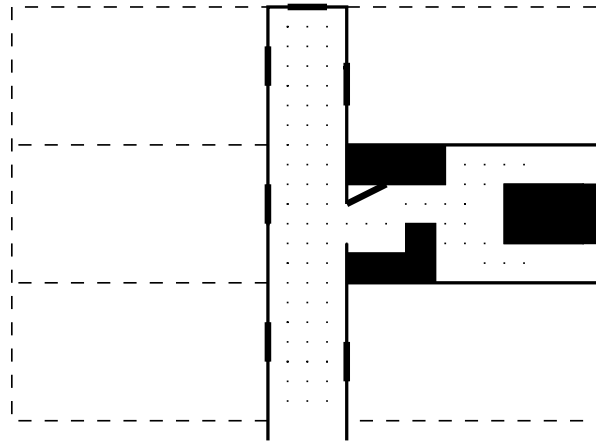


Abbildung 4.1: Die Büroumgebung

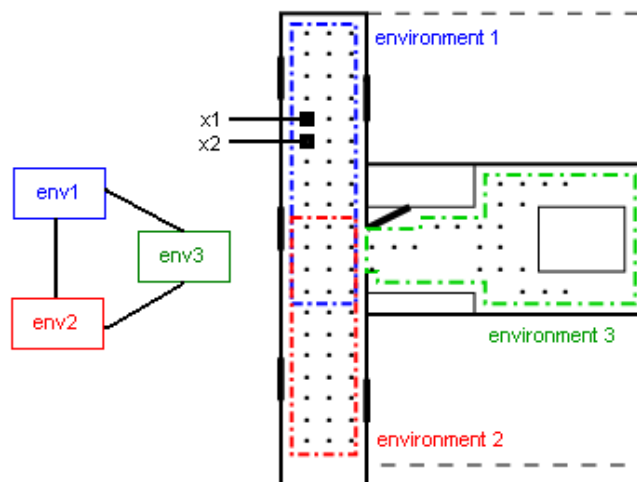


Abbildung 4.2: Die drei Abschnitte der Büroumgebung (rechts), und zugehöriges topologisches Modell dieser Umgebung (links)

Diese Abschnitte entsprechen den drei Knoten eines topologischen Modells, die durch Kanten miteinander verbunden sind (Abbildung 4.2 links). Die Lokalisierung findet nun jeweils in nur einem der Abschnitte statt. Es wird vorausgesetzt, daß der Roboter bereits weiß, in welchem Abschnitt er sich befindet. Das kann durch globales "Position Tracking" ermöglicht werden.

Durch diese Aufteilung in mehrere Abschnitte kann der Ansatz leicht auf größere Büroumgebungen erweitert werden. Es müssen lediglich weitere Abschnitte gelernt werden. Wenn der Roboter in der Lage ist, sich in diesem Gangabschnitt am Fenster zu lokalisieren, wird dies voraussichtlich auch in anderen Gangabschnitten am Fenster möglich sein. Dasselbe gilt für die anderen Abschnitte.

Vorverarbeitung

Die Kamerabilder werden nun in rechteckige Panoramabilder transformiert und in Grauwertbilder umgewandelt (siehe Kapitel 6: Abbildung 6.3). Es entstehen Bilder der Größe $360 \times 200 = 72\,000$ Pixel. Bei unseren Experimenten haben wir ausschließlich die transformierten Panoramabilder verwendet. Im Rahmen dieser Arbeit wurden zudem Versuche unternommen, die Bilder mit verschiedenen Bildbearbeitungsmethoden vorzuverarbeiten. Zum Beispiel wurden Gaborfilter und Canny-Kantendetektoren angewendet (Jain *et al.* [JKS95]). Da diese Experimente jedoch keine Verbesserungen ergaben, wird darauf im folgenden nicht näher eingegangen.

4.2 PCA - Lernen der Umgebung

Nun soll aus den Trainingsbildern die Umgebung gelernt werden. Dazu wird für jeden Umgebungsabschnitt ein Eigenraum mittels PCA erzeugt. Wie in Kapitel 2 beschrieben, werden die Trainingsbilder zunächst normalisiert, dann wird das Durchschnittsbild subtrahiert und schließlich werden mit SVD (S. 17) die Eigenvektoren der Kovarianzmatrix der Trainingsbildmenge berechnet, die den Eigenraum aufspannen. Von diesen Eigenvektoren werden die ersten k ausgewählt; wie in Kapitel 2 beschrieben, handelt es sich hierbei um die Eigenvektoren mit den größten Eigenwerten. In unseren Experimenten wurde $k = 10$ gewählt (siehe Teil II: Experimente).

Schließlich werden die Trainingsbilder in den zugehörigen Eigenraum projiziert, indem sie mit den Eigenvektoren multipliziert werden. Die gelernte Umgebung wird demnach durch die Projektionen der Trainingsbilder repräsentiert. Hierbei handelt es bei unseren Experimenten um Vektoren der Länge 10, im Gegensatz zu den Bildvektoren der Länge 72 000.

4.3 Lokalisierung

Um sich zu lokalisieren, nimmt der Roboter zunächst ein Testbild an seiner momentanen Position auf. Dieses wird analog zu den Trainingsbildern in ein rechteckiges Panoramabild

transformiert. Schließlich wird es in den Eigenraum projiziert und einem Trainingsbild zugeordnet.

Projektion des Testbildes

Analog zur Projektion der Trainingsbilder wird das Testbild in den Eigenraum projiziert. Wie auch die Trainingsbilder entspricht es jetzt einem Punkt im k -dimensionalen Eigenraum.

Klassifikation

Schließlich wird die Position des Testbildes geschätzt. Dazu wird das Testbild mit dem in Kapitel 3.2 beschriebenen Nächste-Nachbar-Klassifikator einer Klasse zugeordnet. Jede Klasse entspricht einer Trainingsbildposition. Die zugeordnete Klasse enthält das Trainingsbild, das dem Testbild am ähnlichsten ist. Die Position, die zu der zugeordneten Klasse gehört, ist nun die geschätzte Position des Testbildes.

4.4 Problemfaktoren

Die bisher beschriebene Lokalisierungsmethode beruht darauf, daß sich Bilder benachbarter Positionen ähnlich sehen, bzw. daß ihr Abstand im Eigenraum gering ist. Um dies zu gewährleisten sind einige Voraussetzungen notwendig, die in natürlichen Umgebungen oft nicht gegeben sind. In diesem Abschnitt sollen einige Schwierigkeiten beschrieben werden, die bei der Vollbild-Lokalisierung auftreten können, sowie mögliche Lösungswege.

Interpolation

Bei der Lokalisierung durch Appearance-based Matching werden Testbilder mit gelernten Trainingsdaten verglichen. Da diese Trainingsdaten nur an diskreten Positionen aufgenommen werden, Testbilder aber von beliebigen Positionen stammen können, kann die Zuordnung Probleme bereiten. In Teil II, Kapitel 7.1.2 wird allerdings gezeigt, daß ein Testbild, das zwischen zwei Trainingspositionen aufgenommen wurde, dennoch diesen benachbarten Bildern ähnlicher sieht als den übrigen Trainingsbildern. So kann die Interpolation zwischen Trainingspositionen erfolgreich durchgeführt werden.

Rotationen

Eine Voraussetzung für die Ähnlichkeit von Bildern benachbarter Positionen ist, daß der Roboter jeweils nach derselben Richtung ausgerichtet ist. In der Realität sind Testbilder aber gewöhnlich um beliebige Winkel rotiert. Dadurch unterscheiden sie sich zum Teil erheblich von den Trainingsbildern. In Teil II, Kapitel 7.1.3 wird gezeigt, daß die Vollbild-Lokalisierung beliebig rotierte Bilder nicht zufriedenstellend klassifizieren kann. Um dennoch mit Rotationen umgehen zu können, gibt es verschiedene Möglichkeiten.

Eine Möglichkeit ist, auch rotierte Bilder zu lernen. Dazu wird zu jedem Trainingsbild eine Menge von Bildern generiert, die aus dem Ursprungsbild durch Rotation entstanden sind (vgl. S. 24). In Kapitel 7 wird gezeigt, daß die Vollbild-Lokalisierung robust gegenüber Rotationen bis etwa 10° ist. Daher könnte beispielsweise ein Bild für jede Rotation um 15° aufgenommen werden, was zu 24 Bildern pro Trainingsposition führen würde. Eine so große Menge von Trainingsbildern führt allerdings sowohl zu Speicherplatzproblemen als auch zu großem zeitlichen Berechnungsaufwand.

Eine Variante, die diese Komplexitätsprobleme umgeht, stellen Matsumo *et al.* in [MIII99] vor. Dabei werden nicht die Trainingsbilder rotiert, sondern die Testbilder. Jede Testbildrotation wird dann klassifiziert und die beste Übereinstimmung liefert das Lokalisierungsergebnis. Hierbei wird allerdings die zusätzlich notwendige Berechnungszeit in die Aktionsphase verlegt, während sie sich bei dem vorherigen Ansatz in der Vorbereitungsphase befand.

Eine ganz andere Möglichkeit zur Behandlung von Rotationen bietet die in Kapitel 5 vorgestellte Sektorenmethode. Da hier einzelne Sektorbilder klassifiziert werden, spielt die Rotation des Roboters keine große Rolle (siehe auch Kapitel 8.2). Eine weitere Möglichkeit stellen Jogan und Leonardis in [JL99] vor.

Beleuchtungsvarianzen: Tag-Nacht-Eigenräume

Wenn Lokalisierung mit visuellen Sensoren betrieben wird, kann unterschiedliche Beleuchtung einen enormen Störfaktor der Sensordaten darstellen. Schon der Unterschied zwischen direkter Sonneneinstrahlung gegenüber bewölktem Himmel kann ein Bild stark verändern. Ein meist noch deutlicherer Unterschied besteht zwischen Bildern, die bei Tag aufgenommen wurden, gegenüber solchen, die bei Dunkelheit mit künstlicher Beleuchtung aufgenommen wurden. Dieser Abschnitt beschreibt eine Methode, die die Lokalisierung anhand von Bildern mit unterschiedlicher Beleuchtung robuster machen soll.

In Kapitel 3.2 wurde bereits erwähnt, daß es verschiedene Möglichkeiten gibt, Prototypen als Klassenrepräsentanten zu bestimmen. Es wurde zwischen zwei grundsätzlichen Möglichkeiten unterschieden:

1. Wähle einen Prototyp pro Klasse
2. Wähle mehrere Prototypen pro Klasse

Bisher wurde zur Klassifikation von Testbildern immer Fall 1. angewendet, wobei als Prototypen die Trainingsbilder gewählt wurden. Jede Position wurde also durch ein Trainingsbild repräsentiert. Um nun die Zuordnung von Nachtbildern zu verbessern, kann stattdessen Fall 2. angewandt werden. Als Trainingsdaten werden nicht nur Tagbilder verwendet, sondern zusätzlich Nachtbilder derselben Positionen. Eine Position wird nun durch mehrere Trainingsbilder repräsentiert und zwar durch ein Tag- und durch ein Nachtbild. Wenn nun ein Nachtbild klassifiziert werden soll, wird es idealerweise dem zugehörigen

Trainingsnachtbild zugeordnet, ein Tagbild wird dem Trainingstagbild zugeordnet. Ein solcher Eigenraum, der aus Tag- und aus Nachtbildern generiert wurde, wird im folgenden *Tag-Nacht-Eigenraum* genannt.

Die Erstellung eines solchen Eigenraums ist aufwendiger als die Verwendung eines reinen Tag-Eigenraums, da doppelt so viele Trainingsbilder aufgenommen werden müssen. Auch wird mehr Zeit benötigt sowohl um den Eigenraum zu erstellen als auch um das Testbild zu klassifizieren. Da beide Berechnungen aber nicht von der Anzahl der Trainingsbilder, sondern von der Anzahl der Pixel pro Bild dominiert werden, fällt dies nicht sonderlich ins Gewicht (vergleiche Kapitel 2.3: Komplexität von PCA). Wenn ein Tag-Nacht-Eigenraum einmal erstellt ist, bietet er deutliche Vorteile gegenüber einem reinen Tag-Eigenraum, wie die Experimente in Kapitel 7.2.3 zeigen.

Okklusionen

Ein weiterer Faktor, der die Lokalisierung erschwert, sind Veränderungen der Umgebung, hervorgerufen durch Gegenstände oder umherlaufende Personen. In Kamerabildern finden sich solche Hindernisse als Okklusionen von Teilbereichen der Bilder wieder. Solche Veränderungen sind in der Realität an der Tagesordnung. Nicht nur Personen, auch geöffnete Türen, verrückte Stühle oder neue Poster an den Wänden können die Bilder stark verändern, so daß die Klassifizierung schwierig wird. Die Experimente in Kapitel 7.3 zeigen, daß die Vollbild-Lokalisierung unter solchen Bedingungen deutlich schlechter ausfällt.

Bessere Ergebnisse liefert die im folgenden Kapitel vorgestellte Sektoren-Lokalisierung. Dabei lokalisiert sich der Roboter anhand der Bildteile, die nicht verdeckt sind und somit eine gute Übereinstimmung mit den Trainingsdaten liefern. Einen anderen Ansatz zur robusten Lokalisierung bezüglich Okklusionen stellen Jogan und Leonardis in [JL00] vor.

Kapitel 5

Sektoren-Lokalisierung

Viele Lokalisierungsmethoden werden unzuverlässig, wenn die Sensordaten gestört sind. Diese Störungen können im Fall visueller Sensorik etwa durch Beleuchtungs- oder Positionierungsschwankungen hervorgerufen werden oder dadurch, daß Teile der Bilder durch Gegenstände oder umherlaufende Personen verdeckt sind. In einer natürlichen Umgebung sind aber solche Störungen an der Tagesordnung.

Dieses Kapitel stellt eine Methode vor, die die Robustheit der Lokalisierung unter solchen Bedingungen vergrößert. Sie basiert auf der Aufspaltung der Sensordaten in viele kleine Datenmengen. Dazu werden die Panoramabilder in einzelne Sektorbilder unterteilt. Diese Sektorbilder werden nun einzeln zur Lokalisierung genutzt. Das hat den Vorteil, daß nicht-verdeckte Sektoren sehr gute Ergebnisse liefern. Allerdings können hierbei leicht Mehrdeutigkeiten auftreten, da ein Sektor deutlich weniger Information enthält als ein globales Panoramabild. So kommt es oft vor, daß ein Sektor einen Teil der Umgebung beinhaltet, dessen Erscheinungsform häufig vorkommt. Diese Ambiguitäten können verringert werden, indem der visuelle Kontext des Sektorbildes betrachtet wird (Paletta, Frintrop & Hertzberg [PFH01b]). Dieser wird durch die Reihenfolge weiterer Sektoren desselben Panoramabildes repräsentiert. Voraussetzung für das Auflösen von Mehrdeutigkeiten ist, daß einige dieser Sektoren ebenfalls nicht okkludiert bzw. nicht gestört sind. Jogan und Leonardis stellen in [JL00] ebenfalls einen Ansatz zur robusten Lokalisierung bezüglich Okklusionen vor.

Dieses Kapitel liefert in Abschnitt 5.1 zunächst eine Motivation dieses Ansatzes. In Abschnitt 5.2 wird dann beschrieben, wie die Aufteilung in Sektoren sowie die Projektion der Sektorbilder in den Eigenraum vonstatten gehen und wie die Klassifikation von Sektoren mittels Bayes'schem Update durchgeführt wird. Abschnitt 5.3 behandelt die Klassifikation unter Berücksichtigung des lokalen visuellen Kontexts und Abschnitt 5.4 geht schließlich auf die weitgehende Rotationsunabhängigkeit des Verfahrens ein.

5.1 Motivation

Ein großer Vorteil omnidirektionaler Kamerabilder ist die umfangreiche Informationsmenge, die ein solches Bild enthält. Aufgrund des großen visuellen Feldes, das durch die Bilder abgedeckt wird, ist es sehr unwahrscheinlich, daß das gesamte Blickfeld durch Hindernisse verdeckt wird. Abbildung 5.1 zeigt eine Situation, in der sich vier Personen in unmittelbarer Nähe des Roboters (Distanz ca. 1m) befinden. Dies dürfte für eine Büroumgebung sicherlich eine Extremsituation darstellen; trotzdem sind Bildteile zu erkennen, die nicht verdeckt sind.



Abbildung 5.1: Panoramabild mit starker Verdeckung

Es liegt nun nahe, die Lokalisierung mit Hilfe dieser nicht verdeckten Bildteile zu realisieren. Das Panoramabild wird dazu in einzelne Elemente aufgeteilt, genannt Sektoren bzw. Sektorbilder. Abbildung 5.2 zeigt das Panoramabild aus Abbildung 5.1, eingeteilt in 36 Sektoren. Die Sektoren sind je 20 Pixel breit und überlappen sich um die Hälfte, d.h. um 10 Pixel. Wie man sieht, sind viele dieser Sektoren nicht verdeckt und sollten somit eine gute Übereinstimmung zu Sektoren der Trainingsbilder liefern.



Abbildung 5.2: In Sektoren aufgeteiltes Panoramabild. Die Sektorbilder überlappen sich rechts und links um je 50%.

5.2 Multisektorenklassifikation

Zunächst werde jedes Bild in Σ Sektoren eingeteilt. Dann gibt es zu jeder Position ϕ_i des Trainingsbildes x_i die Sektorenssequenz $(x_{i,1}, \dots, x_{i,\Sigma})$. Nun wird ein Eigenraum aus den Sektorbildern aller Trainingsbilder berechnet, d.h. aus der Bildmenge $\{x_{1,1}, \dots, x_{N,\Sigma}\}$. Zu jeder Roboterposition ϕ_i gehört nun die Punktsequenz $(g_{i,1}, \dots, g_{i,\Sigma})$ im Eigenraum, die den zugehörigen Sektorbildern entspricht.

Da die Aufteilung der Bilder in Sektoren diskret ist, tritt ein Problem auf, wenn der Roboter um einen beliebigen Winkel rotiert ist, denn dadurch hat ein Testbild eine andere Sektoraufteilung als die Trainingsbilder (vgl. Kapitel 4.4: Rotationsproblematik bei der Vollbild-Lokalisierung). Als mögliche Lösung können die Trainingsbildsektoren als Stützstellen genutzt werden, die zu einer geschlossenen Kurve interpoliert werden, im folgenden Sektor-Trajektorie genannt (vgl. Paletta, Frintrop & Hertzberg [PFH01b]). Abbildung 5.3 zeigt zwei solcher Sektor-Trajektorien, wobei die Kästchen die Sektorprojektionen darstellen. Man sieht hier, daß die Trajektorien sehr dicht beieinander liegen, was auf eine große Ähnlichkeit der zugehörigen Bilder x_1 und x_2 schließen läßt. Tatsächlich stammen die Bilder von benachbarten Roboterpositionen.

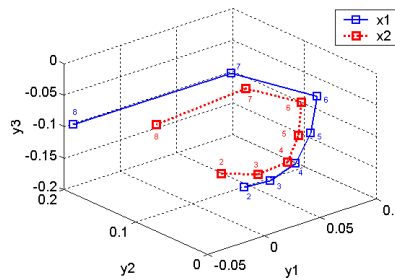


Abbildung 5.3: Die Projektionen der Sektoren der Bilder x_1 und x_2 von benachbarten Positionen ϕ_1 und ϕ_2 .

Bayes'sche Interpretation

Um ein Testbild y zu klassifizieren, wird dieses in die Sektoren (y_1, \dots, y_Σ) aufgeteilt und in den Eigenraum projiziert. Dort stellt es die Punktsequenz (h_1, \dots, h_Σ) dar. Um die Position des Roboters zu bestimmen, werden diese Testsektoren einzeln klassifiziert. Dazu wird zunächst für jede Testsektorprojektion h_k die Wahrscheinlichkeit berechnet, an Position ϕ_i mit Rotation ω_j zu sein. Dies geschieht mit der Bayes'schen Formel:

$$P(\phi_i, \omega_j | h_k) = p(h_k | \phi_i, \omega_j) P(\phi_i, \omega_j) / p(h_k), \quad (5.1)$$

mit $1 \leq i, k \leq \Sigma$ und $0 \leq j < \Sigma$. Die Wahrscheinlichkeit $p(h_k | \phi_i, \omega_j)$ wird hierbei aus einer Menge von Trainingsdaten mit festem ϕ_i geschätzt, die sich leicht durch Beleuchtungs- und Positionsvarianzen unterscheiden. Die Roboterrotation ω_j wird dabei insofern berücksichtigt, als daß der Testsektor y_k mit dem Trainingssektor $x_{i, (j+k) \bmod \Sigma}$ verglichen wird, bzw. Testprojektion h_k mit Trainingsprojektion $g_{i, (j+k) \bmod \Sigma}$. Die Wahrscheinlichkeiten $P(\phi_i, \omega_j)$ und $p(h_k)$ werden in dieser Arbeit als gleichverteilt angenommen. Falls der Roboter zusätzliche Informationen hat, die beispielsweise durch andere Sensoren gewonnen worden sind, können diese dazu dienen, diese Wahrscheinlichkeiten genauer festzulegen.

Nach dem Satz der totalen Wahrscheinlichkeit (Feller [Fel50]) können nun für jede Position die Wahrscheinlichkeiten aller Trainingssektoren aufsummiert werden:

$$P(\phi_i|h_k) = \sum_j P(\phi_i, \omega_j|h_k). \quad (5.2)$$

Ergebnis der Klassifikation von Testpunkt h_k ist nun die Position mit der höchsten Wahrscheinlichkeit:

$$\Phi_k = \arg \max_{\phi_i} P(\phi_i|h_k). \quad (5.3)$$

Es ergibt sich ein Vektor (Φ_1, \dots, Φ_k) , der für jeden Testsektor eine geschätzte Position enthält. Um schließlich die Position Φ des globalen Testbildes y zu bestimmen, kann ein Mehrheitsentscheid über alle Sektoren (h_1, \dots, h_Σ) getroffen werden:

$$\Phi = \text{maj } \Phi_k \quad \text{mit} \quad k = 1.. \Sigma, \quad (5.4)$$

wobei mit *maj* die Majoritätsfunktion bezeichnet wird. Diese wählt den Positionswert Φ_k aus, der am häufigsten auftritt. Alternativ kann auch der vertrauenswürdigste Sektor ausgewählt werden, d.h. derjenige der die besten Werte liefert. Die Formeln 5.3 und 5.4 werden dazu ersetzt durch:

$$Q(\phi_i) = \max_k P(\phi_i|h_k), \quad (5.5)$$

$$\Phi = \arg \max_{\phi_i} Q(\phi_i). \quad (5.6)$$

Ein Beispiel

Veranschaulichen wir dieses Verfahren nun an einem Beispiel. Sei $y = (y_1, \dots, y_\Sigma)$ ein Testbild, das an Roboterposition ϕ_3 aufgenommen wurde, wobei der Roboter um 4 Sektorbreiten rotiert war, und (h_1, \dots, h_Σ) sei die Punktsequenz der Sektoren im Eigenraum. Wir setzen voraus, daß das Testbild unokkludiert ist und perfekt zugeordnet wird. Wenn nun Formel 5.1 angewandt wird, sollten die Wahrscheinlichkeiten $P(\phi_3, \omega_4|h_k)$ höhere Werte liefern als die Wahrscheinlichkeiten anderer Positionen und Rotationen. Bei diesen Werten wird der Punkt h_k mit der Trainingssektorprojektion $g_{3,(4+k) \text{ MOD } \Sigma}$ verglichen, bzw. auf die Sektordarstellung bezogen, der Testsektor y_k mit dem Trainingssektor $x_{3,(4+k) \text{ MOD } \Sigma}$; d.h. y_1 wird mit $x_{3,5}$ verglichen, y_2 mit $x_{3,6}$ usw. Abbildung 5.4 veranschaulicht die Übereinstimmung von Test- und Trainingssektoren bei diesen Wahrscheinlichkeiten.

Tabelle 5.1 zeigt die Wahrscheinlichkeiten, die für Position ϕ_3 berechnet werden. Jede Spalte enthält dabei alle Wahrscheinlichkeiten für einen Testpunkt, die Zeilen enthalten alle

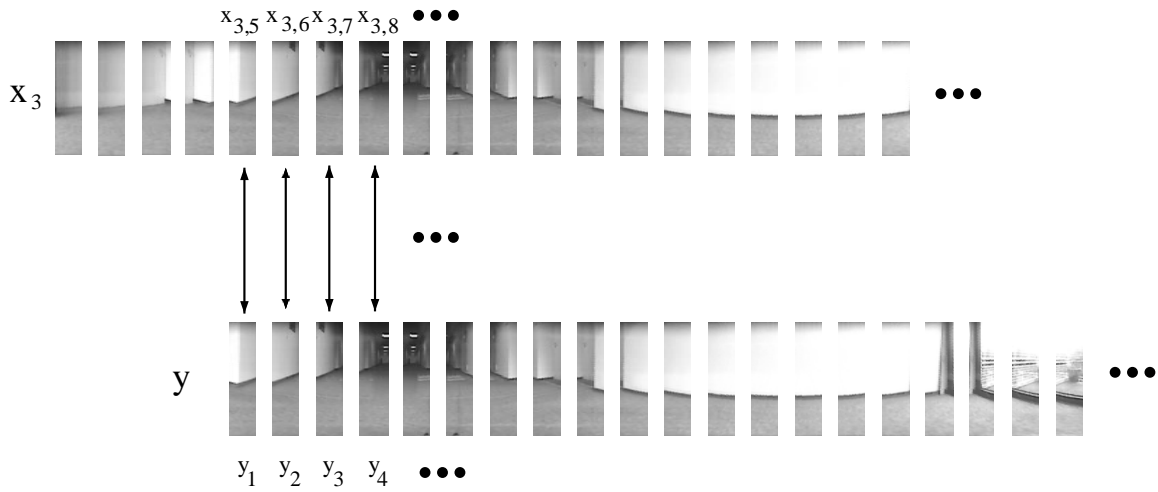


Abbildung 5.4: Trainingsbildsektoren (oben) und die geschifteten Testbildsektoren (unten)

Wahrscheinlichkeiten für eine Roboterrotation. Die Wahrscheinlichkeiten, die hohe Werte liefern, sind durch Rahmen hervorgehoben. In der untersten Zeile der Tabelle werden die Werte $P(\phi_3|h_k)$ dargestellt, die man durch Aufsummieren der einzelnen Spalten erhält (vergleiche Gleichung 5.2). Eine solche Tabelle wird nun für jede Position ϕ_i berechnet. Die Tabellen der übrigen Positionen sollten im Idealfall durchschnittlich niedrigere Wahrscheinlichkeitswerte enthalten als die Tabelle für Position ϕ_3 .

Tabelle 5.2 zeigt die Wahrscheinlichkeiten aller Positionen, die man durch Gleichung 5.2 erhält. Wieder enthält eine Spalte die Werte für einen Testpunkt, in den Zeilen sind

	Testpunkt $h_k \longrightarrow$			
Rotation ω_j ↓	$P(\phi_3, \omega_1 h_1)$	$P(\phi_3, \omega_1 h_2)$	$P(\phi_3, \omega_1 h_3)$...
	$P(\phi_3, \omega_2 h_1)$	$P(\phi_3, \omega_2 h_2)$	$P(\phi_3, \omega_2 h_3)$...
	$P(\phi_3, \omega_3 h_1)$	$P(\phi_3, \omega_3 h_2)$	$P(\phi_3, \omega_3 h_3)$...
	$P(\phi_3, \omega_4 h_1)$	$P(\phi_3, \omega_4 h_2)$	$P(\phi_3, \omega_4 h_3)$...
	$P(\phi_3, \omega_5 h_1)$	$P(\phi_3, \omega_5 h_2)$	$P(\phi_3, \omega_5 h_3)$...
	$P(\phi_3, \omega_6 h_1)$	$P(\phi_3, \omega_6 h_2)$	$P(\phi_3, \omega_6 h_3)$...
	$P(\phi_3, \omega_7 h_1)$	$P(\phi_3, \omega_7 h_2)$	$P(\phi_3, \omega_7 h_3)$...
	\vdots	\vdots	\vdots	\ddots
\sum_j	$P(\phi_3 h_1)$	$P(\phi_3 h_2)$	$P(\phi_3 h_3)$...

Tabelle 5.1: Wahrscheinlichkeitswerte für Position ϕ_3

	Testpunkt $h_k \longrightarrow$			
Position ϕ_i	$P(\phi_1 h_1)$	$P(\phi_1 h_2)$	$P(\phi_1 h_3)$	\dots
\downarrow	$P(\phi_2 h_1)$	$P(\phi_2 h_2)$	$P(\phi_2 h_3)$	\dots
	$P(\phi_3 h_1)$	$P(\phi_3 h_2)$	$P(\phi_3 h_3)$	\dots
	$P(\phi_4 h_1)$	$P(\phi_4 h_2)$	$P(\phi_4 h_3)$	\dots
	\vdots	\vdots	\vdots	\ddots
$\arg \max_{\phi_i}$	ϕ_3	ϕ_3	ϕ_3	\dots

Tabelle 5.2: Die Wahrscheinlichkeiten $P(\phi_i|h_k)$

die unterschiedlichen Positionen dargestellt. Die hohen Wahrscheinlichkeiten sind erneut durch Rahmen hervorgehoben. Diesmal stammen diese von den Werten zu Position ϕ_3 , da hier die besten Übereinstimmungen mit den Trainingsdaten zu finden sind. Nun wird von jeder Spalte das Maximum bestimmt, um die Wahrscheinlichkeitsschätzung für einen Testpunkt zu gewinnen. Die zugehörige Position ist der Wert Φ_k , der in der untersten Zeile der Tabelle 5.2 dargestellt wird (vergleiche Gleichung 5.3). Ergebnis der Lokalisierung ist schließlich die Position, die in dieser Zeile am häufigsten vorkommt (Gleichung 5.4). In unserem Fall wäre dies also die Position ϕ_3 .

Wenn stattdessen von dem vertrauenswürdigsten Sektor auf die Position geschlossen werden soll, wird aus Tabelle 5.2 der höchste Wert ausgewählt (Gleichungen 5.5 und 5.6). Die zugehörige Position ist Ergebnis der Lokalisierung. In unserem Fall wäre dies erneut die Position ϕ_3 .

Mehrdeutigkeiten

Wie bereits erwähnt (S. 37), kann es bei obiger Klassifikationsmethode leicht zu Verwechslungen kommen. Wenn Sektoren verdeckt sind, ergeben die in Tabelle 5.1 eingerahmten Wahrscheinlichkeiten nicht mehr die höchsten Werte. Außerdem sehen sich einzelne Sektorbilder aufgrund ihrer geringen Informationsmenge häufig sehr ähnlich. Das führt dazu, daß in Tabelle 5.2 auch Werte anderer Positionen hohe Wahrscheinlichkeiten liefern, so daß Φ_k falsche Positionsschätzungen zugewiesen bekommt. Dadurch können die Ergebnisse eventuell sogar schlechter ausfallen als bei globaler Lokalisierung ohne Sektoraufteilung.

Erst wenn mehrere Sektorbilder einer Position betrachtet werden, kann wieder die ursprüngliche Erkennungsrate erzielt werden. Mit dem im folgenden Abschnitt beschriebenen Verfahren können sogar bessere Ergebnisse erzielt werden als im globalen Fall. Dies trifft vor allem in Situationen zu, in denen einige Sektoren gute und andere schlechte Übereinstimmung mit den Trainingsbildern liefern, wie es z.B. der Fall ist, wenn Teilbereiche der Bilder durch Okklusionen verdeckt sind.

5.3 Visueller Kontext

Wie oben erwähnt, erzielen einzelne Sektorbilder bei Projektion in den Eigenraum häufig keine guten Erkennungsraten. Wenn allerdings der Kontext des Sektorbildes einbezogen wird, können Mehrdeutigkeiten verringert werden; so können auch bei gestörten Bildern und Bildern mit Okklusionen gute Erkennungsraten erzielt werden. Dabei verstehen wir unter Kontext die Information der übrigen Sektorbilder desselben globalen Bildes, wobei die Reihenfolge berücksichtigt wird.

Motivation

Zunächst eine Motivation dieses Sachverhalts anhand eines Beispiels. Ein zu klassifizierendes Bild y sei aufgeteilt in die Sektoren (y_1, \dots, y_Σ) , die zugehörigen Projektionen im Eigenraum seien (h_1, \dots, h_Σ) ; weiterhin seien $x_i = (x_{i,1}, \dots, x_{i,\Sigma})$ und $x_j = (x_{j,1}, \dots, x_{j,\Sigma})$ zwei Trainingsbilder und $g_i = (g_{i,1}, \dots, g_{i,\Sigma})$ und $g_j = (g_{j,1}, \dots, g_{j,\Sigma})$ die zugehörigen Projektionen im Eigenraum.

Einer der Testpunkte h_k liege nun sehr nah an den Trainingssektorprojektionen $g_{i,l}$ und $g_{j,m}$. Da die zugehörigen Trainingsbilder x_i und x_j sich sehr ähnlich sehen, liegen ihre Projektionen im Eigenraum dicht beieinander. Um zu bestimmen, zu welchem dieser Bilder der Testsektor y_k gehört, betrachte einen weiteren Sektor des globalen Bildes y , beispielsweise den Sektor y_{k+5} . Dieser werde ebenfalls in den Eigenraum projiziert und stelle dort den Punkt h_{k+5} dar. Nun werde getestet, ob diese Projektion besser zu Trainingsbild x_i oder zu Trainingsbild x_j paßt. Dazu wandere auf den Sektor-Trajektorien dieser Bilder ebenfalls um 5 Schritte weiter, und erhalte so die Punkte $g_{i,l+5}$ und $g_{j,m+5}$. Die Klassifikationsentscheidung fällt nun auf x_i , wenn sich der zweite Testpunkt h_{k+5} näher an $g_{i,l+5}$ als an $g_{j,m+5}$ befindet, ansonsten fällt sie auf x_j . Abbildung 5.5 stellt dieses Beispiel für den Fall dar, in dem das Trainingsbild x_i als Ergebnis der Klassifikation ausgewählt wird.

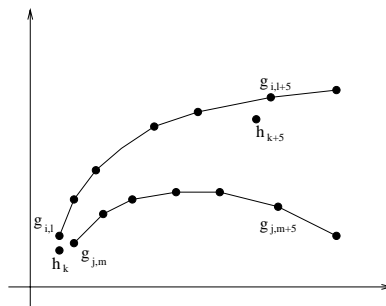


Abbildung 5.5: Klassifikation eines Testbildes unter Berücksichtigung des visuellen Kontexts.

Formalisierung

Diese Berücksichtigung des Kontexts soll nun formalisiert werden. Wenn zwei Sektoren betrachtet werden und in die Berechnungen einfließen, sieht die Bayes'sche Formel folgendermaßen aus:

$$P(\phi_i, \omega_j | h_k, h_{k+a}) = \alpha p(h_{k+a} | \phi_i, \omega_j, h_k) P(\phi_i, \omega_j | h_k), \quad (5.7)$$

mit $k+a \leq \Sigma$ und $\alpha = 1/p(h_{k+a} | h_k)$ als Normalisierungsfaktor. Bei gegebener Kombination (ϕ_i, ω_j) ist die Wahrscheinlichkeit h_{k+a} zu messen unabhängig von der Tatsache h_k zu messen, mit anderen Worten:

$$p(h_{k+a} | \phi_i, \omega_j, h_k) = p(h_{k+a} | \phi_i, \omega_j). \quad (5.8)$$

Mit dieser Umformung und erneuter Anwendung der Bayes'schen Formel auf den Term $P(\phi_i, \omega_j | h_k)$ erhält man nun

$$\begin{aligned} P(\phi_i, \omega_j | h_k, h_{k+a}) &= \alpha p(h_{k+a} | \phi_i, \omega_j) p(h_k | \phi_i, \omega_j) P(\phi_i, \omega_j) / p(h_k) \\ &= \alpha' p(h_{k+a} | \phi_i, \omega_j) p(h_k | \phi_i, \omega_j) P(\phi_i, \omega_j), \end{aligned} \quad (5.9)$$

mit $k+a \leq \Sigma$ und $\alpha' = 1/p(h_{k+a}, h_k)$. Dabei wird der Testpunkt h_k mit dem Trainingspunkt $g_{i,(j+k) \bmod \Sigma}$ verglichen und h_{k+a} mit $g_{i,(j+k+a) \bmod \Sigma}$. Die Wahrscheinlichkeitsverteilung für die Positionsschätzung lautet jetzt:

$$P(\phi_i | h_k, h_{k+a}) = \sum_j P(\phi_i, \omega_j | h_k, h_{k+a}). \quad (5.10)$$

Als Ergebnis der Lokalisierung wird schließlich die Position ϕ_r ausgewählt, für die diese Wahrscheinlichkeit maximal ist:

$$Q(\phi_i) = \max_{k,a} P(\phi_i | h_k, h_{k+a}), \quad (5.11)$$

$$\phi_r = \arg \max_{\phi_i} Q(\phi_i). \quad (5.12)$$

Wenn nun nicht nur zwei, sondern d Sektorbilder in Betracht gezogen werden sollen, lautet die Bayes'sche Formel:

$$P(\phi_i, \omega_j | h_k, h_{k+s_1}, \dots, h_{k+s_{d-1}}) = \alpha' p(h_{k+s_{d-1}} | \phi_i, \omega_j) P(\phi_i, \omega_j | h_k, h_{k+s_1}, \dots, h_{k+s_{d-2}}),$$

mit $s_i = \sum_{j=1}^i a_j$ und $\alpha' = 1/p(h_k, h_{k+s_1}, \dots, h_{k+s_{d-1}})$. In dieser Formel können alle Kombinationen von k und d auftreten, was bedeutet, daß jede mögliche Kombination der Testsektoren betrachtet werden muß. Das würde zu einer sehr großen Anzahl von Wahrscheinlichkeitswerten für jede Position ϕ_i führen. In dieser Arbeit wurden daher nur jeweils zwei bzw. drei Sektoren betrachtet (siehe Kapitel 8).

Ein Beispiel

Betrachten wir zur Veranschaulichung wieder das Beispiel von Seite 40. Wiederum sei $y = (y_1, \dots, y_\Sigma)$ ein Testbild, das an Roboterposition ϕ_3 aufgenommen wurde, wobei der Roboter um 4 Sektorbreiten rotiert war und (h_1, \dots, h_Σ) die Punktsequenz der Sektoren im Eigenraum. Das Testbild sei unokkludiert und werde perfekt klassifiziert.

Mit dem oben beschriebenen Verfahren ergeben sich dann hohe Werte für alle $P(\phi_3, \omega_4 | h_k, h_{k+a})$ mit $k + a \leq \Sigma$, d.h. für folgende Wahrscheinlichkeiten: $P(\phi_3, \omega_4 | h_1, h_2)$, $P(\phi_3, \omega_4 | h_1, h_3)$, $P(\phi_3, \omega_4 | h_2, h_3)$, usw. Abbildung 5.6 zeigt, welche Sektoren bei diesen Wahrscheinlichkeiten verglichen werden. Als nächstes werden mit Gleichung 5.10 die Wahrscheinlichkeiten $P(\phi_i | h_k, h_{k+a})$ für alle möglichen Werte von k und a berechnet. Die sich ergebenden Werte werden für ϕ_3 in Tabelle 5.3 dargestellt. Aus dieser Wahrscheinlichkeits-

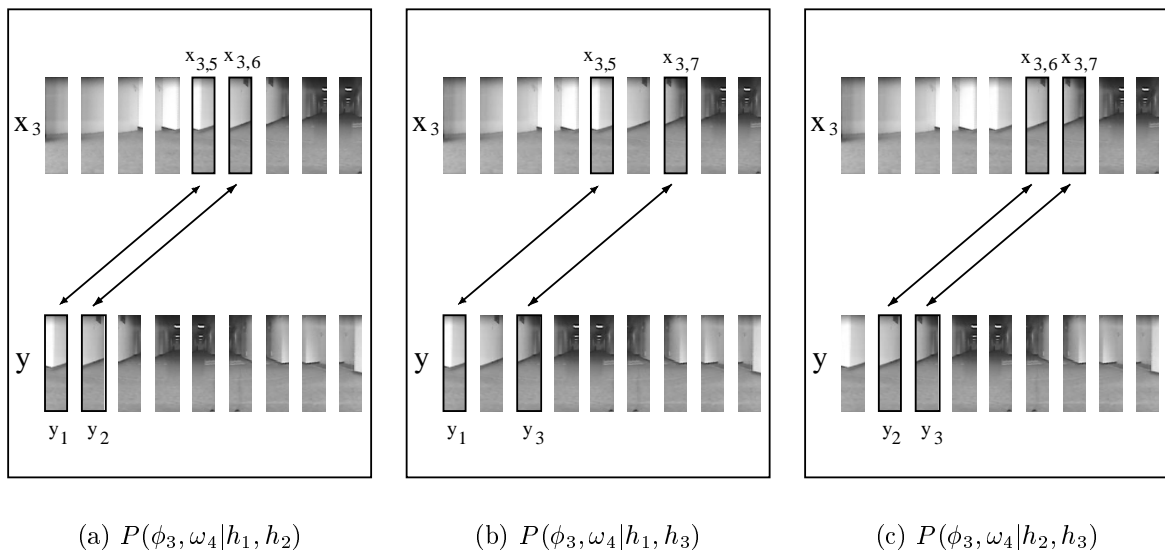


Abbildung 5.6: Darstellung der Sektorübereinstimmung zu drei Wahrscheinlichkeiten, die hohe Werte liefern (mit Position Φ_3 , Rotation ω_4 und Projektion h_k des Testsektors y_k).

tabelle wird nun mit Hilfe von Gleichung 5.11 der höchste Wert ausgewählt um so $Q(\phi_3)$ zu erhalten. Die entgültige Position erhält man schließlich, indem man mit Gleichung 5.12 das ϕ_i bestimmt, das den maximalen Wert $Q(\phi_i)$ liefert. In diesem Beispiel sollte das ϕ_3 sein.

Dieses Verfahren ist besser in der Lage Mehrdeutigkeiten aufzulösen als die Lokalisierung mit einzelnen Sektoren aus Kapitel 5.2. Je mehr Sektoren mitsamt ihres Kontextes betrachtet werden, desto erfolgreicher die Auflösung von Mehrdeutigkeiten. Man beachte jedoch, daß bei einer festen Zahl s von Sektoren nur dann gute Ergebnisse erzielt werden können, wenn mindestens s Sektoren des Testbildes nicht verdeckt sind. Es muß also ein Kompromiß gefunden werden zwischen dem Wunsch, Mehrdeutigkeiten auflösen zu können, und dem, Bilder mit starken Okklusionen klassifizieren zu können. Als Verbesserungsmöglichkeit könnte die Zahl verwendeter Sektoren dynamisch generiert werden, indem

$P(\phi_3 h_1, h_2)$	$P(\phi_3 h_2, h_3)$	$P(\phi_3 h_3, h_4)$	\dots	$P(\phi_3 h_{\Sigma-1}, h_{\Sigma})$
$P(\phi_3 h_1, h_3)$	$P(\phi_3 h_2, h_4)$	\dots	$P(\phi_3 h_{\Sigma-2}, h_{\Sigma})$	
\vdots	\vdots			
$P(\phi_3 h_1, h_{\Sigma-1})$	$P(\phi_3 h_2, h_{\Sigma})$			
$P(\phi_3 h_1, h_{\Sigma})$				

Tabelle 5.3: Die Wahrscheinlichkeiten $P(\phi_i|h_k, h_{k+a})$

all die Sektoren genutzt werden, die einen gewissen Schwellwert an Übereinstimmung mit den Trainingsdaten überschreiten. Die Experimente zu diesem Verfahren werden in Kapitel 8 präsentiert.

5.4 Rotationsunabhängige Lokalisierung

In Kapitel 4.4 wurde dargelegt, daß rotierte Bilder mit der Vollbild-Lokalisierung nicht ohne weiteres zu klassifizieren sind. Die Sektoren-Lokalisierung bietet nun die Möglichkeit, auch rotierte Bilder ohne weitere Vorverarbeitung zu klassifizieren. Dadurch, daß ein oder mehrere Testsektoren mit allen möglichen Trainingssektoren verglichen werden, ist es unwichtig, welche Ausrichtung der Roboter hatte. Die Beispiele in den Kapiteln 5.2 und 5.3 verdeutlichen dies, da das Testbild hier um 4 Sektorbreiten rotiert war.

Da die Wahrscheinlichkeiten $P(\phi_i, \omega_j|h_k)$ bzw. $P(\phi_i, \omega_j|h_k, h_{k+a})$ berechnet werden, wird die Wahrscheinlichkeit für eine Rotation bei der Positionsbestimmung gleich mitbestimmt. Wenn also die Rotation von Interesse ist, sollte direkt das Maximum dieser Wahrscheinlichkeiten gesucht werden, anstatt zunächst in Gleichung 5.2 bzw. 5.10 über die möglichen Rotationen zu summieren. Die Rotation R ergibt sich dann durch

$$R_k = \arg \max_j P(\phi_r, \omega_j|h_k), \quad (5.13)$$

$$R = \text{maj } R_k, \quad (5.14)$$

bzw. für die Lokalisierung mit Kontext:

$$R_k = \arg \max_j P(\phi_r, \omega_j|h_k, h_{k+a}), \quad (5.15)$$

$$R = \text{maj } R_k. \quad (5.16)$$

Man beachte, daß R nicht den exakten Rotationswinkel angibt, sondern nur die Information, um wieviele Sektoren das Testbild verschoben ist. Die Genauigkeit der Schätzung hängt also von der Anzahl verwendeter Sektoren ab. Diese Information kann zusammen

mit Werten anderer Rotationsmessungen in eine Sensorfusion einfließen, um eine genauere Rotationsschätzung zu erhalten.

Die Rotationsunabhängigkeit des Verfahrens gilt folglich auch nur für Rotationen, die ein Vielfaches des Abstandes zwischen benachbarten Sektoren beträgt. Um Rotationen um beliebige Winkel zuzulassen, müßte zwischen den Punkten der Sektoren im Eigenraum interpoliert werden, um Sektor-Trajektorien zu erhalten (vgl. S. 39), was in dieser Arbeit allerdings nicht behandelt wurde. Ohne diese Sektor-Trajektorien werden die Sektoren leicht rotierter Bilder wie beschrieben den ähnlichsten Trainings-Projektionen zugeordnet. Da die Sektoren nur sehr wenig Information enthalten, ist das Verfahren anfälliger gegenüber kleinen Rotationsschwankungen als die Vollbild-Lokalisierung. Die Güte der Klassifikation schwankt also zwischen sehr gut an Vielfachen der Sektorabstände und einem schlechteren Wert zwischen diesen. Wie in Kapitel 9.2 gezeigt wird, zeigen sich die Vorteile der Sektoren-Lokalisierung bezüglich Rotationen besonders ab einem Rotationswinkel von 20° , da die Vollbild-Lokalisierung solche und noch stärker rotierte Bildern nicht erfolgreich klassifizieren kann. In Kapitel 8 werden Experimente beschrieben, in denen rotierte Bilder mit der Sektoren-Lokalisierung klassifiziert werden.

5.5 Komplexität

Die Komplexität der Sektoren-Lokalisierung mit Kontext hängt wesentlich von der Anzahl der Sektoren pro Bild ab. Weiterhin hängt sie von der Anzahl von Sektoren ab, die zur Lokalisierung herangezogen werden. Die folgende Analyse bezieht sich auf die Lokalisierung mit zwei Sektoren. Sei N die Anzahl der Roboterpositionen und Σ die Anzahl der Sektoren pro Bild. Damit gibt es folglich auch Σ mögliche Rotationen. Dann werden bei der Lokalisierung eines Testbildes $(\Sigma - 1)^2/2$ Sektorkombinationen betrachtet (vergleiche Tabelle 5.3). Da für jede dieser Sektorkombinationen alle Positionen und alle Rotationen überprüft werden müssen, erfolgen jeweils $N\Sigma$ Wahrscheinlichkeitsberechnungen.

Dies resultiert in einer Komplexität von $O(N\Sigma^3)$. Damit sind Echtzeitberechnungen möglich, zumal sowohl N als auch Σ keine hohen Werte annehmen, sondern üblicherweise im zweistelligen Bereich bleiben. In unseren Experimenten lag N zwischen 20 und 40 und Σ betrug 36 (vgl. Teil II: Experimente).

Teil II

Experimente

Übersicht zum zweiten Teil

Im zweiten Teil dieser Arbeit soll die Anwendung der Methoden des ersten Teils in der Praxis gezeigt werden. In verschiedenen Anwendungsszenarien wurden die Möglichkeiten und die Grenzen dieser Methoden getestet. Insbesondere wurde geprüft, wie robust die vorgestellten Ansätze gegenüber Beleuchtungsvarianzen und Okklusionen sind.

Die hier verwendeten Testbilder wurden ein halbes Jahr nach Aufnahme der Trainingsbilder aufgenommen. Dies gewährleistet eine realistische Untersuchung, da in der Zwischenzeit geringfügige Veränderungen der Umgebungen stattfanden, wie sie in jeder realen, belebten Umgebung üblich sind. Dazu gehören einerseits die Veränderungen bedingt durch die Jahreszeit, beispielsweise kahle bzw. belaubte Bäume hinter den Fenstern, sowie unterschiedlicher Lichteinfall. Andererseits ändert sich auch häufig innerhalb der Umgebungen einiges, beispielsweise die Art und Anzahl von Postern im Flur oder das Vorhandensein von Möbeln. Außerdem können Lampen defekt sein oder die Sensoren des Roboters verändern sich geringfügig in ihrer Genauigkeit. So bekam das Schutzglas unserer Kamera in der Zwischenzeit einen leichten Riss, der sich auch in den Testbildern wiederfindet.

Die stärksten Veränderungen der Umgebung treten natürlich in Räumen auf. Dort werden häufig Möbel verstellt, Gegenstände hinzugefügt oder entfernt und andere gravierende Veränderungen getätigt. Um die Lokalisierung im Raum nicht nahezu unmöglich zu machen, haben wir uns bemüht, die Möbelaufstellung in den Zustand der Trainingsaufnahmen zu bringen sowie umherliegende Gegenstände zu entfernen.

Kapitel 6 stellt zunächst das Arbeitsszenario vor, in dem die Experimente durchgeführt wurden. Kapitel 7 präsentiert die Experimente der Vollbild-Lokalisierung, die in Kapitel 4 beschrieben wurde. Die Experimente der Sektoren-Lokalisierung aus Kapitel 5 werden in Kapitel 8 vorgestellt. Kapitel 9 schließt mit einem Vergleich von Vollbild- und Sektoren-Lokalisierung ab.

Kapitel 6

Das Arbeitsszenario

Das folgende Kapitel beschreibt das Arbeitsszenario dieser Diplomarbeit. Zunächst werden der Roboter und die omnidirektionale Kamera vorgestellt. Dann werden einige Beispielfotografien präsentiert, die mit dieser Kamera aufgenommen wurden und die Büroumgebung, in der der Roboter agiert. Schließlich wird beschrieben, wie die Experimente durchgeführt wurden, welche Ergebnisse berechnet wurden und wie diese in den folgenden Kapiteln präsentiert werden.

6.1 Roboter und omnidirektionale Kamera

Der in unseren Experimenten verwendete Roboter ist der autonome mobile Roboter *Kurt II* [Kur], eine Weiterentwicklung von *Kurt* (Kircher & Hertzbert [KH97]) – einer Roboterplattform für Steuerungs- und Sensorikexperimente in Kanalisationssystemen. Aufgrund seiner geringen Höhe eignet er sich gut für den Einsatz einer omnidirektionalen Kamera, da die mit dieser Kamera aufgenommenen Bilder auch die Bereiche enthalten, die sich in unmittelbarer Nähe des Roboters auf dem Boden befinden (siehe Beispielfotografien auf S. 56).

Abbildung 6.1 zeigt *Kurt II* mit obenauf montierter Rundumsichtkamera. Diese ist durch eine Vorrichtung erhöht angebracht, so daß der Roboter selbst nicht ins Blickfeld der Kamera gerät. Dadurch werden die aufgenommenen Bilder nicht durch Teile des Roboters beeinträchtigt.

Als Sensor wurde eine NAVICAM-Rundumsichtkamera der Firma Neuronics [nav] verwendet (siehe Abbildung 6.2). Sie ermöglicht Auflösungen von 330 bis zu 450 horizontalen TV-Linien. Ein konvexer Aluminiumspiegel erlaubt eine Verarbeitung visueller Information bis ca. 30° über und 58° unter der horizontalen Achse des Spiegels. Der Spiegel lenkt die Bildinformation zu der im unteren Gehäuse befindlichen Kamera um.

In der Mitte des Spiegels befindet sich eine zentrale Öffnung, die einen Blick nach oben gewährleistet. Dadurch wird auch Bildverarbeitung oberhalb des Spiegels liegender Bereiche ermöglicht. So können beispielsweise Deckenvariationen erkannt und zur Navigation genutzt werden. Falls nicht benötigt, wird die Öffnung durch einen Deckel verschlossen.



Abbildung 6.1: Der Roboter *Kurt II* mit aufmontierter Kamera

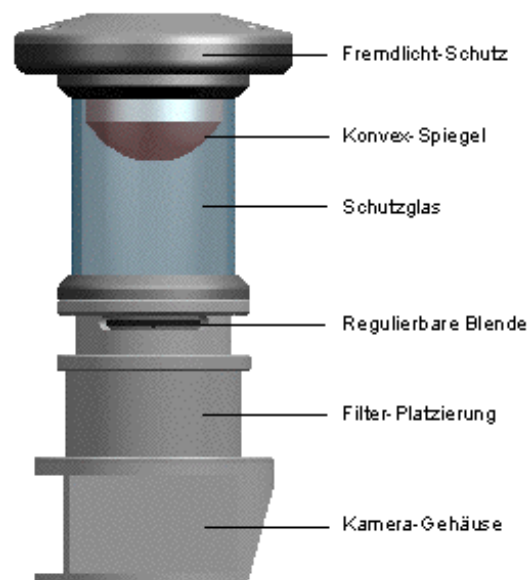


Abbildung 6.2: Die NAVICAM-Rundumsichtkamera

Die wichtigsten Daten der Kamera:

Durchmesser des Spiegels:	44 mm
Durchmesser der zentralen Öffnung:	10 mm
Entfernung Kameraoptik-Spiegel:	90 mm
Winkelverstärkung:	10
Sichtbereich über Horizont:	30°
Sichtbereich unter Horizont:	58°
Modul-Gesamthöhe:	190 mm
Modul-Gesamtbreite:	90 mm

6.2 Die Panoramabilder

In diesem Abschnitt werden die von der Kamera aufgenommenen Bilder präsentiert. Abbildung 6.3 zeigt auf der linken Seite ein solches omnidirektionales Rundumbild. Es ist ein Bitmap-Bild der Größe $768 \times 576 = 442\,368$ Pixel. Dieses Bild wurde in einem Büroflur aufgenommen. Im unteren Teil sieht man den hellen Fensterbereich, oben setzt sich der Gang fort und rechts und links sind Türen zu erkennen. Der Kreis in der Mitte entspricht der Position, an der der Roboter plaziert ist; er zeigt durch die zentrale Öffnung einen Blick nach oben zur Decke.

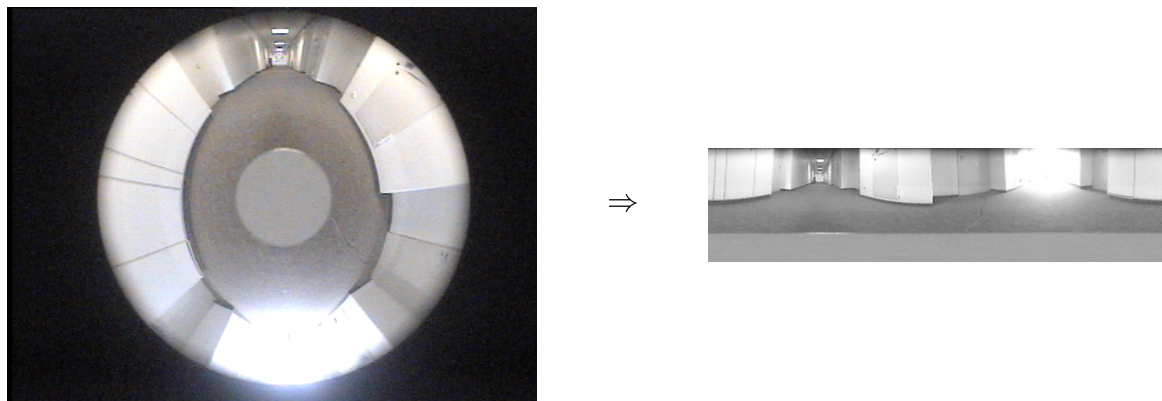


Abbildung 6.3: Ein Kamerabild und das transformierte Panoramabild

Dieses Kamerabild wird nun in ein rechteckiges Panoramabild transformiert, wie in Abbildung 6.3 rechts zu sehen ist. Hierbei findet eine erste Datenreduktion statt, da der schwarze Rand entfernt wird, das Farbbild in ein Schwarzweißbild umgewandelt wird und die Pixelanzahl auf ein gewünschtes Maß begrenzt wird. Es entsteht ein Sun-Rasterfile-Bild der Größe $360 \times 270 = 97\,200$ Pixel. Wenn die Deckeninformation abgeschnitten wird, entsteht ein Bild der Größe $360 \times 200 = 72\,000$ Pixel. Die Verzerrung des Bildes ist nicht von

Belang, da wir keine Merkmalsextraktion vornehmen, sondern Appearance based matching (siehe Kapitel 3.1).

Bei unseren Experimenten haben wir ausschließlich die transformierten Panoramabilder verwendet. Abbildung 6.4 zeigt als Beispiele einige weitere Panoramabilder, bei denen die Deckeninformation diesmal bereits abgeschnitten wurde.



Abbildung 6.4: Von links nach rechts: 1. Reihe: Tagbilder im Flur am Fenster, in der Gangmitte, im Raum; 2. Reihe: dieselben Bilder bei Nacht mit künstlicher Beleuchtung; 3. Reihe: Bilder mit verschieden starker Verdeckung durch Personen und offene Türen.

Die Bilddaten:

Kamerabild:	Bitmap	$768 \times 576 =$	442 368 Pixel
Panoramabild mit Decke:	Sun-Rasterfile	$360 \times 270 =$	97 200 Pixel
Panoramabild ohne Decke:	Sun-Rasterfile	$360 \times 200 =$	72 000 Pixel

6.3 Die Büroumgebung

Die Umgebung, in der unsere Experimente stattfanden, ist ein Korridor des Fraunhofer Instituts für Autonome Intelligente Systeme (AIS) in Sankt Augustin, Gebäude C2, Obergeschoss¹. Diese Umgebung entspricht einer typischen Indoor-Büroumgebung, die aus Fluren und Räumen besteht. Abbildung 6.5 zeigt eine Skizze dieser Büroumgebung. Die Punkte im Flur und einem der Räume markieren die Stellen, an denen bei unseren Experimenten Trainingsbilder aufgenommen wurden. Eine Büroumgebung besteht im allgemeinen aus Fluren und Räumen, somit bildet unsere Umgebung einen repräsentativen Ausschnitt einer Büroumgebung.

Diese Umgebung wurde nun in drei Abschnitte aufgeteilt, wie in Kapitel 4 beschrieben wurde. Abbildung 6.6 zeigt erneut diese drei Abschnitte. Jeder Abschnitt stellt einen repräsentativen Teil einer Büroumgebung dar. Die Lokalisierung fand jeweils in einem der

¹Inzwischen wird ein Gebäude an der Stirnseite (Fenster) des Flurs angebaut, so daß die exakte Situation jetzt nicht mehr reproduzierbar ist.

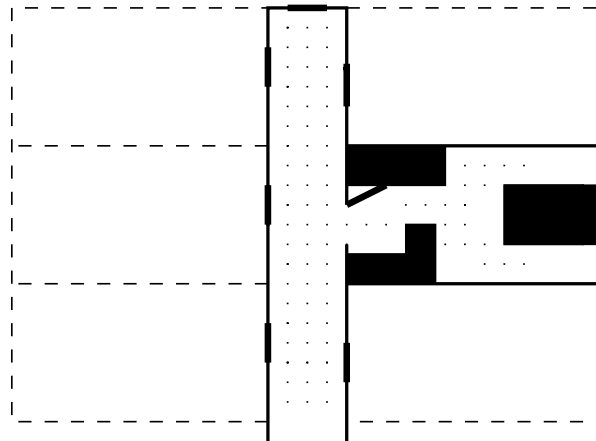
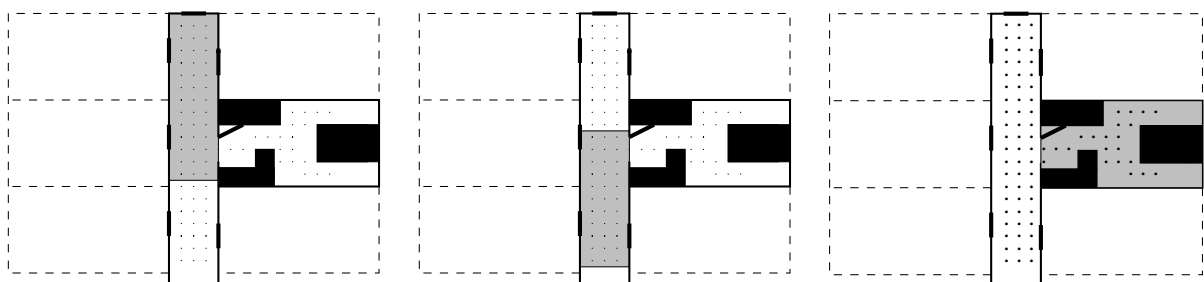


Abbildung 6.5: Die Bürsumgebung

Abschnitte statt, die im folgenden Umgebungen genannt werden. So wurden Experimente in Fensternähe, in der Flurmitte sowie im Raum durchgeführt.



(a) Umgebung 1: Flur am Fenster

(b) Umgebung 2: Flur ohne Fenster

(c) Umgebung 3: Raum

Abbildung 6.6: Die Umgebung in drei Abschnitte aufgeteilt

6.4 Fehlerauswertung und Präsentation

In diesem Abschnitt soll beschrieben werden, wie unsere Experimente durchgeführt wurden, welche Ergebnisse wir berechnet haben und wie diese Ergebnisse in den folgenden Kapiteln präsentiert werden.

6.4.1 Allgemeines

In unseren Experimenten wurde je ein Testszenario bestimmt, welches dann ausgewertet wurde. Ein Testszenario wird festgelegt durch die Umgebung, in der die Tests stattfanden, die Art des Eigenraums sowie die Menge der Testbilder. Zunächst wurde also eine der drei Umgebungen ausgewählt. Die meisten Tests wurden für alle drei Umgebungen durchgeführt; bei einigen reichte es dagegen, nur eine oder zwei Umgebungen zu betrachten. Dann wurde der Eigenraum bestimmt. Dieser hängt von der verwendeten Umgebung ab, zusätzlich wurde aber die Art der Trainingsbilder variiert. Bei den meisten Tests bestanden die Trainingsbilder nur aus Tagbildern, bei einigen Tests aber wurden zusätzlich Nachtbilder als Trainingsdaten verwendet. Die daraus generierten Eigenräume werden Tag-Nacht-Eigenräume genannt (siehe Kapitel 4.4).

Letztendlich festgelegt wurde das Testszenario durch die Testbilder. Diese Testdaten wurden so gewählt, daß jeweils ein bestimmter Aspekt getestet werden konnte. Dazu wurde eine Standardeinstellung bestimmt, von der in den Tests je ein Aspekt variiert wurde. Diese Standardeinstellung war: Bilder an Trainingspositionen, keine Rotationen, Tageslicht, geschlossene Türen und keine Okklusionen durch Gegenstände oder Personen. Um beispielsweise die Robustheit gegenüber Rotationen zu testen, wird eine Testbildmenge ausgewählt, in der nur dieser eine Aspekt (Rotation) variiert wird. Das zugehörige Testszenario wird beschrieben durch {Umgebung 1, Tag-Eigenraum, rotierte Bilder}.

Nachdem das Testszenario festgelegt wurde, wurden die Testbilder in den Eigenraum projiziert. Bei der Sektoren-Lokalisierung wurden die Testbilder zunächst in Sektoren aufgeteilt, welche dann in den Sektoren-Eigenraum projiziert wurden. Schließlich wurden die Abstände der Testpunkte zu den Trainingsbildprojektionen im Eigenraum bestimmt. Zu jedem Testbild haben wir also eine Folge von Trainingsbildern erhalten, sortiert nach ihrem Abstand zu diesem Testbild, sowie den zugehörigen Abstandswert. So könnte beispielsweise zu einem Testbild von Position (12,3) eine Folge von Trainingsbildern mit den zugehörigen Positionen ((13, 3), (12, 3), (12, 4), (12, 2), ...) generiert werden. Die zugehörigen Abstände dazu seien (0.24, 0.26, 0.29, 0.30, ...). Bei der Sektor-Lokalisierung erhält man keine Folge von Abständen, sondern von Wahrscheinlichkeiten. Da diese nicht aufsteigend, sondern absteigend sortiert sind, müssen in diesem Fall die reziproken Werte der Abstände verwendet werden. Im folgenden wird nur auf die Berechnungen für Abstände eingegangen.

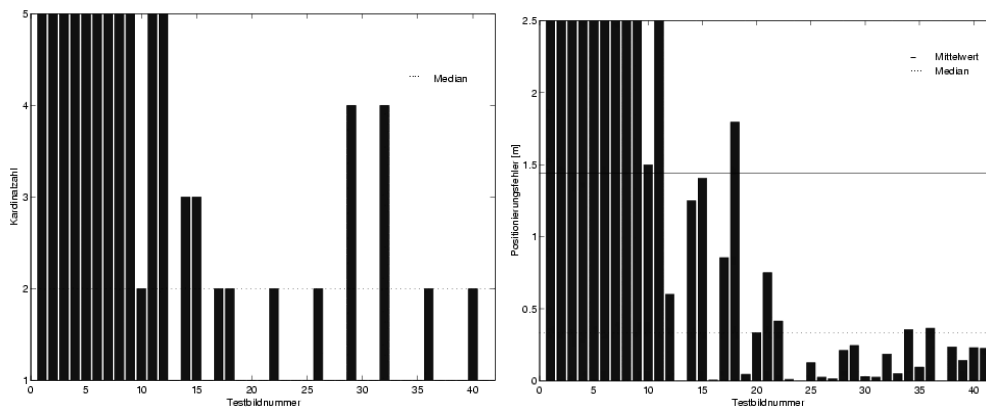
6.4.2 Bewertungskriterien

Aus der Liste der Nachbarn des Testbildes, sortiert nach ihrer Distanz, sowie der Liste dieser Distanzwerte wurden verschiedene Fehlermaße generiert, die im folgenden beschrieben werden sollen.

Kardinalzahl

Die Kardinalzahl gibt an, an welcher Stelle der geschätzten Folge von Positionen die tatsächliche Position steht. In unserem Beispiel hätte das Testbild also eine Kardinalzahl

von 2, da die tatsächliche Position an zweiter Stelle der Folge steht. Die Kardinalzahlen werden in den folgenden Kapiteln als Balkendiagramme dargestellt. Abbildung 6.7 (a) zeigt ein Beispiel eines solchen Diagramms eines Testszenarios. Für jedes Testbild gibt es einen Balken, dessen Höhe die Kardinalzahl angibt. Die vertikale Achse der Diagramme beginnt bei 1, da dies die kleinstmögliche Kardinalzahl ist und endet bei 5; alle Werte größer 5 wurden abgeschnitten, um sich auf die relevanten Werte zu beschränken. Bilder mit Kardinalzahl 1 werden im folgenden als *perfekt klassifiziert* bezeichnet. Außerdem wurden Mittelwerte und Mediane der Kardinalzahlen berechnet und unter der Grafik angegeben; zudem wurden sie als waagerechte Linien in der Grafik dargestellt, sofern der Wert innerhalb der dargestellten Grenzen liegt.



(a) Kardinalzahl (Mittelwert: 5.88, Median: 2)

(b) Positionierungsfehler (Mittelwert: 1.45, Median: 0.34)

Abbildung 6.7: Beispiele für die Fehlerdarstellung

Geschätzte Position

Desweiteren wurde für jedes Testbild eine geschätzte Position berechnet. Sie ergibt sich aus den nächsten drei Nachbarn des Testbildes und deren Abständen zu diesem im Eigenraum. Die Position eines Bildes wird als Koordinaten in der Umgebung ausgedrückt. Die x-Koordinate der geschätzten Position berechnet sich aus den x-Koordinaten der ersten drei Nachbarn, gewichtet mit einem Wert, der sich aus den Abständen der Nachbarn berechnet. Je näher sich der Nachbar im Eigenraum an dem Testbild befindet, desto höher das Gewicht. Die y-Koordinate berechnet sich analog. Sei (x_i, y_i) die Position des i-ten Nachbarn und d_i sein Abstand zu dem Testbild im Eigenraum. Dann berechnet sich das i-te Gewicht durch

$$w_i = \frac{1/d_i^2}{\sum_i 1/d_i^2}.$$

Die geschätzte Position (x_s, y_s) berechnet sich nun aus

$$x_s = w_1x_1 + w_2x_2 + w_3x_3$$

und

$$y_s = w_1y_1 + w_2y_2 + w_3y_3.$$

In unserem Beispiel ergeben sich die Gewichte $w_1 = 0.39$, $w_2 = 0.32$ und $w_3 = 0.27$, was zu einer geschätzten Position von $(12.39, 3.28)$ führt. Für einige Testszenarien werden die geschätzten Positionen in Grafiken dargestellt, denen eine Karte der Umgebung zugrunde liegt. Dies ist z.B. dann sinnvoll, wenn die Testbilder einem Testlauf durch die Umgebung entsprechen, so daß aus den geschätzten Positionen im Idealfall der zugrundeliegende Testlauf wiedererkannt werden kann. Abbildung 6.8 zeigt ein Beispiel der geschätzten Positionen eines solchen Testlaufs.

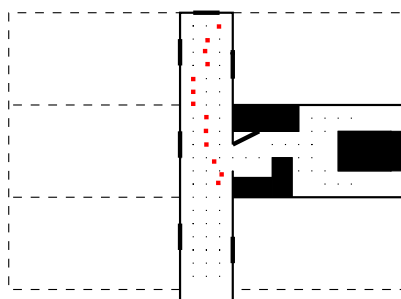


Abbildung 6.8: Geschätzte Positionen eines Testlaufs

Positionierungsfehler

Der Positionierungsfehler p berechnet sich aus dem euklidischen Abstand zwischen der tatsächlichen Position (x, y) und der geschätzten Position (x_s, y_s) :

$$p = \frac{1}{2} \sqrt{(x_s - x)^2 + (y_s - y)^2}.$$

Der Faktor $1/2$ dient dazu, aus dem Abstand eine Meterangabe zu gewinnen; da die Koordinatenpunkte, die den Positionswerten zugrundeliegen, nicht 1 m sondern 50 cm voneinander entfernt sind, ist dieser Faktor notwendig. In unserem Beispiel ergäbe sich somit ein Positionierungsfehler von 24 cm. Abbildung 6.7 (b) zeigt die Darstellung der Positionierungsfehler eines Testszenarios. Die Werte werden bei 2.5 abgeschnitten, da höhere Werte zu schlecht sind, als daß sie von Interesse wären. Anders als bei den Kardinalzahlen beginnen die Werte hier bei 0 statt bei 1. Mittelwerte und Mediane werden wie bei den Kardinalzahlen dargestellt.

Kapitel 7

Ergebnisse der Vollbild-Lokalisierung

Dieses Kapitel zeigt die Anwendung der in Kapitel 4 beschriebenen Vollbildmethode. Es beginnt in Abschnitt 7.1 mit einer allgemeinen Testreihe von Bildern, um die Funktionsweise der Methode unter günstigen Bedingungen zu zeigen. Abschnitt 7.2 präsentiert Schwierigkeiten im Umgang mit Beleuchtungsvarianzen sowie die Verbesserungen durch Verwendung von Tag-Nacht-Eigenräumen. Abschnitt 7.3 zeigt schließlich die Probleme, die auftreten, wenn Teile der Bilder durch Hindernisse verdeckt sind.

7.1 Allgemeine Tests

Zunächst wurde die Methodik allgemein anhand von Testbildern geprobt. Dazu wurde ein Testlauf des Roboters simuliert, der durch den gesamten Gang der Büroumgebung führt (*Umgebungen 1 und 2*). Entlang dieses Testlaufs wurden Bilder sowohl an den Trainingspositionen als auch zwischen diesen aufgenommen. Letztere sollen dazu dienen, die Fähigkeit des Verfahrens zur Interpolation zu testen. Ebenso wurden einige Testbilder im Raum (*Umgebung 3*) aufgenommen. Abbildung 7.1 zeigt die Testpositionen aller drei Umgebungen.

In diesem Abschnitt werden zunächst die Ergebnisse gezeigt, die die Lokalisierung von Bildern an Trainingspositionen ergeben hat. Danach folgt eine Darstellung der Klassifikation von Bildern der Zwischenräume. Schließlich wird gezeigt, wie die Vollbild-Lokalisierung auf rotierte Bilder reagiert.

7.1.1 Tests an Trainingspositionen

Betrachten wir also zunächst nur die Bilder, die an den Trainingspositionen aufgenommen wurden. In Abbildung 7.2 werden die geschätzten Positionen dargestellt, die die Lokalisierung in den drei Umgebungen ergeben hat. Man erkennt deutlich, daß die geschätzten Testläufe in den Umgebungen 1 und 2 kaum von dem realen Pfad abweichen. Die Testdaten im Raum zeigen dagegen größere Abweichungen. Genauere Informationen liefern die Werte der Kardinalzahlen und Positionierungsfehler für die drei Umgebungen, die in den Abbildungen 7.3 bis 7.6 dargestellt werden.

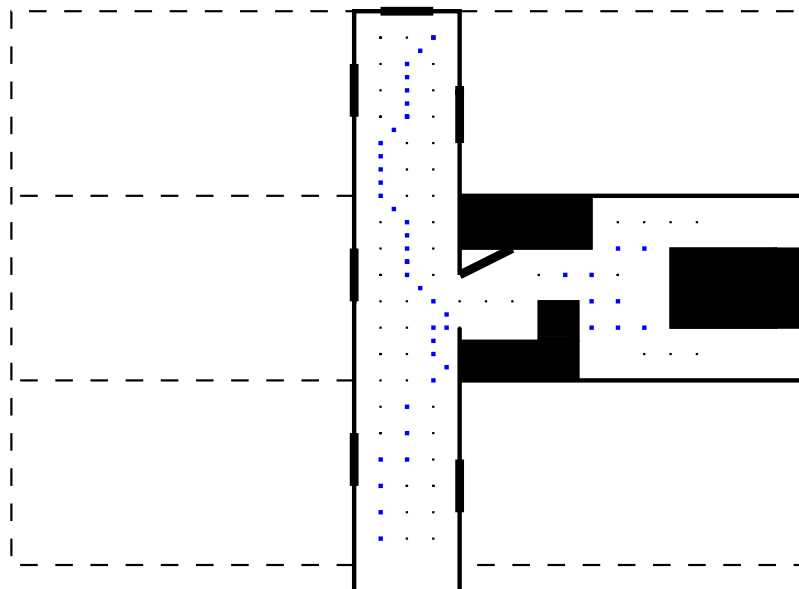
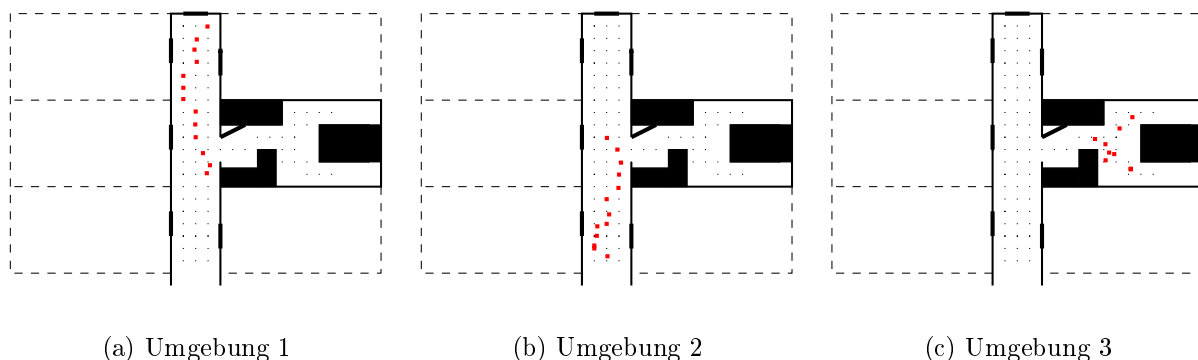


Abbildung 7.1: Die Testpositionen

Wenn wir die Kardinalzahlen betrachten (Abbildung 7.3 (a) und 7.4 (a)), zeigt sich, daß in den Umgebungen 1 und 2 alle Bilder mit nur einer Ausnahme perfekt zugeordnet wurden. Bei dieser Ausnahme handelt es sich um dasselbe Bild. Dies resultiert aus der Überlappung der Umgebungen. Das Bild wird in den beiden Umgebungen verschieden gut klassifiziert: In Umgebung 1 ergibt sich die Kardinalzahl 4 und ein Positionierungsfehler von 25 cm. In Umgebung 2 ergeben sich mit der Kardinalzahl 2 und einem Positionierungsfehler von 7 cm deutlich bessere Werte. Dieses Beispiel verdeutlicht, daß die Güte der Klassifikation eines Testbildes nicht nur von der Qualität des Bildes, sondern auch von den zugrundeliegenden Trainingsbildern abhängt.

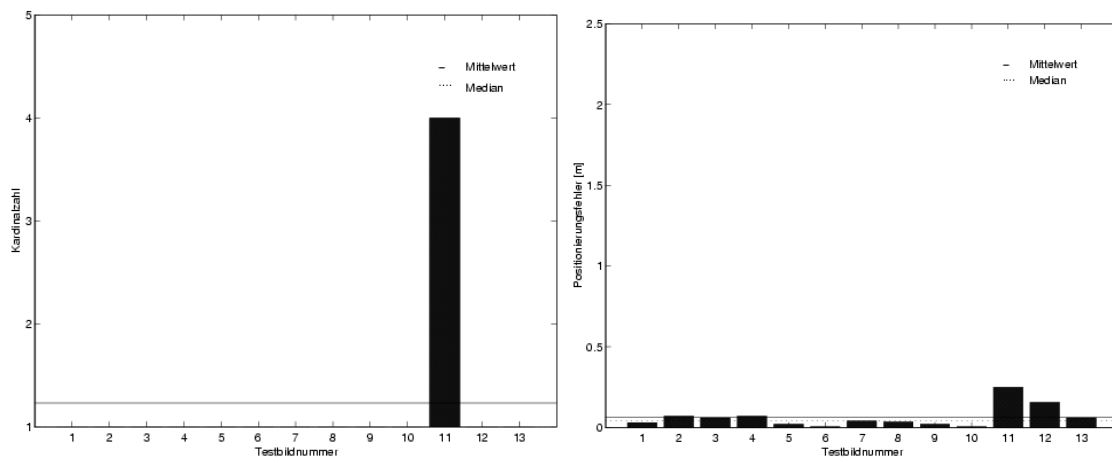


(a) Umgebung 1

(b) Umgebung 2

(c) Umgebung 3

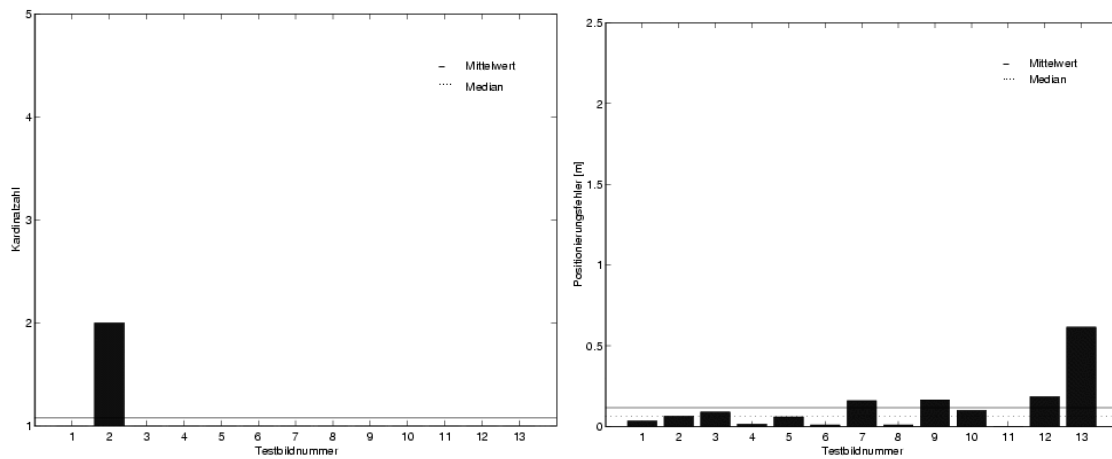
Abbildung 7.2: Die geschätzten Positionen der Testaufnahmen



(a) Kardinalzahl (Mittelwert: 1.23, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.07, Median: 0.04)

Abbildung 7.3: Klassifikation des Testlaufs – Umgebung 1



(a) Kardinalzahl (Mittelwert: 1.08, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.12, Median: 0.07)

Abbildung 7.4: Klassifikation des Testlaufs – Umgebung 2

Wenn man das schlechter klassifizierte Bild genauer betrachtet (Abbildung 7.5), zeigt sich, warum die Zuordnung hier Probleme bereitet. Durch einfallendes Sonnenlicht ist die Region des Flurfensters deutlich heller als bei dem zugehörigen Trainingsbild, so daß die Klassifikation erschwert wird.



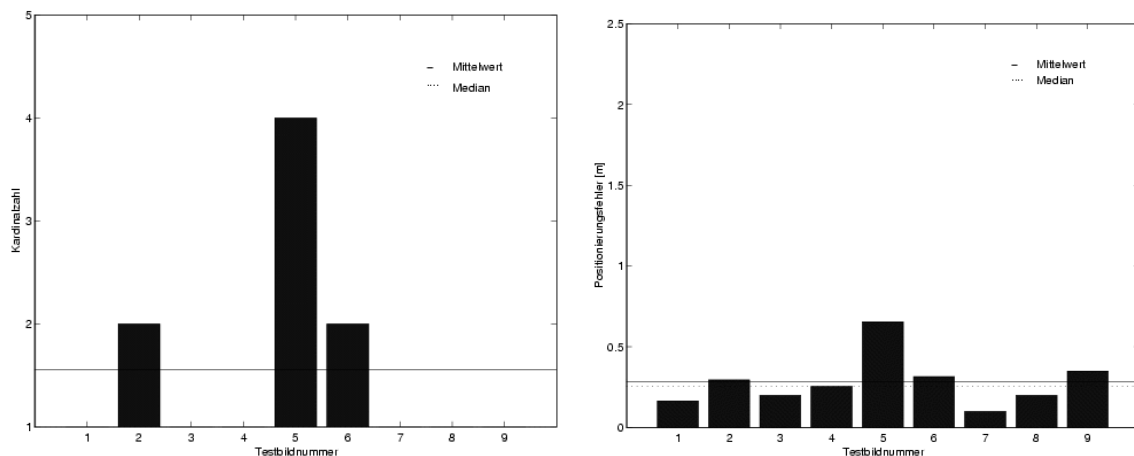
(a) Ein schlecht klassifiziertes Testbild

(b) Das zugehörige Trainingsbild

Abbildung 7.5: Erschwerte Klassifikation durch Beleuchtungsvarianzen

Insgesamt liefert der Testlauf in den beiden Umgebungen jedoch sehr gute Ergebnisse, was besonders durch Mediane der Kardinalzahlen von 1 sowie Mediane der Positionierungsfehler von 0.04 und 0.07 verdeutlicht wird.

Im Raum sind die Ergebnisse ein wenig schlechter. Dies läßt sich leicht erklären, da die Testbilder wie bereits erwähnt ein halbes Jahr später aufgenommen wurden als die Trainingsbilder und in dieser Zeit viele Veränderungen in einem Raum stattfinden. Trotzdem werden zwei Drittel der Bilder perfekt zugeordnet, wie Grafik 7.6 zeigt, und selbst die übrigen Bilder liefern mit Kardinalzahlen von 2 und 4 akzeptable Werte. Der Positionierungsfehler liefert durchschnittlich schlechtere Werte als die anderen Umgebungen. Daß dies auch bei Bildern mit guter Kardinalzahl der Fall ist, liegt daran, daß sich benachbar-



(a) Kardinalzahl (Mittelwert: 1.56, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.28, Median: 0.26)

Abbildung 7.6: Klassifikation der Testaufnahmen – Umgebung 3

te Bilder im Raum stärker unterscheiden als im Flur. Selbst wenn das Testbild also dem richtigen Trainingsbild zugeordnet wird, liegen das zweit- und drittnächste Trainingsbild häufig weiter entfernt. Da der Positionierungsfehler aus den drei nächstgelegenen Bildern berechnet wird, ergeben sich so etwas schlechtere Werte. Eine Möglichkeit zur Verbesserung könnte es sein, im Raum das nächstgelegte Bild noch stärker zu gewichten.

7.1.2 Interpolation

In der Realität befindet sich der Roboter meist nicht an den Positionen, an denen Trainingsbilder aufgenommen wurden, sondern an beliebigen Positionen. Daher ist es von Interesse, die Fähigkeit des Verfahrens zur Interpolation zu testen. In Abbildung 7.1 wurden bereits die Positionen präsentiert, an denen Testbilder zu diesem Zweck aufgenommen wurden; diese Bilder werden im folgenden *interpolierte Bilder* genannt. Abbildung 7.7 zeigt in (a) die geschätzten Positionen nur der interpolierten Bilder, in (b) sind die geschätzten Positionen des kompletten Testlaufs zusammengefaßt. Der Weg des Testlaufs ist weiterhin deutlich zu erkennen, die geschätzten Positionen weichen nur geringfügig von den tatsächlichen ab.

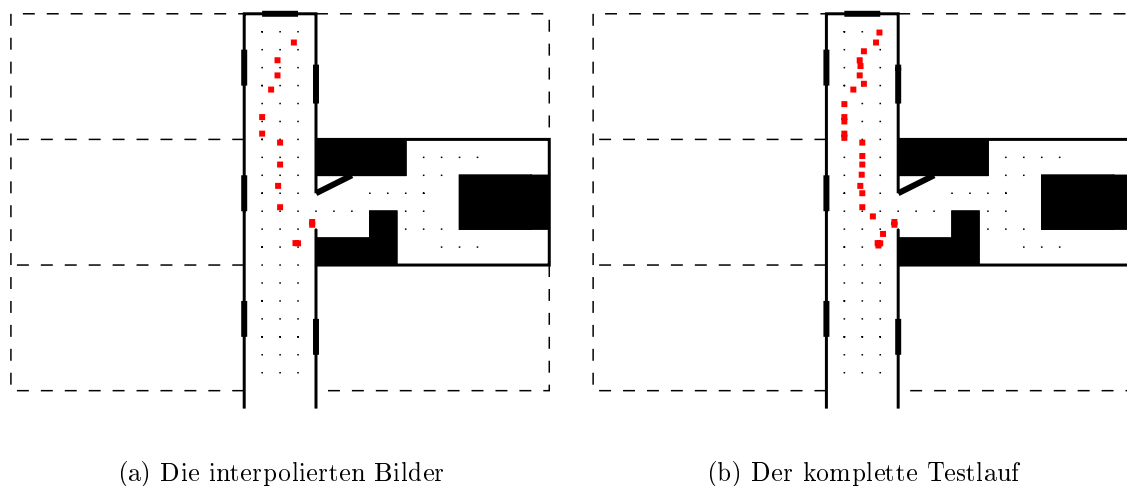
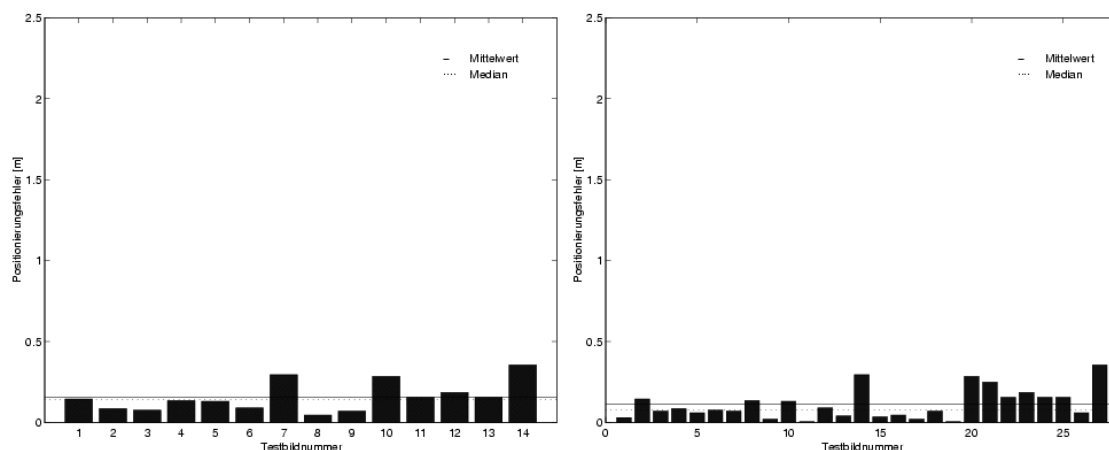


Abbildung 7.7: Die geschätzten Positionen der interpolierten Testaufnahmen

Abbildung 7.8 zeigt die Positionierungsfehler der interpolierten Bilder (a) und des gesamten Testlaufs (b). In (a) sieht man, daß der Positionierungsfehler der interpolierten Bilder durchschnittlich zwar deutlich schlechter ist als der der Bilder von Trainingspositionen (was nicht verwundern dürfte), mit einem Mittelwert von 32 cm aber dennoch überraschend gut ist.

Eine genauere Analyse der Testergebnisse zeigt, daß die Fähigkeit des Roboters zur Interpolation besser ist, wenn die Positionsverschiebung in Längsrichtung des Flurs stattfindet als in der Querrichtung. Dies erklärt sich dadurch, daß sich benachbarte Bilder in Längsrichtung ähnlicher sehen als dies bei benachbarten Bildern in Querrichtung der Fall ist.



(a) Interpolierte Bilder (Mittelwert: 0.16, Median: 0.14)

(b) Kompletter Testlauf (Mittelwert: 0.11, Median: 0.08)

Abbildung 7.8: Die Positionierungsfehler der interpolierten Testaufnahmen

Wird also ein Bild zwischen zwei in Längsrichtung benachbarten Bildern aufgenommen, läßt der Roboter sich nicht so leicht „verwirren“ wie im selben Fall in Querrichtung.

7.1.3 Rotationen

Die bisher betrachteten Bilder wurden alle unter den Voraussetzungen aufgenommen, daß der Roboter in der gleichen Richtung ausgerichtet war wie bei Aufnahme der Trainingsbilder. So können Testbilder direkt den Trainingsbildern zugeordnet werden. In der Realität sind Testbilder aber um beliebige Winkel rotiert. Dadurch unterscheiden sie sich erheblich von den Trainingsbildern, was die Klassifikation erschwert oder gar unmöglich macht. Abbildung 7.9 zeigt ein um 30° rotiertes Bild sowie das zugehörige nicht rotierte Bild.



(a) Um 30° rotiertes Bild

(b) Das nicht rotierte Trainingsbild

Abbildung 7.9: Rotiertes und nicht rotiertes Bild

Wo das menschliche Auge leicht die Ähnlichkeit erkennt, bekommt die vorgestellte Lokalisierungsmethode Probleme. Abbildung 7.10 zeigt Kardinalzahlen und Positionierungsfehler von Testbildern, die um 30° rotiert wurden. Man sieht, daß die Klassifizierung sehr

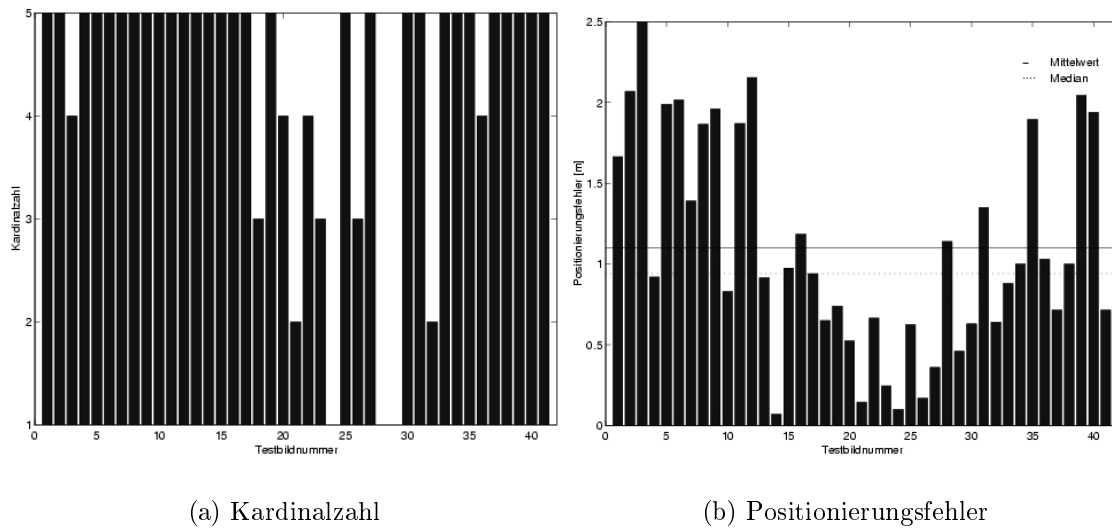


Abbildung 7.10: Klassifikation um 30° rotierter Bilder

schlecht ausfällt. Um zu untersuchen, bis zu welcher Rotation die Vollbild-Lokalisierung robust ist, wurden verschiedene Rotationen von 5° bis 50° getestet. Es hat sich gezeigt, daß die Vollbild-Lokalisierung gegenüber Bildern, die um 5° rotiert wurden, robust ist; die Bilder werden perfekt klassifiziert. Auch Rotationen um 10° liefern sehr gute Ergebnisse: 95% der Bilder werden perfekt klassifiziert. Bei größeren Rotationen nimmt die Erkennungsrate dann drastisch ab. Abbildung 7.11 zeigt die Mediane von Kardinalzahlen und Positionierungsfehlern der rotierten Bilder.

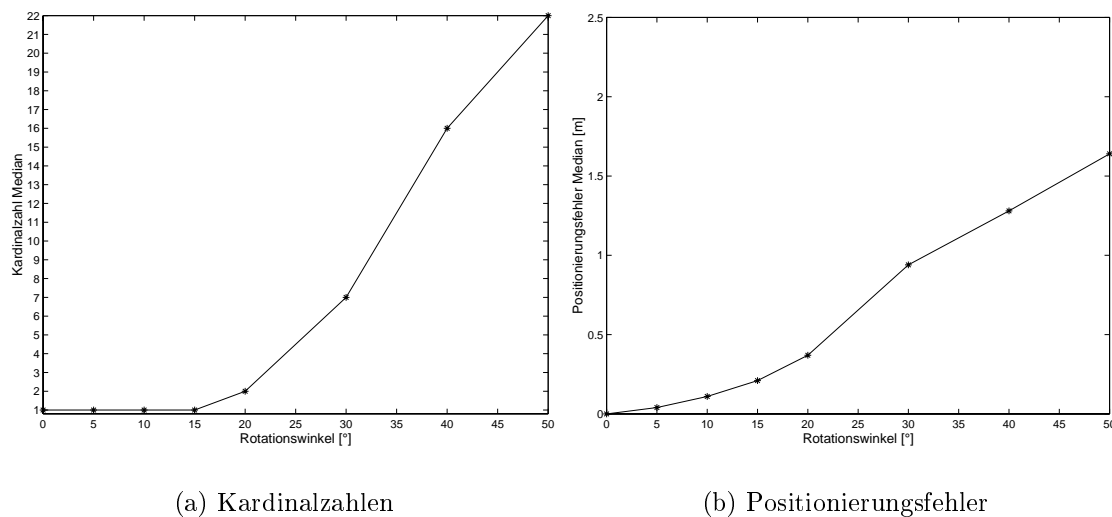


Abbildung 7.11: Mediane der Lokalisierung rotierter Bilder. Pro Rotationswinkel wurden 41 Bilder klassifiziert.

Hier zeigt sich, daß der Großteil der Bilder bis zu Rotationen von 15% perfekt zugeordnet werden. Um beliebig rotierte Bilder klassifizieren zu können, kann die Vollbild-Lokalisierung entweder um die auf S. 34 beschriebenen Methoden ergänzt werden, oder es kann die Sektoren-Lokalisierung verwendet werden, die eine größere Rotationsunabhängigkeit gewährleistet (siehe Kapitel 8).

7.2 Beleuchtungsvarianzen

Wenn Lokalisierung mit visuellen Sensoren betrieben wird, kann unterschiedliche Beleuchtung einen enormen Störfaktor der Sensordaten darstellen. Schon der Unterschied zwischen direkter Sonneneinstrahlung gegenüber bewölktem Himmel kann ein Bild stark verändern. Ein weiterer deutlicher Unterschied besteht zwischen Bildern, die bei Tag aufgenommen wurden, gegenüber solchen, die bei Dunkelheit mit künstlicher Beleuchtung aufgenommen wurden. Abbildung 7.12 zeigt auf der linken Seite drei bei Nacht aufgenommene Bilder und rechts daneben die entsprechenden Trainingsbilder derselben Positionen bei Tag.

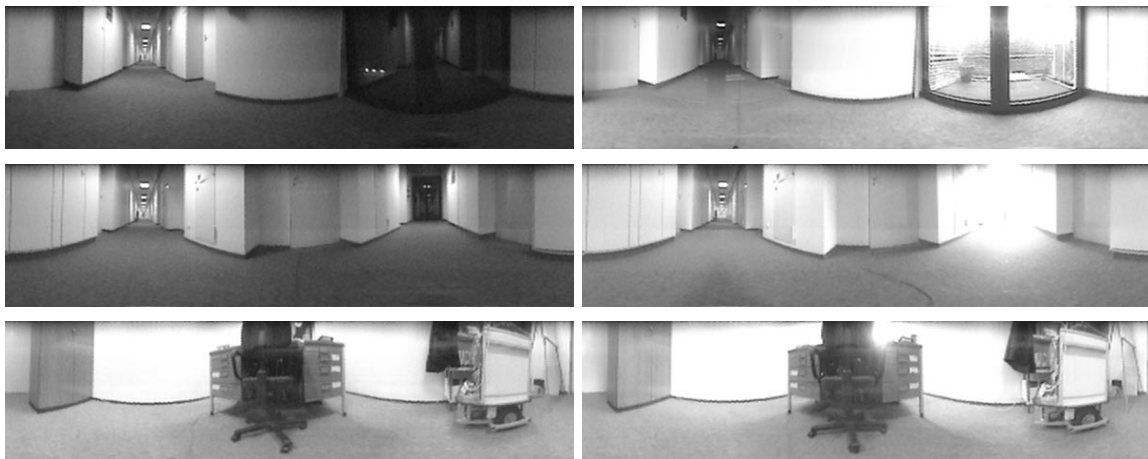


Abbildung 7.12: Links Nachtbilder, rechts die entsprechenden Trainingsbilder bei Tag

Die deutlichsten Unterschiede erkennt man bei dem obersten Bild, welches in unmittelbarer Nähe des Flurfensters aufgenommen wurde. Es leuchtet ein, daß sich hier Helligkeitsunterschiede von draußen am stärksten bemerkbar machen. Das mittlere Bild wurde in der Mitte des Flurs aufgenommen. Hier sind die Unterschiede bereits weniger gravierend. Die geringsten Differenzen finden sich in dem untersten Bild. Dieses wurde innerhalb des Raumes aufgenommen. Da die Kamera sich nur etwa einen halben Meter über dem Boden befindet, ist das Fenster nicht in dem unmittelbaren Sichtfeld des Roboters, da es durch Möbel verdeckt wird. Dadurch spiegeln sich Helligkeitsunterschiede weniger deutlich in den Bilddaten wieder.

Es liegt nun die Vermutung nahe, daß die Klassifikation von Nachtbildern im Raum weniger stark beeinträchtigt wird als im Flur und daß im Flur die Probleme ansteigen, je

näher sich der Roboter an einem Fenster befindet. In diesem Abschnitt soll zunächst untersucht werden, ob sich diese Vermutung bei unserer Lokalisierungsmethodik bestätigt. Dazu werden zunächst Nachtbilder aller Positionen mittels eines Tag-Eigenraums klassifiziert. Weiterhin wurde ein Testlauf bei Nacht generiert, der einen direkten Vergleich mit dem schon beschriebenen Tagtestlauf ermöglicht. Schließlich wurde ein Tag-Nacht-Eigenraum generiert, um Verbesserungen in der Klassifizierung zu erlangen. Getestet wurde dieser Eigenraum anhand der bekannten Testläufe bei Tag und bei Nacht. Der Abschnitt schließt mit einem Vergleich der Tag-Eigenräume mit den Tag-Nacht-Eigenräumen ab.

7.2.1 Nachtbilder

Um zu testen, wie eine normale Menge von Tag-Trainingsdaten mit Nachtbildern umgeht, wurde an jeder Position der Trainingsbilder ein Bild bei Nacht aufgenommen. Diese Bilder wurden dann in den üblichen Tag-Eigenraum projiziert. Abbildung 7.13 zeigt Kardinalzahl und Positionierungsfehler dieser Daten für Umgebung 1.

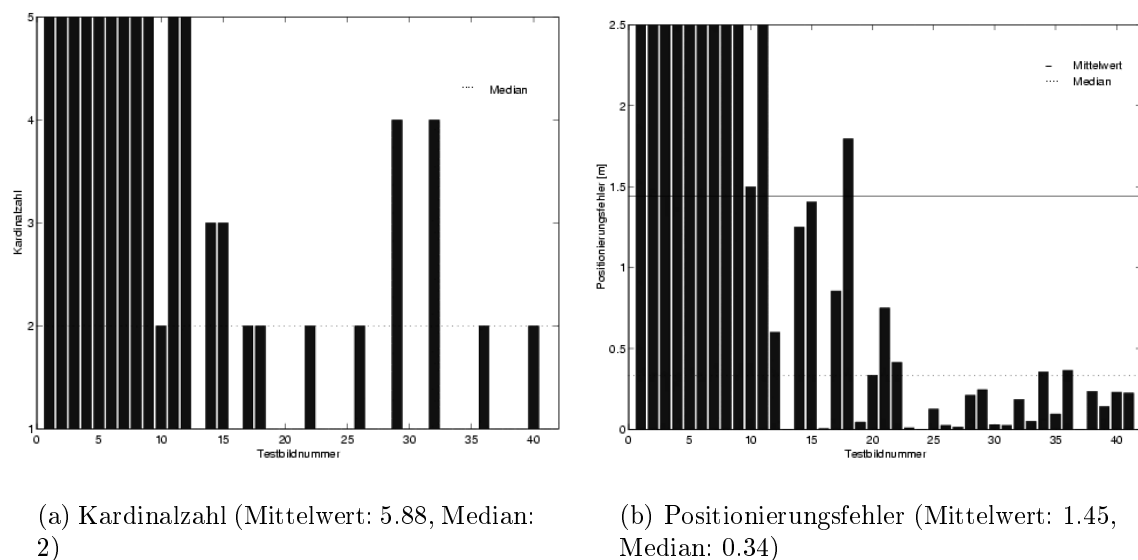


Abbildung 7.13: Nachtbilder – Umgebung 1

Man erkennt deutlich, daß auf der linken Seite sehr schlechte Ergebnisse stehen, die nach rechts hin immer besser werden. Die links stehenden Daten entsprechen Bildern, die in der Nähe des Fensters aufgenommen wurden. Daten auf der rechten Seite entsprechen Bildern in der Flurmitte. Die Relation zwischen Güte der Klassifikation und Position im Flur zeigt sich noch deutlicher in Abbildung 7.14. Hier ist der Positionierungsfehler bezüglich der Position dargestellt. Der Grundriß stellt den Flur da; jede Position wird hier durch einen Balken repräsentiert, wobei sich auf der linken Seite das Fenster befindet. Man erkennt, daß die drei Reihen, die direkt am Fenster liegen, sehr schlechte Ergebnisse liefern. Es folgen

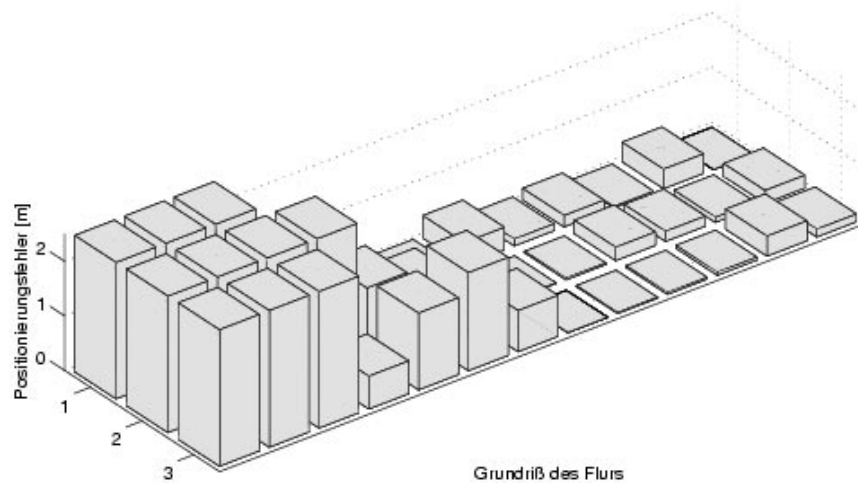
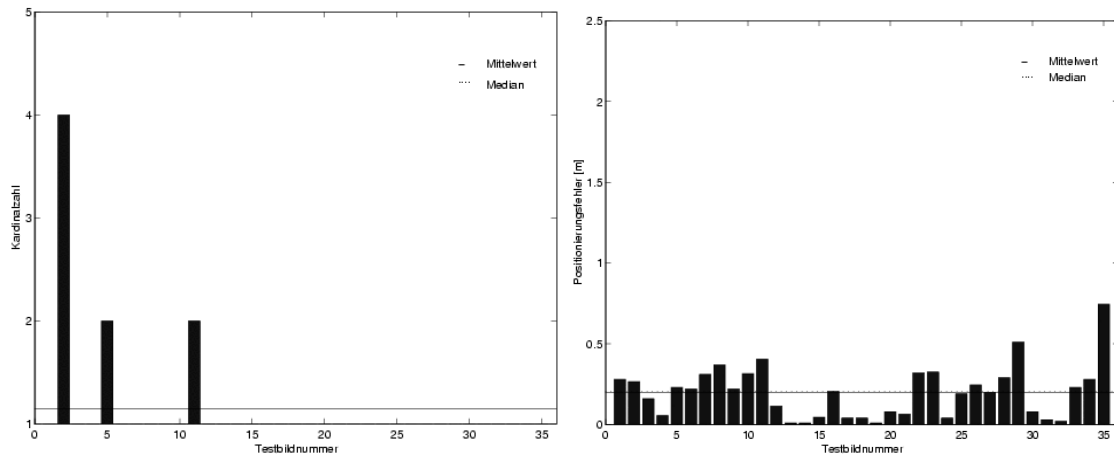


Abbildung 7.14: Die Positionierungsfehler der Nachtbilder bezüglich der zugrundeliegenden Umgebungskarte

vier weitere Reihen mit mittelmäßigen Ergebnissen und die übrigen sechs Reihen zeigen akzeptable Werte.

Da die Bilder in der Flurmitte recht gute Ergebnisse liefern, liegt die Vermutung nahe, daß Umgebung 2 deutlich bessere Ergebnisse liefern sollte als Umgebung 1. Daß diese Vermutung zutrifft, zeigt Abbildung 7.15. Hier sehen wir, daß die Kardinalzahlen in Umgebung 2 kaum schlechter ausfallen als bei reinen Tagbildern: Nur drei Bilder werden nicht beim ersten Versuch erkannt, der Mittelwert der Kardinalzahlen liegt mit 1.14 kaum über dem der Tagbilder (Mittelwert Tagbilder: 1.08). Der Positionierungsfehler fällt ein wenig schlechter aus; mit einem Mittelwert von 0.20 m und einem Median von 0.21 m liegen die Werte in akzeptablem Bereich, sind aber dennoch etwa doppelt so hoch wie die der reinen Tagdaten (Mittelwert: 0.12 m, Median: 0.07 m).

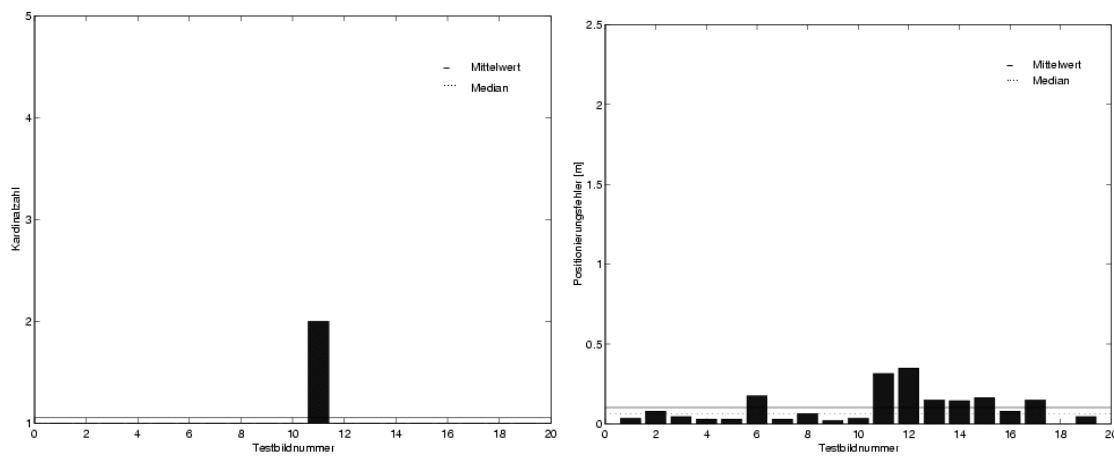
Im Raum fallen die Ergebnisse der Nachtbilder sogar sehr gut aus, wie Abbildung 7.16 zeigt. Nur ein Bild wird nicht perfekt zugeordnet, und der mittlere Positionierungsfehler liegt bei 0.11 m, der Median bei 0.07 m. Dies überrascht, da diese Werte sogar deutlich besser sind als die Werte des Tagtestlaufs, bei denen der mittlere Positionierungsfehler bei 0.28 m und der Median bei 0.26 m liegen. Erklären läßt sich dies dadurch, daß die Nachtdaten am selben Datum aufgenommen wurden wie die Tag-Trainingsbilder und nicht, wie die Testläufe, ein halbes Jahr später. Eine Veränderung des Raumes ist also kaum gegeben. Für einen direkten Vergleich der Testdaten bei Tag und bei Nacht verweise ich auf den Nachtttestlauf im nächsten Abschnitt.



(a) Kardinalzahl (Mittelwert: 1.14, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.20, Median: 0.21)

Abbildung 7.15: Nachtbilder – Umgebung 2



(a) Kardinalzahl (Mittelwert: 1.05, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.11, Median: 0.07)

Abbildung 7.16: Nachtbilder – Umgebung 3

7.2.2 Testlauf bei Nacht

In diesem Abschnitt folgt die Lokalisierung eines Nachttestlaufs, der zum gleichen Zeitpunkt aufgenommen wurde wie der Tagtestlauf. So läßt sich ein direkter Vergleich ziehen. Abbildung 7.17 zeigt die geschätzten Positionen in den drei Umgebungen.

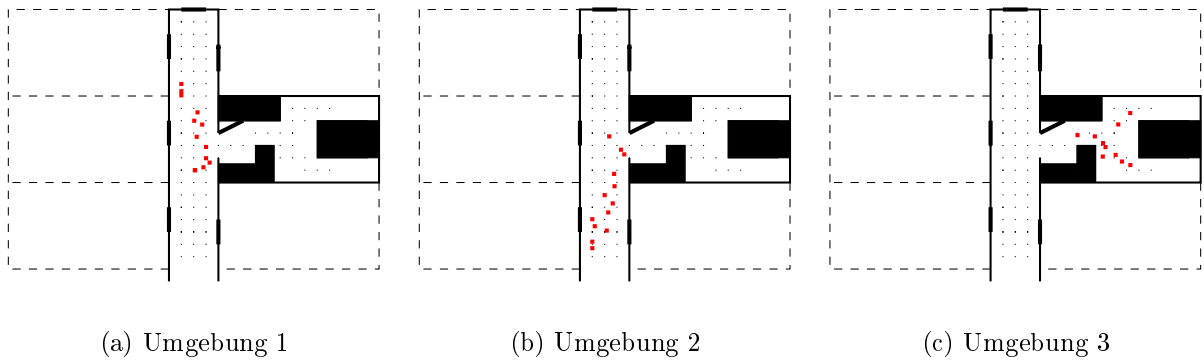
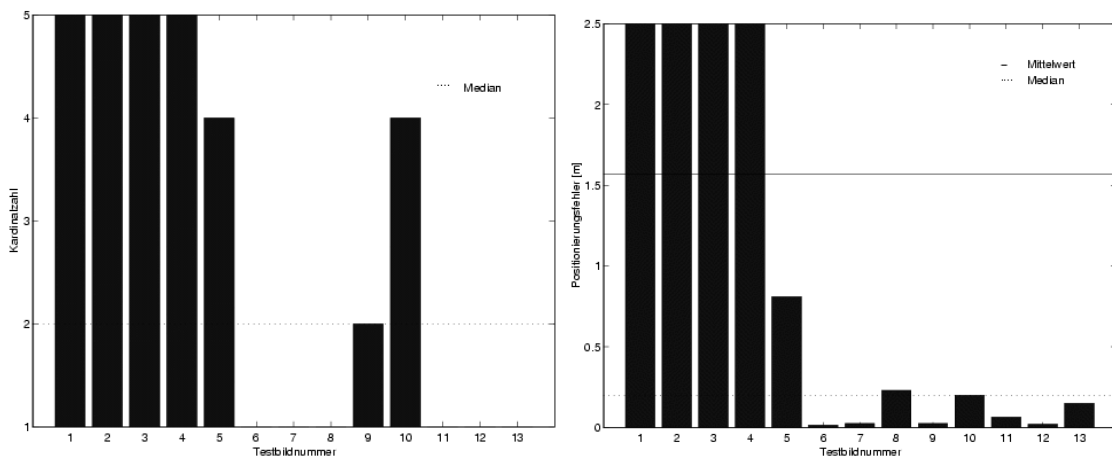


Abbildung 7.17: Die geschätzten Positionen der Testaufnahmen

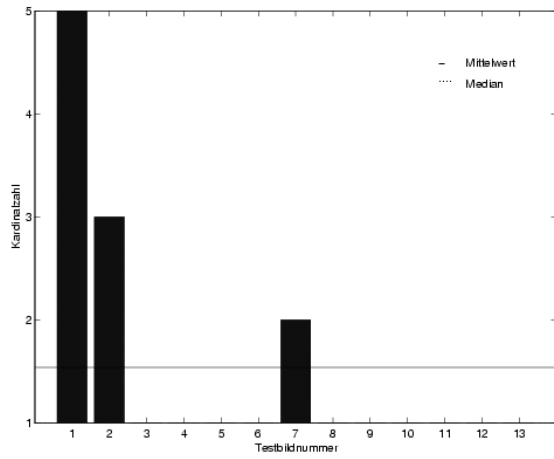
Auch hier sieht man deutlich, daß in Umgebung 1 die Bilder in Fensternähe völlig falsch zugeordnet werden. Sie werden allesamt Bildern in der Flurmitte zugewiesen, da diese eine geringere Helligkeit aufweisen und so den Nachtbildern stärker ähneln. In den Umgebungen 2 und 3 ergeben sich bessere Werte. Trotzdem sind in Umgebung 2 die geschätzten Positionen deutlich ungenauer als bei dem Tagtestlauf. Die Kardinalzahlen und Positionierungsfehler des Nachttestlaufs, die in den Abbildungen 7.18 bis 7.20 dargestellt sind,



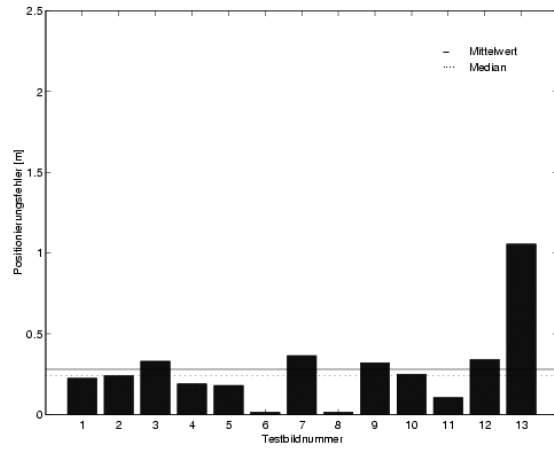
(a) Kardinalzahl (Mittelwert: 7.77, Median: 2)

(b) Positionierungsfehler (Mittelwert: 1.57, Median: 0.20)

Abbildung 7.18: Testlauf bei Nacht – Umgebung 1

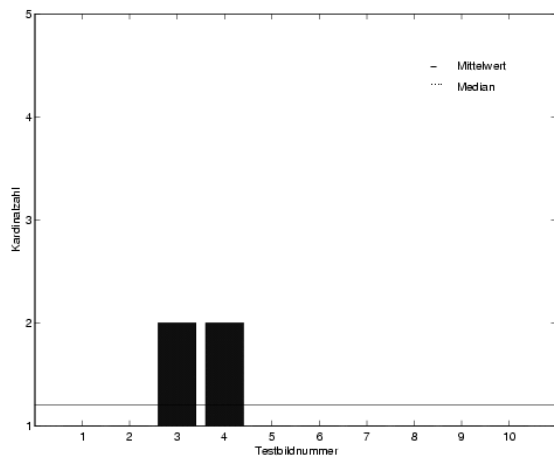


(a) Kardinalzahl (Mittelwert: 1.54, Median: 1)

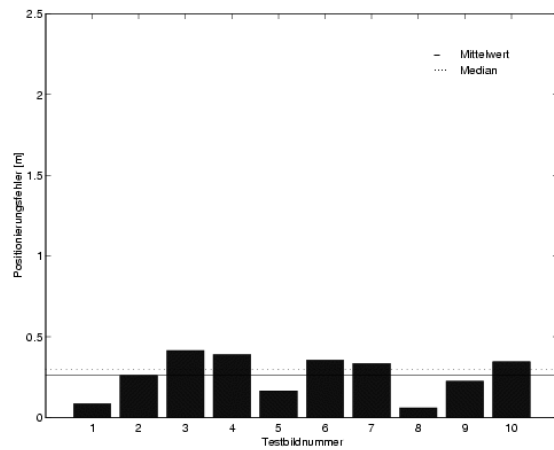


(b) Positionierungsfehler (Mittelwert: 0.28, Median: 0.24)

Abbildung 7.19: Testlauf bei Nacht – Umgebung 2



(a) Kardinalzahl (Mittelwert: 1.2, Median: 1)



(b) Positionierungsfehler (Mittelwert: 0.27, Median: 0.28)

Abbildung 7.20: Testaufnahmen bei Nacht – Umgebung 3

ähneln sehr stark den Ergebnissen der Nachtbilder und sollen daher hier nicht weiter kommentiert werden. Es sei nur angemerkt, daß die Werte des Raums jetzt sehr viel näher an den Klassifikationsergebnissen des Tagtestlaufs liegen, da die Daten zum gleichen Zeitpunkt aufgenommen wurden.

Insgesamt kann man sagen, daß sich die anfangs getätigte Vermutung bestätigt: Nachtbilder führen im Raum zu recht guten Klassifikationsergebnissen, im Flur ergeben sich Probleme, die umso stärker sind, je näher das Bild sich in Fensternähe befindet.

7.2.3 Tag-Nacht-Eigenraum

Wie wir im letzten Abschnitt gesehen haben, bereitet die Klassifikation von Nachtbildern je nach Aufnahmeposition mittel- bis sehr große Probleme. Um diesen Problemen beizukommen, wurde ein Tag-Nacht-Eigenraum generiert (siehe Kapitel 4.4). Bei diesem bestehen die Trainingsdaten sowohl aus Tag- als auch aus Nachtbildern. Die Klassifikation mittels dieser Eigenräume wurde an dem Tag- und an dem Nachttestlauf geprüft. Es wurden nur die Umgebungen 1 und 2 in die Tests einbezogen, da die Klassifikation von Nachtbildern im Raum keine großen Probleme bereitet, wie wir im letzten Abschnitt gesehen haben. Beginnen wir mit den Ergebnissen des Tagtestlaufs.

Testlauf bei Tag

Abbildung 7.21 zeigt die geschätzten Positionen des Tagtestlaufs. Man erkennt deutlich den Pfad, der dem Testlauf zugrunde lag. Auch die Abbildungen 7.22 und 7.23, die Kardinalzahlen und Positionierungsfehler darstellen, zeigen, daß der Tagtestlauf gute Klassifikationsergebnisse liefert. Einen direkten Vergleich mit den Daten des Tag-Eigenraums liefert Kapitel 7.2.4.

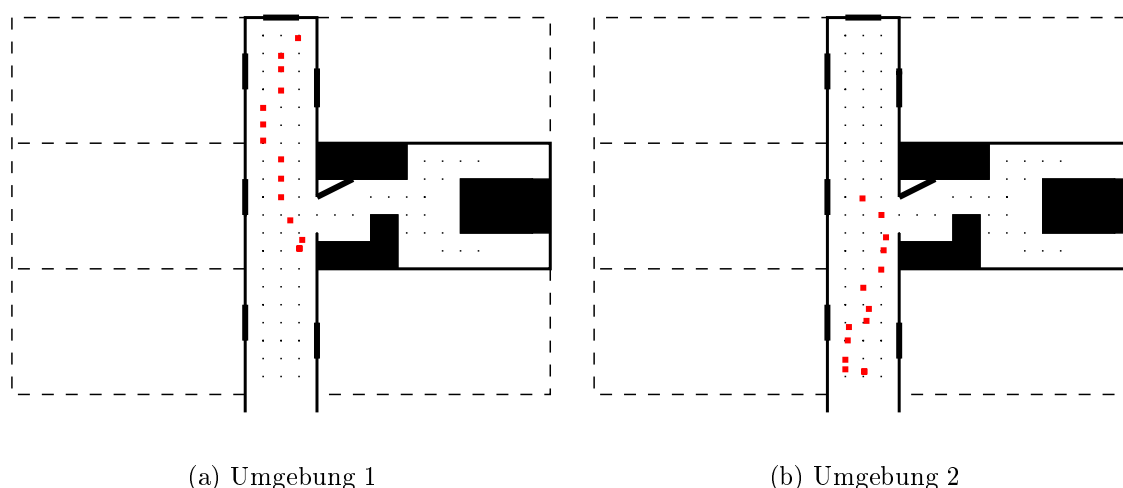
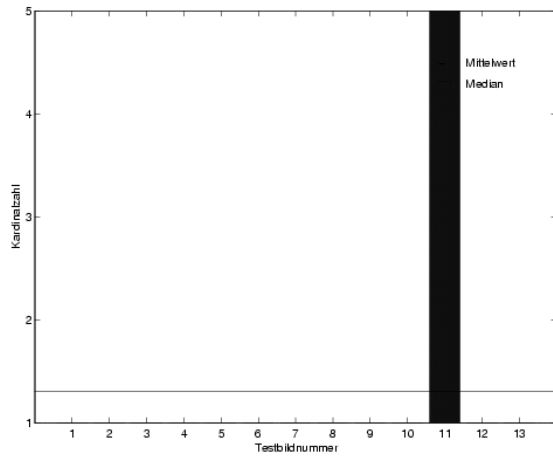
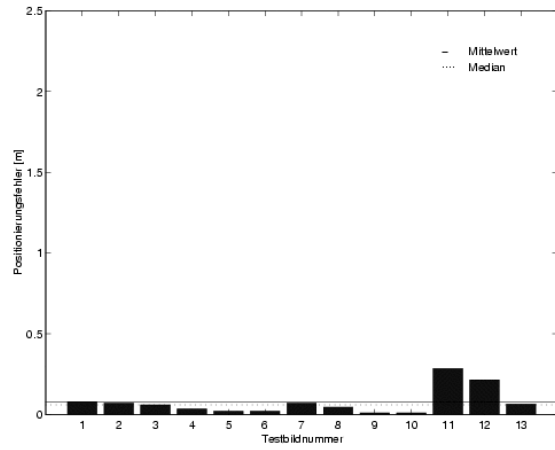


Abbildung 7.21: Die geschätzten Positionen des Testlaufs bei Tag

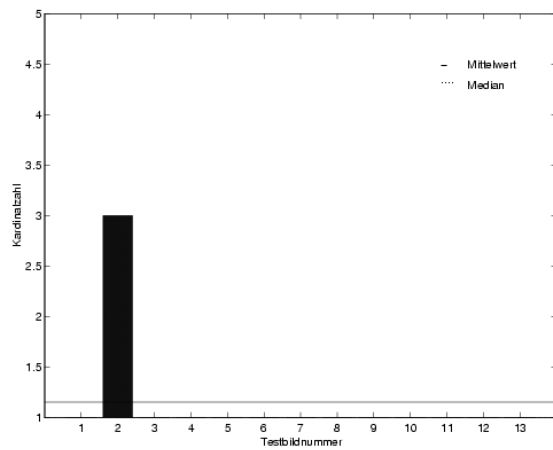


(a) Kardinalzahl (Mittelwert: 1.31, Median: 1)

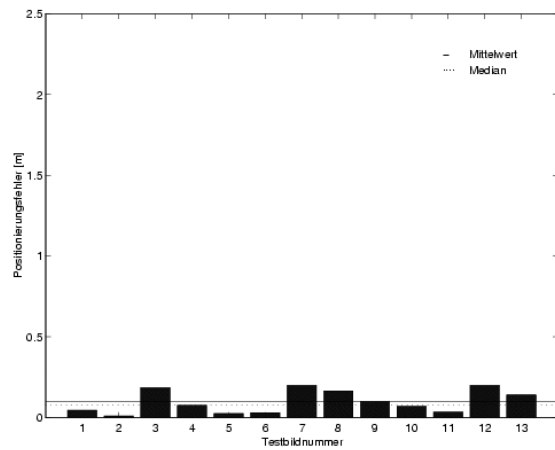


(b) Positionierungsfehler (Mittelwert: 0.08, Median: 0.06)

Abbildung 7.22: Testlauf bei Tag – Umgebung 1



(a) Kardinalzahl (Mittelwert: 1.15, Median: 1)



(b) Positionierungsfehler (Mittelwert: 0.10, Median: 0.08)

Abbildung 7.23: Testlauf bei Tag – Umgebung 2

Testlauf bei Nacht

Als nächstes folgt der interessantere Teil zum Testen des Tag-Nacht-Eigenraums: Die Klassifikation des Nachttestlaufs. Schon Abbildung 7.24 zeigt eine sehr deutliche Verbesserung der Klassifikation gegenüber der Verwendung des Tag-Eigenraums (vgl. S. 72). Der zugrundeliegende Pfad ist wie bei dem Tagtestlauf deutlich aus den geschätzten Positionen zu erkennen.

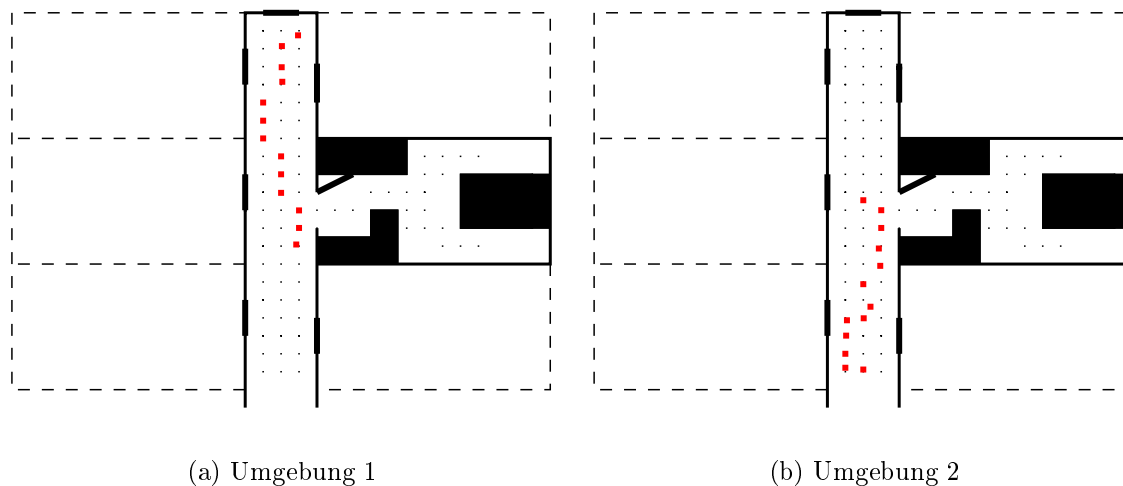
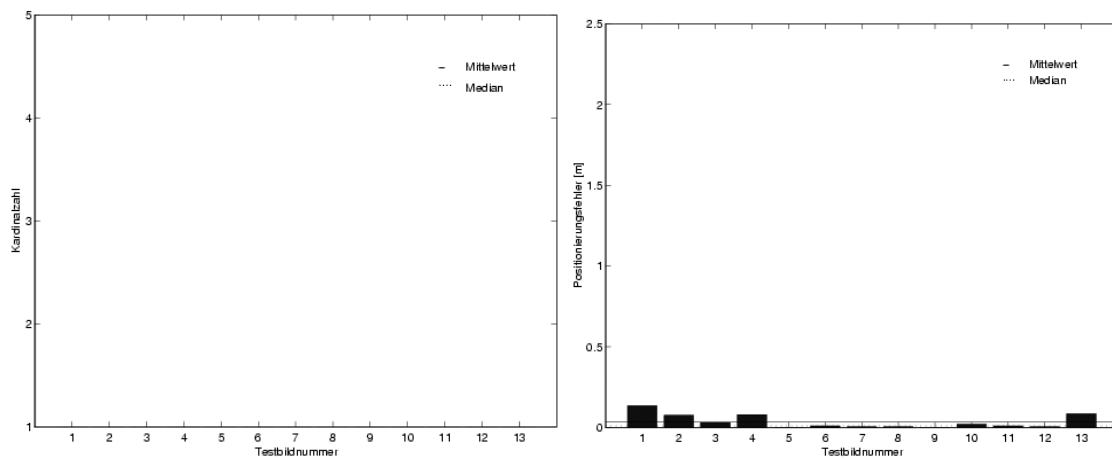


Abbildung 7.24: Die geschätzten Positionen des Testlaufs bei Nacht

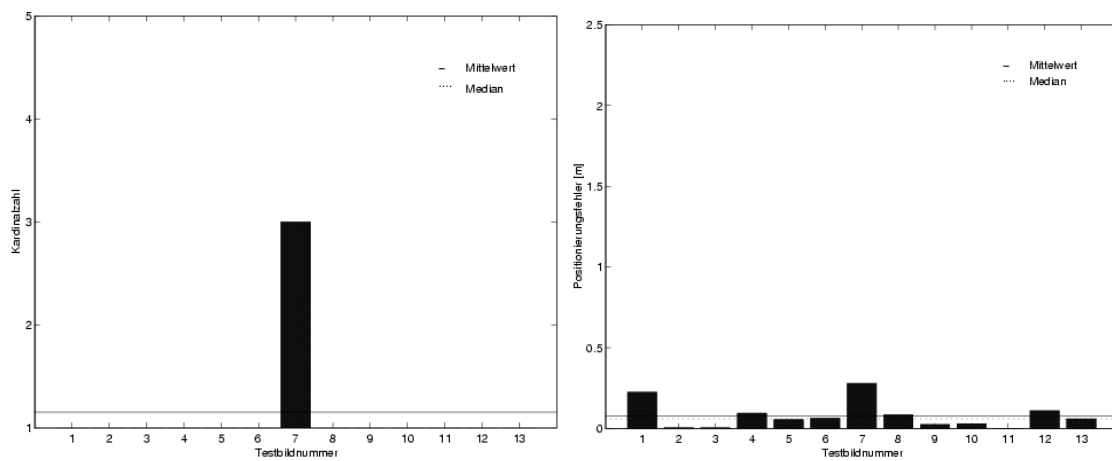
Auch die Kardinalzahlen und Positionierungsfehler (Abbildungen 7.25 und 7.26) sind eindeutig: Die Klassifikation liefert sehr gute Ergebnisse. Besonders beeindruckend ist Abbildung 7.25 (a): Es ist kein Balken zu erkennen, alle Testbilder haben Kardinalzahl 1, wurden also perfekt zugeordnet.



(a) Kardinalzahl (Mittelwert: 1.00, Median: 1), perfekte Klassifikation!

(b) Positionierungsfehler (Mittelwert: 0.04, Median: 0.01)

Abbildung 7.25: Testlauf bei Nacht – Umgebung 1



(a) Kardinalzahl (Mittelwert: 1.15, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.08, Median: 0.06)

Abbildung 7.26: Testlauf bei Nacht – Umgebung 2

7.2.4 Vergleich von Tag- und Tag-Nacht-Eigenräumen

In diesem Kapitel soll verglichen werden, inwiefern sich die Klassifikation verbessert, wenn statt eines reinen Tag-Eigenraums ein Tag-Nacht-Eigenraum verwendet wird. Tabelle 7.1 zeigt die Mittelwerte und Mediane von Kardinalzahlen und Positionierungsfehlern der Testläufe für Umgebung 1.

Testumgebung	Eigenraum	Kardinalzahl Mittelwert	Kardinalzahl Median	Pos.fehler Mittelwert [m]	Pos.fehler Median [m]
Tagtestlauf	Tag	1.23	1	0.07	0.08
Tagtestlauf	Tag-Nacht	1.31	1	0.08	0.06
Nachttestlauf	Tag	7.77	2	1.57	0.20
Nachttestlauf	Tag-Nacht	1.00	1	0.04	0.01

Tabelle 7.1: Testläufe klassifiziert mit Tag- und Tag-Nacht-Eigenräumen – Umgebung 1, Pro Testlauf 13 Bilder.

Während bei der Klassifikation des Tagtestlaufs mit verschiedenen Eigenräumen kaum ein Unterschied zu erkennen ist, ist der Unterschied bei dem Nachttestlauf gravierend: Einer mittleren Kardinalzahl von 7.77 im Tag-Eigenraum steht ein Wert von 1.00 (perfekte Klassifikation!) im Tag-Nacht-Eigenraum gegenüber. Ebenso deutlich sprechen die Werte der Positionierungsfehler: Der Median verbessert sich von 0.20 m auf 0.01 m um das 20-fache, der Mittelwert von 1.57 m auf 0.04 m gar um mehr als das 39-fache. Die Abbildungen 7.27 und 7.28 zeigen dies sehr anschaulich. Abbildung 7.27 zeigt die Klassifikation des Testlaufs bei Tag. Die blauen Balken für die Klassifikation mit dem Tag-Eigenraum und die roten Balken für die Klassifikation mit dem Tag-Nacht-Eigenraum haben immer eine ähnliche Höhe. Die y-Achse der Kardinalzahlengrafik beginnt hier ausnahmsweise bei 0.8, um die Kardinalzahlen mit Wert 1 besser vergleichen zu können.

Ganz anders sieht es in Abbildung 7.28 aus, die den Testlauf bei Nacht darstellt. Die blauen Balken der Klassifikation mit dem Tag-Eigenraum sind sehr hoch, die roten Balken der Klassifikation mit dem Tag-Nacht-Eigenraum sind dagegen sehr niedrig. Man kann sehr anschaulich die deutliche Verbesserung der Klassifikation erkennen, die durch Verwendung eines Tag-Nacht-Eigenraums erreicht wird.

Die Abbildungen 7.29 und 7.30 zeigen die gleichen Grafiken für Umgebung 2. Hier sind die Unterschiede nicht so gravierend, da Helligkeitsunterschiede zwischen Tag- und Nachtbildern in der Flurmitte nicht so sehr ins Gewicht fallen wie in Fensternähe (vgl. Kapitel 7.2.1). Trotzdem sind die Verbesserungen auch hier deutlich: In Abbildung 7.29 des Testlaufs bei Tag zeigen sich wiederum kaum Unterschiede in der Balkenhöhe, in Abbildung 7.30 des Testlaufs bei Nacht dagegen sind die roten Balken der Klassifikation mit dem Tag-Nacht-Eigenraum deutlich niedriger als die blauen Balken.

Tabelle 7.2 zeigt die Mittelwerte und Mediane von Kardinalzahlen und Positionierungsfehlern der Testläufe für Umgebung 2. Auch hier erkennt man die deutliche Verbesserung:

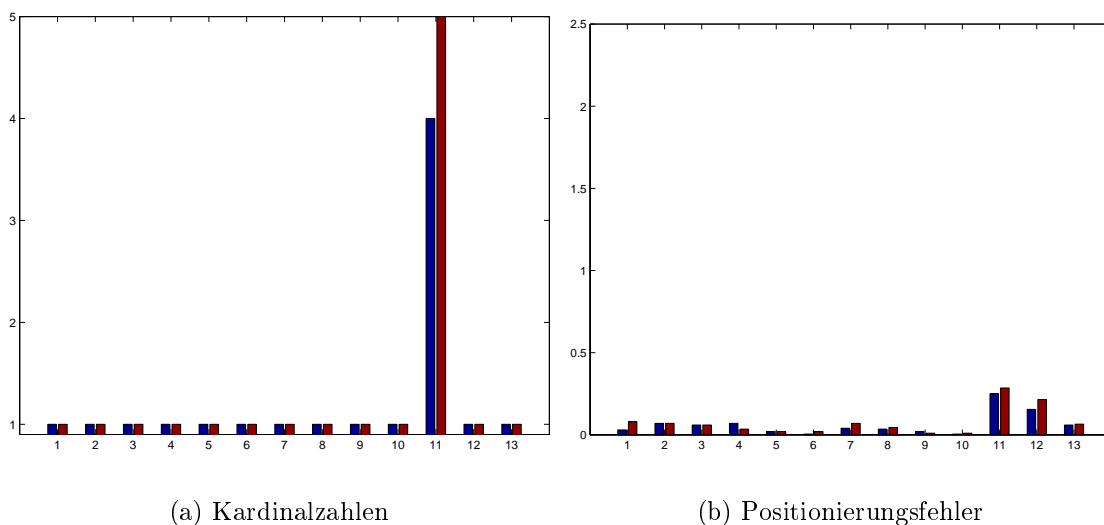


Abbildung 7.27: Klassifikation des Testlaufs bei Tag mit Tag-Eigenraum (blaue Balken) und mit Tag-Nacht-Eigenraum (rote Balken) – Umgebung 1.

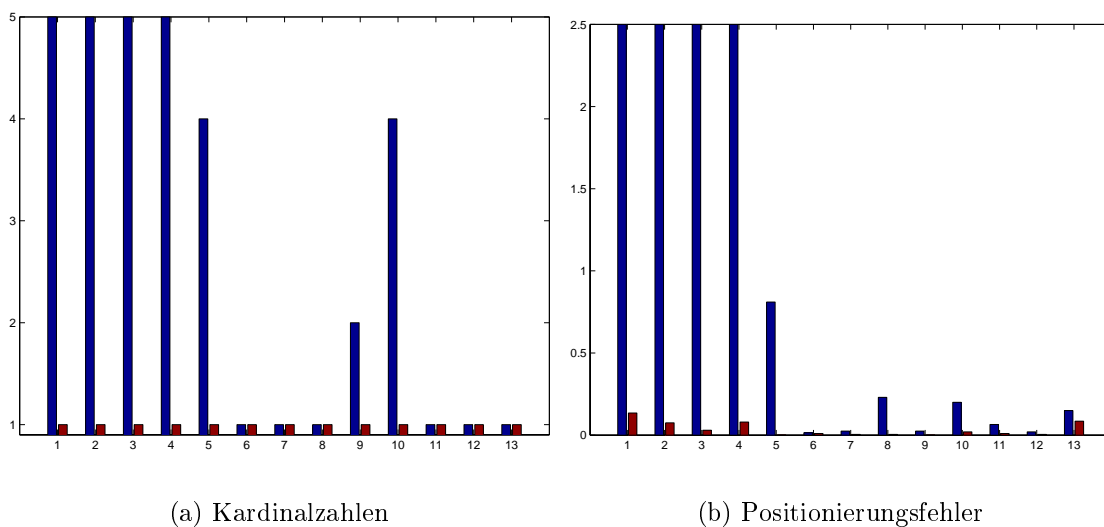


Abbildung 7.28: Klassifikation des Testlaufs bei Nacht mit Tag-Eigenraum (blaue Balken) und mit Tag-Nacht-Eigenraum (rote Balken) – Umgebung 1.

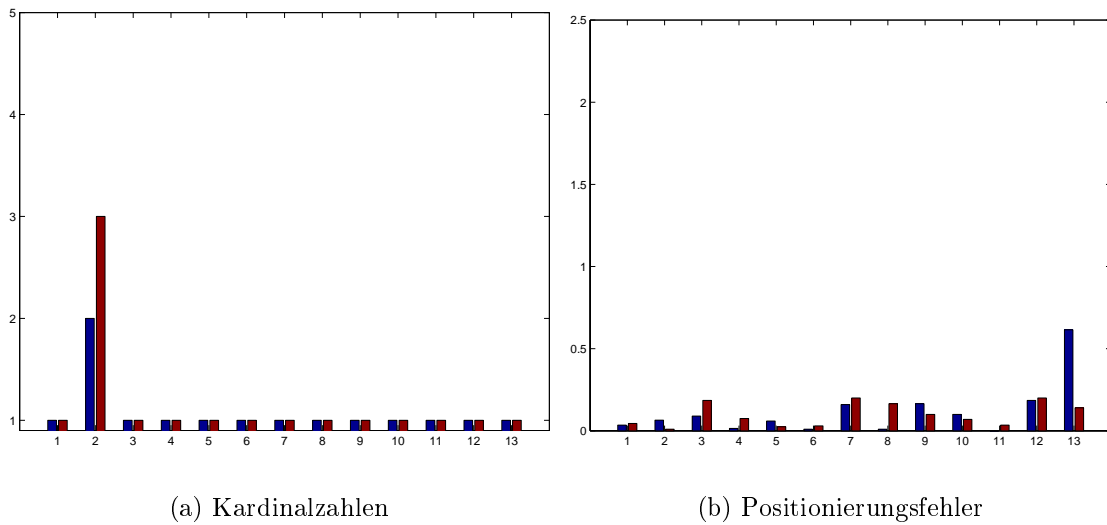


Abbildung 7.29: Klassifikation des Testlaufs bei Tag mit Tag-Eigenraum (blaue Balken) und mit Tag-Nacht-Eigenraum (rote Balken) – Umgebung 2.

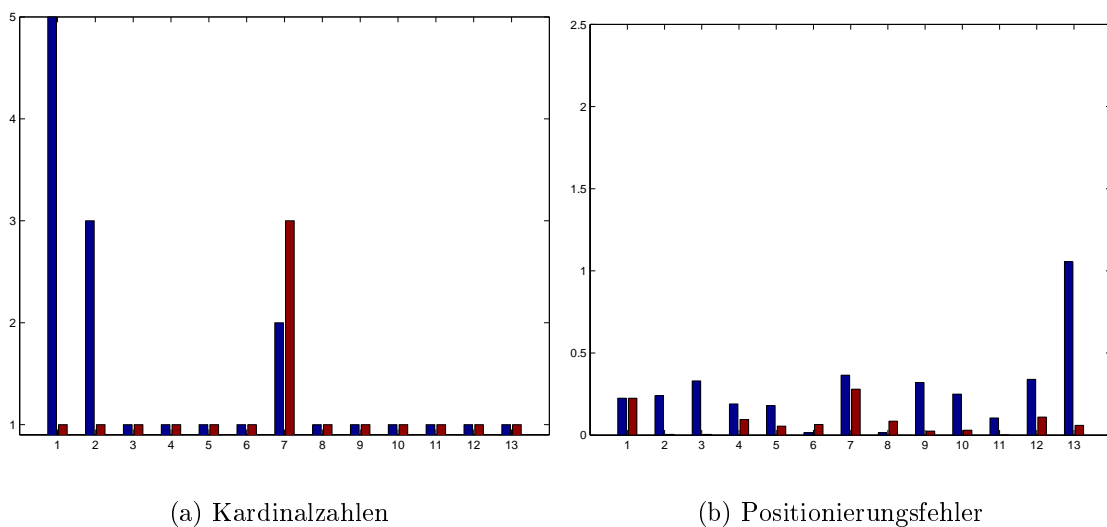


Abbildung 7.30: Klassifikation des Testlaufs bei Nacht mit Tag-Eigenraum (blaue Balken) und mit Tag-Nacht-Eigenraum (rote Balken) – Umgebung 2.

Der mittlere Positionierungsfehler des Nachttestlaufs verbessert sich von 0.28 m auf 0.08 m um mehr als das dreifache, der Median von 0.24 m auf 0.06 m um das vierfache.

Testumgebung	Eigenraum	Kardinalzahl Mittelwert	Kardinalzahl Median	Pos.fehler Mittelwert [m]	Pos.fehler Median [m]
Tagtestlauf	Tag	1.08	1	0.12	0.07
Tagtestlauf	Tag-Nacht	1.15	1	0.10	0.08
Nachtttestlauf	Tag	1.54	1	0.28	0.24
Nachtttestlauf	Tag-Nacht	1.15	1	0.08	0.06

Tabelle 7.2: Testläufe klassifiziert mit Tag- und Tag-Nacht-Eigenräumen – Umgebung 2, Pro Testlauf 13 Bilder.

Abschließend kann gesagt werden, daß die Verwendung von Tag-Nacht-Eigenräumen zu deutlich besseren Klassifikationsergebnissen führt, sofern auch Nachtbilder klassifiziert werden sollen. Die Erstellung eines solchen Eigenraums ist zwar aufwendiger (vgl. Kapitel 4.4), dieser zusätzliche Zeitaufwand bezieht sich aber nur auf die Vorbereitungsphase und nicht auf die Aktionsphase. Die zusätzliche Zeit in der Aktionsphase beschränkt sich auf die doppelte Anzahl von Trainingsbildern, mit denen das Testbild bei der Klassifikation verglichen werden muß. Dies fällt bei der Berechnung kaum ins Gewicht. Wenn der Tag-Nacht-Eigenraum also einmal erstellt ist, bietet er deutliche Vorteile gegenüber einem reinen Tag-Eigenraum.

7.3 Okklusionen

In natürlichen Umgebungen ist Lokalisierung oft schwierig, weil die Sensordaten gestört sind. Störungen treten z.B. durch Beleuchtungsvarianzen auf, wie im letzten Abschnitt behandelt, oder aber durch Veränderungen der Umgebung. Diese können durch Gegenstände oder durch umherlaufende Personen hervorgerufen werden. In Kamerabildern finden sich solche Hindernisse als Okklusionen von Teilbereichen der Bilder wieder. Da solche Probleme in der Realität an der Tagesordnung sind, soll in diesem Kapitel untersucht werden, wie die Vollbild-Lokalisierung mit Okklusionen umgeht.

In Kapitel 7.3.1 wird untersucht, wie die Methode auf geöffnete Türen reagiert, da diese deutliche Unterschiede in den Bildern hervorrufen können. In Kapitel 7.3.2 wird dann der Umgang mit Okklusionen durch Personen behandelt.

7.3.1 Geöffnete Türen

Wenn die Umgebung mit geschlossenen Türen gelernt wurde, können geöffnete Türen gravierende Unterschiede in den Bilddaten hervorrufen. Der Grad der Veränderung hängt natürlich davon ab, wie nah sich die geöffnete Tür an dem Roboter befindet. Türen am

anderen Ende des Ganges rufen kaum Veränderungen hervor, eine Tür direkt vor dem Roboter verändert einen beträchtlichen Teil des Bildes. Da nicht jede mögliche Kombination von geöffneten und geschlossenen Türen gelernt werden kann, haben wir untersucht, wie ein nur aus Bildern mit geschlossenen Türen gelernter Eigenraum mit geöffneten Türen umgeht. Für die Tests wurden alle Türen geöffnet, was den „worst case“ darstellt.

Abbildung 7.31 zeigt zwei Bilder mit geöffneten Türen, die trotzdem perfekt zugeordnet werden auf der linken Seite, rechts daneben die zugehörigen Trainingsbilder mit geschlossenen Türen. Man sieht, daß hier die Türen nur kleine Teile der Bilder einnehmen. In



Abbildung 7.31: Zwei Beispielbilder mit geöffneten Türen, die trotzdem perfekt zugeordnet werden links, die zugehörigen Trainingsbilder rechts

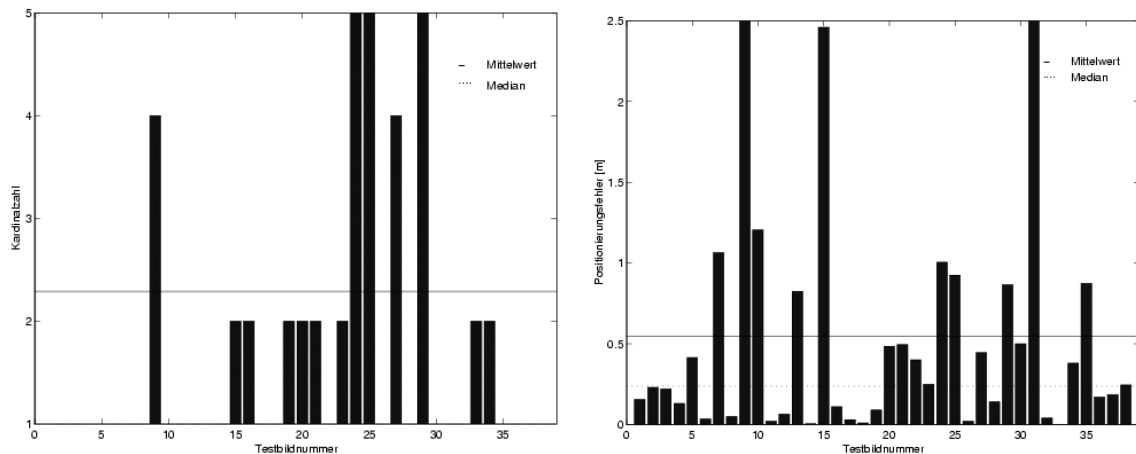


Abbildung 7.32: Zwei Beispielbilder mit geöffneten Türen, die falsch zugeordnet werden links, die zugehörigen Trainingsbilder rechts

Abbildung 7.32 sind dagegen zwei Bilder dargestellt, auf denen die Türen die Bilder in so großem Maße beeinträchtigen, daß eine akzeptable Klassifizierung nicht mehr möglich ist. Die Bilder werden völlig falsch zugeordnet.

Die Abbildungen 7.33 und 7.34 zeigen Kardinalzahlen und Positionierungsfehler der Tests mit geöffneten Türen für die Umgebungen 1 und 2. Die Ergebnisse der beiden Umgebungen unterscheiden sich kaum. In beiden erkennt man, daß einige Bilder sehr gut, andere sehr schlecht klassifiziert werden. Über die Hälfte der Bilder werden weiterhin perfekt klassifiziert, wie die Mediane der Kardinalzahlen mit Wert 1 zeigen. Dennoch zeigen

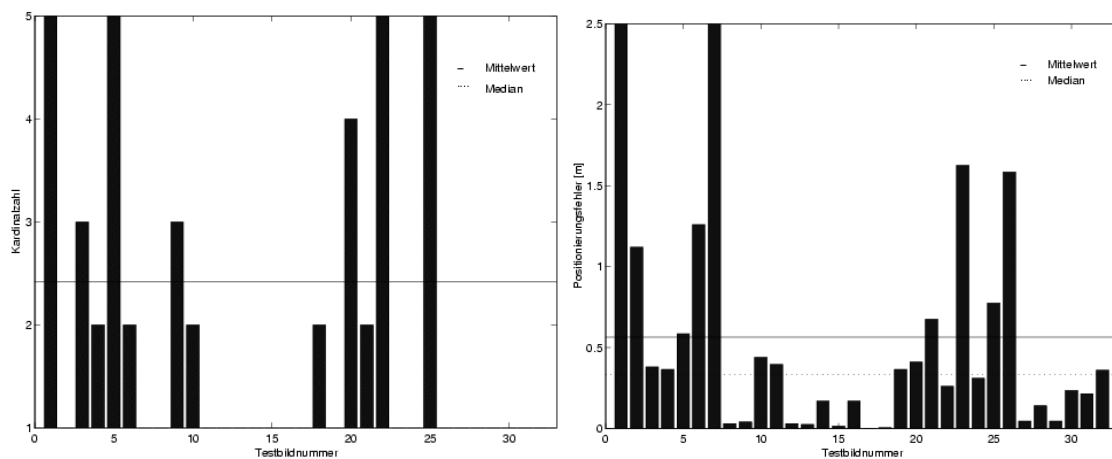
die mittleren Positionierungsfehler von über 50 cm, daß die Vollbild-Methode die Bilder zwar meist grob der richtigen Region zuordnet, eine genaue Zuordnung dagegen allerdings häufig nicht funktioniert.



(a) Kardinalzahl (Mittelwert: 2.29, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.55, Median: 0.24)

Abbildung 7.33: Geöffnete Türen – Umgebung 1



(a) Kardinalzahl (Mittelwert: 2.42, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.57, Median: 0.34)

Abbildung 7.34: Geöffnete Türen – Umgebung 2

7.3.2 Personen

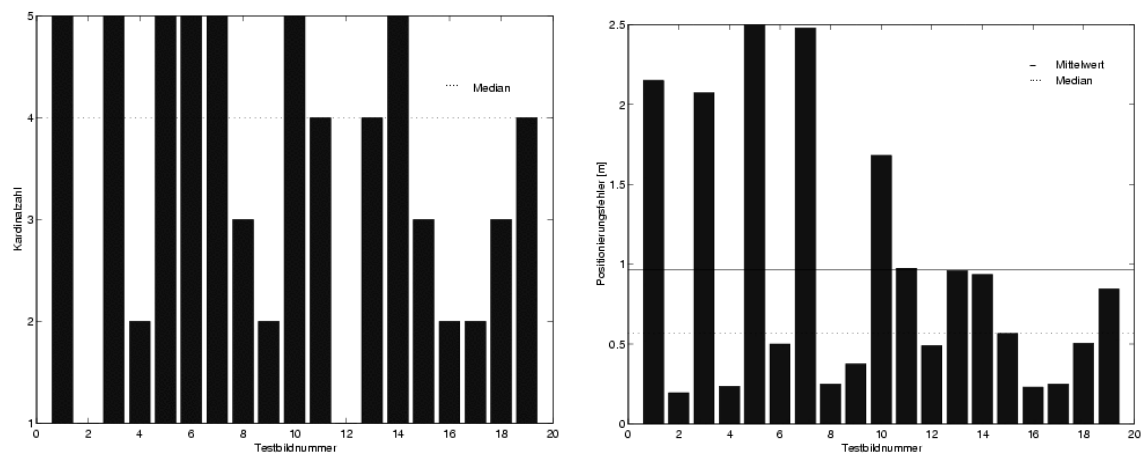
Noch stärkere Okklusionen als durch geöffnete Türen treten auf, wenn Personen die Umgebung bevölkern. In diesem Abschnitt soll gezeigt werden, wie die Vollbild-Lokalisierung arbeitet, wenn die Testbilder solche Okklusionen enthalten. Dazu wurden 19 Testbilder mit bis zu vier Personen und diversen Gegenständen in unmittelbarer Nähe des Roboters aufgenommen. Abbildung 7.35 zeigt links einige Beispielbilder und rechts die zugehörigen Trainingsbilder ohne Okklusionen. Für eine Büroumgebung sind dies sicherlich Extremsituationen; in anderen Umgebungen, wie z.B. Museen muß mit noch mehr Personen gerechnet werden.



Abbildung 7.35: Bilder mit Personen links, die zugehörigen Trainingsbilder rechts

Das oberste Bild wurde perfekt zugeordnet, das mittlere liegt mit einer Kardinalzahl von 3 im mittleren Bereich und das untere wurde völlig falsch zugeordnet. Abbildung 7.36 zeigt Kardinalzahlen und Positionierungsfehler zu den Tests mit Personen. Man sieht, daß die Werte ziemlich schlecht ausfallen. Sie sind zwar weit von einer zufälligen Zuordnung entfernt, was bei der Stärke der Okklusionen überraschen mag, trotzdem ist ein mittlerer Positionierungsfehler von fast einem Meter kein sehr guter Wert für eine Lokalisierungsmethode.

In diesem Abschnitt haben wir also gesehen, daß die Vollbild-Lokalisierung große Probleme bereitet, wenn die Bilder starke Okklusionen beinhalten. Im folgenden Kapitel werden wir sehen, wie die Erkennung von Bildern mit Okklusionen durch die in Kapitel 5 vorgestellte Sektoren-Methode verbessert werden kann.



(a) Kardinalzahl (Mittelwert: 5.63, Median: 4)

(b) Positionierungsfehler (Mittelwert: 0.97, Median: 0.57)

Abbildung 7.36: Klassifikation von Bildern mit Personen – Umgebung 1

Kapitel 8

Ergebnisse der Sektoren-Lokalisierung

Dieses Kapitel zeigt die Ergebnisse der in Kapitel 5 beschriebenen Sektoren-Lokalisierung. Die Panoramabilder wurden in 36 Sektorbilder mit einer Breite von 20 Pixeln unterteilt. Dies entspricht 20° des gesamten Blickfeldes. Die Sektoren beginnen in 10° -Abständen, so daß sich zwei benachbarte Sektoren je um die Hälfte überlappen. Aus den Sektoren der Trainingsbilder wurde dann ein Eigenraum generiert, in den die Sektorbilder projiziert wurden. Die Klassifikation geschah wie in Kapitel 5 beschrieben durch Bayes'sches Update. Die Experimente beschränken sich hier auf die Umgebung 1, da der Schwerpunkt nicht auf dem Unterschied zwischen den Umgebungen liegt, sondern auf dem Vergleich der verschiedenen Lokalisierungsmethoden.

In Abschnitt 8.1 wird zunächst die Funktionsweise der Sektoren-Lokalisierung unter günstigen Bedingungen gezeigt sowie der Unterschied zwischen der Verwendung von zwei und von drei Sektoren. In Abschnitt 8.2 wird die Lokalisierung mit rotierten Bildern präsentiert und Abschnitt 8.3 zeigt, wie das Verfahren auf Bilder mit starken Okklusionen reagiert.

8.1 Anzahl von Sektoren

In diesem Abschnitt werden die Ergebnisse der Sektoren-Lokalisierung unter Idealbedingungen gezeigt. Als Testdaten werden die Bilder des Testlaufs bei Tag verwendet, die auch einigen Experimenten des letzten Kapitels zugrunde lagen (siehe Abbildung 7.1). Um zu untersuchen, wie die Methode von der Anzahl verwendeter Sektoren abhängt, wurden die Experimente mit zwei und mit drei Sektoren durchgeführt.

Zwei Sektoren

Zunächst wird die einfachste Variante der Sektoren-Lokalisierung mit Kontext gewählt: Die Lokalisierung mit zwei Sektoren. Dabei wird das Sektorpaar ermittelt, das die beste Übereinstimmung mit den Trainingsdaten bietet und zur Klassifikation herangezogen. Abbildung 8.1 zeigt die Kardinalzahlen und Positionierungsfehler dieser Experimente. Die

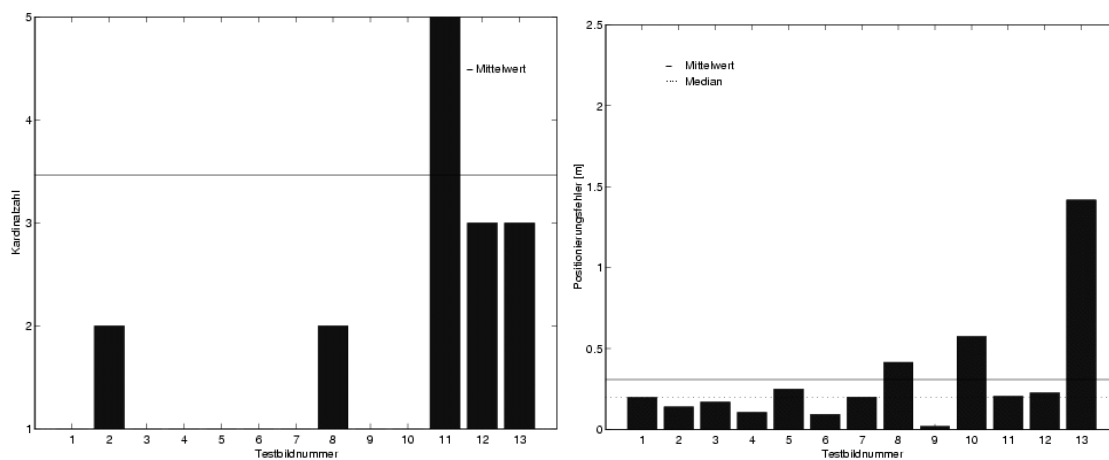
Werte zeigen, daß das Verfahren korrekt arbeitet; die Klassifikation ist in den meisten Fällen erfolgreich, was der Median der Kardinalzahlen von 1 belegt. Die Werte sind allerdings schlechter als die der Vollbild-Lokalisierung (siehe auch Kapitel 9: Vergleich der Lokalisierungsmethoden). Das liegt daran, daß die Sektoren sehr schmal sind und wenig Information beinhalten. Normale Schwankungen zwischen Test- und Trainingsdaten können so zu Mehrdeutigkeiten führen. Eine Verbreiterung der Sektoren oder das Einbeziehen von mehr als zwei Sektoren sollte zu einer Verbesserung führen. Letztere Möglichkeit wird im folgenden Abschnitt präsentiert.

Drei Sektoren

Da die Lokalisierung mit Hilfe von zwei Sektoren häufig zu Mehrdeutigkeiten führt, werden in diesem Abschnitt drei Sektoren zur Lokalisierung herangezogen. Abbildung 8.2 zeigt die Kardinalzahlen und Positionierungsfehler dieser Experimente. Man sieht, daß nur zwei Bilder nicht perfekt klassifiziert werden, im Gegensatz zu 5 Bildern der 2-Sektoren-Lokalisierung. Bei beiden Methoden wird wieder Bild Nr. 11 schlecht klassifiziert, das auch schon bei der Vollbild-Lokalisierung Probleme bereitet hatte (vgl. Abb. 7.5, S. 64). Der Positionierungsfehler verringert sich durchschnittlich auf die Hälfte (siehe Tabelle 8.1). Es zeigt sich also, daß die Klassifizierung von nicht-okkludierten Testbildern besser ausfällt, wenn mehr Sektoren betrachtet werden. Es läßt sich vermuten, daß mit noch mehr Sektoren noch bessere Ergebnisse erzielt werden könnten. Allerdings wird bei einer größeren Sektorenzahl der Umgang mit starken Okklusionen erschwert (vgl. Kapitel 8.3). In Kapitel 9.4 wird genauer diskutiert, durch welche Optimierungen das Verfahren verbessert werden könnte.

Lokalisierungsmethode	Kardinalzahl Mittelwert	Kardinalzahl Median	Pos.fehler Mittelwert [m]	Pos.fehler Median [m]
2-Sektoren-Lokalisierung	3.46	1	0.31	0.20
3-Sektoren-Lokalisierung	3.31	1	0.15	0.12

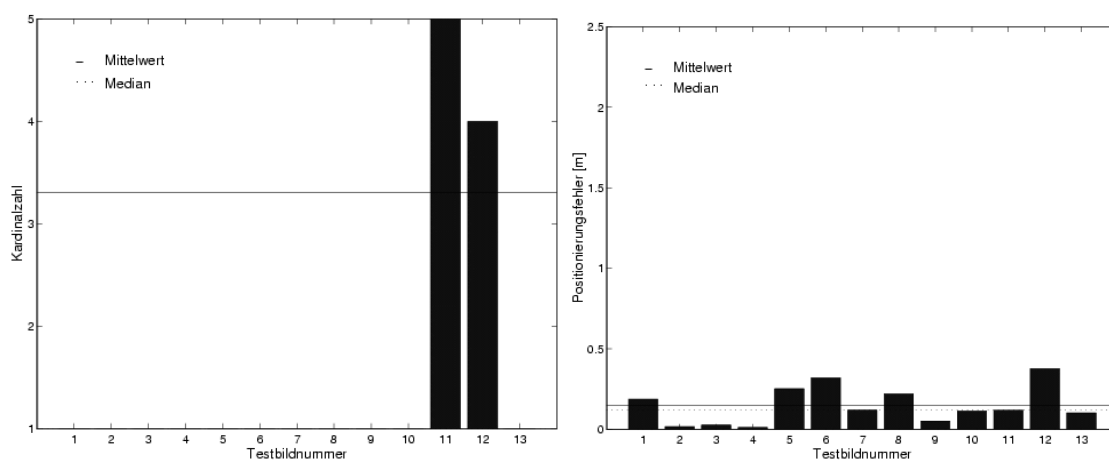
Tabelle 8.1: Klassifikation des Testlaufs mit verschiedener Sektorenanzahl, pro Testlauf 13 Bilder.



(a) Kardinalzahlen (Mittelwert: 3.46, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.31, Median: 0.20)

Abbildung 8.1: Testlauf mit 2-Sektoren-Lokalisierung



(a) Kardinalzahlen (Mittelwert:3.31 , Median: 1)

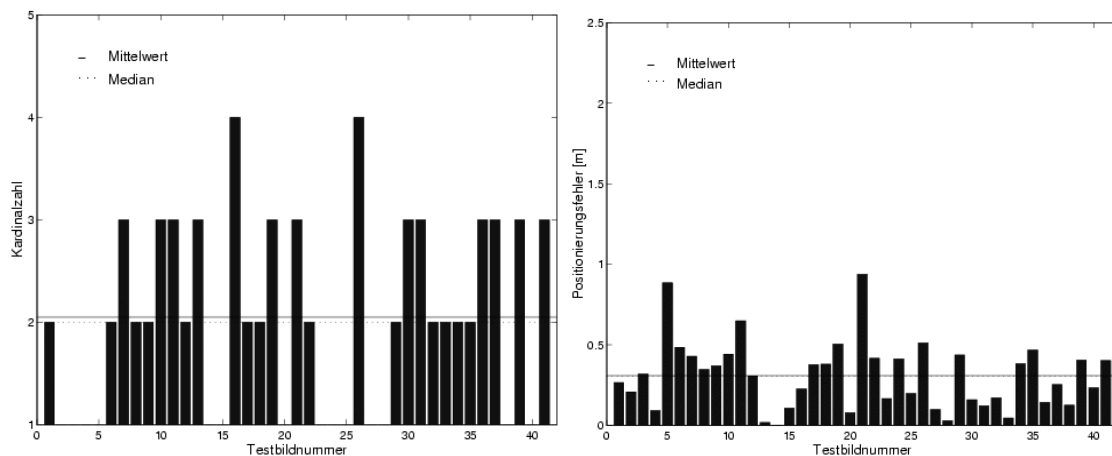
(b) Positionierungsfehler (Mittelwert: 0.15, Median: 0.12)

Abbildung 8.2: Testlauf mit 3-Sektoren-Lokalisierung

8.2 Rotationen

In Kapitel 5.4 wurde beschrieben, daß die Sektoren-Lokalisierung rotationsunabhängig ist, wenn die Rotation ein Vielfaches des Abstandes zwischen benachbarten Sektoren beträgt. In unseren Experimenten betrug dieser Abstand 10 Pixel, was 10° des gesamten Blickfeldes entspricht. Um dies zu belegen, wurden die Trainingsbilder um 30° rotiert und mit der 3-Sektoren-Lokalisierung klassifiziert. Es zeigte sich, daß die Klassifizierung perfekt ausfiel; alle Bilder wurden korrekt zugeordnet.

Um zu überprüfen, wie die Methode mit beliebig rotierten Bildern umgeht, wurden um 5° rotierte Bilder klassifiziert. Bei Sektorenabständen von 10° stellt dies den „worst case“ dar. Die Ergebnisse dieser Experimente sind in Abbildung 8.3 dargestellt. Die Klassifizierung fällt deutlich schlechter aus als die Klassifizierung derselben Bilder mit der Vollbild-Lokalisierung. Da die Sektoren sehr schmal sind (20°), macht eine Rotation um 5° ein Viertel eines Sektors aus, was die Methode anfälliger gegenüber kleinen Rotationen macht. Die Vorteile der Sektoren-Lokalisierung bezüglich Rotationen zeigt sich erst ab Rotationswinkeln von etwa 20° , bei denen die Vollbild-Lokalisierung versagt. Ein direkter Vergleich von Vollbild- und Sektoren-Lokalisierung bezüglich Rotationen findet sich in Kapitel 9.2.



(a) Kardinalzahlen (Mittelwert: 2.05 , Median: 2)

(b) Positionierungsfehler (Mittelwert: 0.31 , Median: 0.31)

Abbildung 8.3: Rotationen um 5°

8.3 Okklusionen

In diesem Kapitel soll überprüft werden, wie die Sektoren-Lokalisierung solche Bilder klassifiziert, von denen Teile durch Okklusionen verdeckt sind. Um die Güte der Klassifikation abhängig vom Grad der Verdeckung zu testen, wurden die Bilder systematisch zu einem festgelegten Prozentsatz mit Bildern einer Datenbank künstlich verdeckt. Die Ergebnisse werden in Abschnitt 8.3.1 dargestellt. In Abschnitt 8.3.2 wird die Methode dann auf Bilder angewendet, die natürliche Okklusionen in Form von Personen und Gegenständen enthalten.

8.3.1 Künstliche Okklusionen

Um die Güte der Klassifizierung abhängig vom Grad der Verdeckung zu testen, wurden zunächst Teile der zu klassifizierenden Bilder systematisch zu einem festgelegten Prozentsatz verdeckt. Um den Umgang mit Okklusionen testen zu können, unabhängig von der Fähigkeit Testbilder zu klassifizieren, wurden für diese Experimente die Trainingsbilder künstlich verdeckt. Zur Verdeckung wurden Bilder einer Datenbank von 25 Bildern ausgewählt, die in Abbildung 8.4 dargestellt ist. Sie besteht etwa zur Hälfte aus Bildern, die Testaufnahmen des Raums entnommen wurden. Sie zeigen Tische, Stühle, Regale und Personen und repräsentieren somit Verdeckungen, wie sie in einem realen Anwendungsfall vorkommen könnten. Die weiteren Bilder der Datenbank wurden so gewählt, daß sowohl helle als auch dunkle Bilder und sowohl kontrastreiche als auch kontrastarme Bilder vorhanden sind.

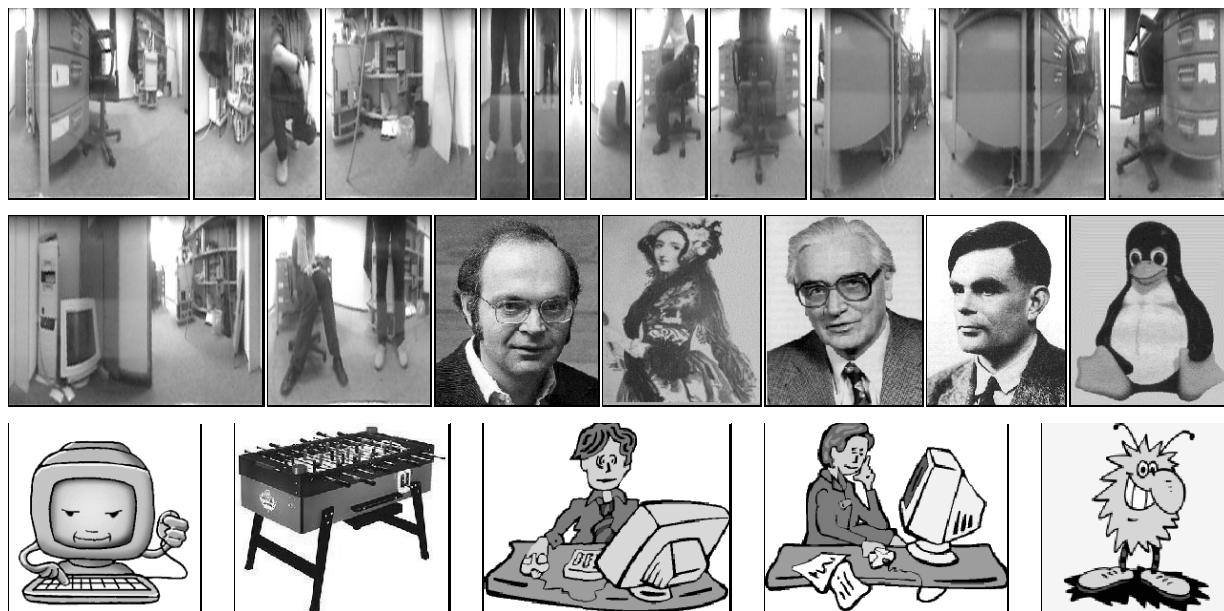


Abbildung 8.4: Die Bilder der Okklusions-Bilddatenbank

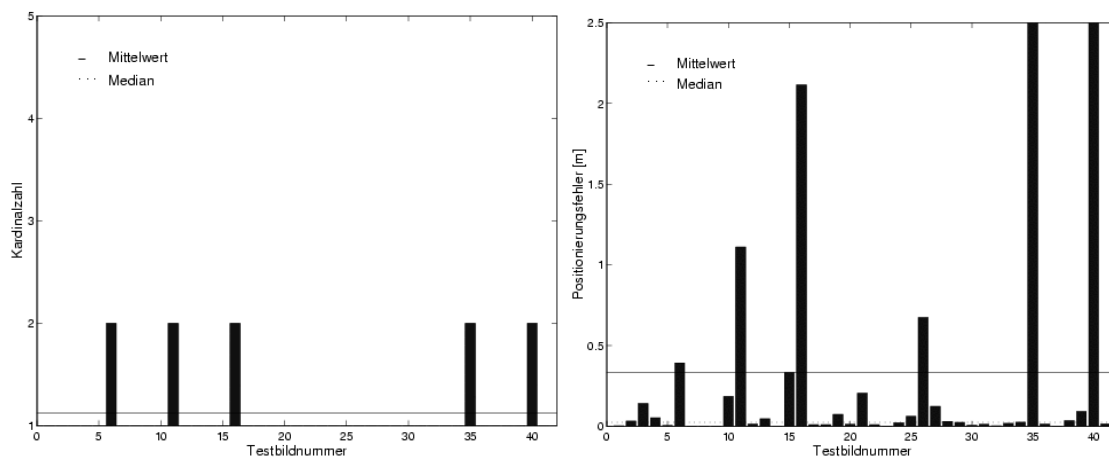
Aus dieser Datenbank wurden nun zufällig Bilder ausgewählt, die zur Verdeckung der Trainingsbilder genutzt wurden. Sie wurden je nach Verdeckungsgrad in drei bis fünf Balken angeordnet, die beliebig angeordnet zur Verdeckung der Bilder dienten. Die Balken wurden nicht nur an Sektorgrenzen, sondern an beliebigen Stellen in die Bilder eingebunden und haben eine Mindestbreite von 10 Pixeln. Ebenso haben die nicht verdeckten Bereiche zwischen zwei Okklusionsbalken die Mindestbreite von 10 Pixeln. Als Verdeckungsgrade wurden 30 %, 50 %, 60 %, 70 %, 80 % und 90 % Verdeckung gewählt. Abbildung 8.5 zeigt einige künstlich okkludierte Bilder mit verschiedenen Verdeckungsgraden.



Abbildung 8.5: Einige künstlich okkludierte Bilder mit verschiedenem Grad von Okklusionen. Von links oben nach rechts unten: Verdeckungsgrad von 30 %, 50 %, 60 %, 70 %, 80 % und 90 %.

Die Experimente wurden sowohl mit der 2- als auch mit der 3-Sektoren-Lokalisierung durchgeführt. Bevor wir genauer auf den Vergleich zwischen den Lokalisierungen mit verschiedener Sektorenanzahl eingehen, hier zunächst einige detaillierte Informationen zur Lokalisierung mit drei Sektoren, da diese größtenteils bessere Ergebnisse liefert.

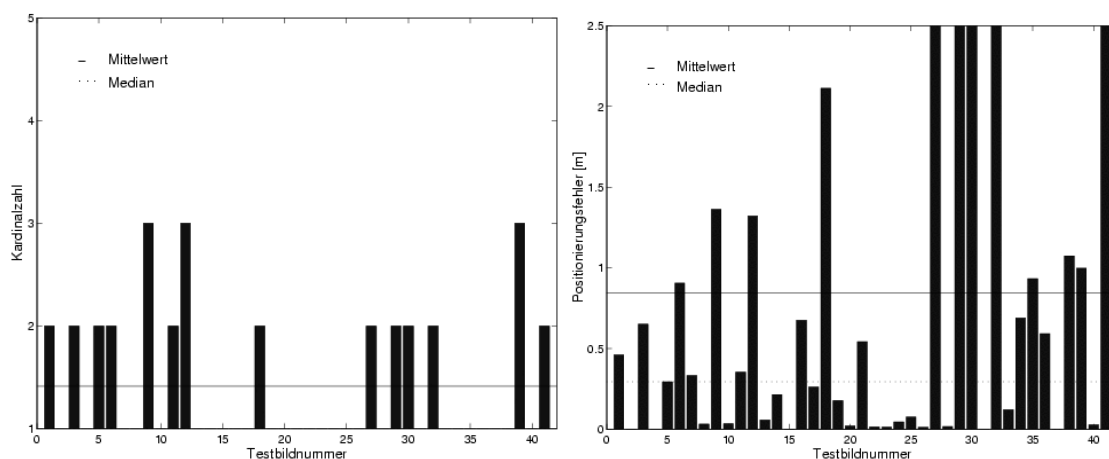
Die Abbildungen 8.6 bis 8.9 stellen die Kardinalzahlen und Positionierungsfehler der Experimente mit 30 %, 60 %, 70 %, und 80 % Verdeckung dar. Bis zu Okklusionen von 60 % werden über die Hälfte der Bilder perfekt klassifiziert. Einige Bilder werden schlechter klassifiziert; das ist vor allem dann der Fall, wenn keine drei unverdeckten Sektoren zur Verfügung stehen, die eine gute Klassifikation ermöglichen. Bei einer Verdeckung von 70 % fallen die Ergebnisse schon deutlich schlechter aus. Bei noch stärkeren Okklusionen ist das Klassifikationsergebnis fast zufällig, was eine mittlere Kardinalzahl von 15.34 bei 80 % iger Verdeckung verdeutlicht (Der Erwartungswert der Kardinalzahlen bei zufälliger Zuordnung beträgt bei 41 Bildern 20.5).



(a) Kardinalzahlen (Mittelwert: 1.12 , Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.33, Median: 0.02)

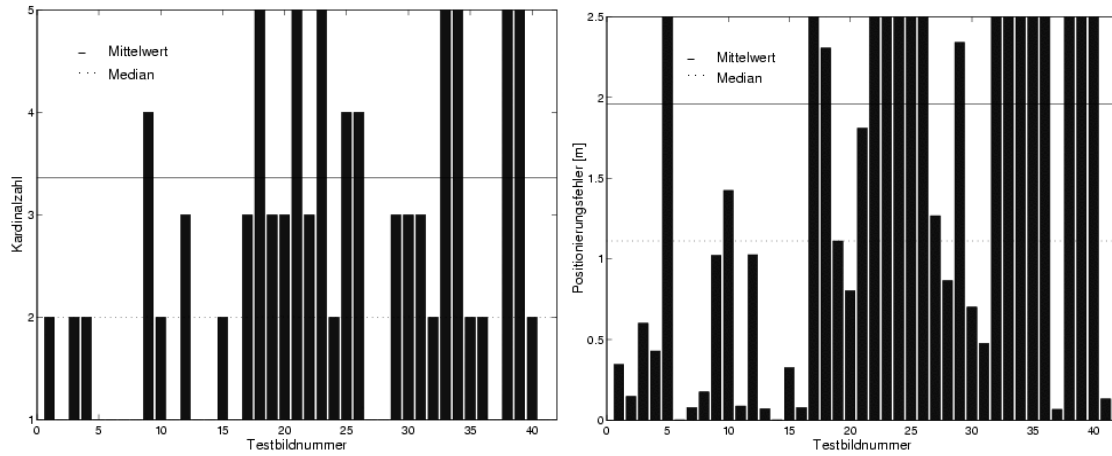
Abbildung 8.6: 30 % Verdeckung



(a) Kardinalzahlen (Mittelwert: 1.42, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.85, Median: 0.29)

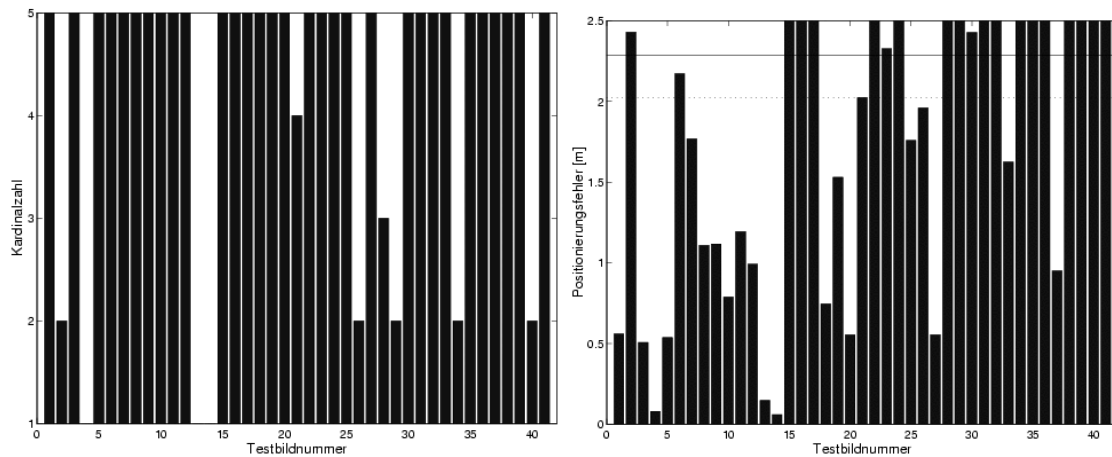
Abbildung 8.7: 60 % Verdeckung



(a) Kardinalzahlen (Mittelwert: 3.37, Median: 2)

(b) Positionierungsfehler (Mittelwert: 1.96, Median: 1.11)

Abbildung 8.8: 70 % Verdeckung



(a) Kardinalzahlen (Mittelwert: 15.34, Median: 14)

(b) Positionierungsfehler (Mittelwert: 2.28, Median: 2.02)

Abbildung 8.9: 80 % Verdeckung

Kommen wir jetzt zu dem Vergleich von 2- und 3-Sektoren-Lokalisierung bezüglich Okklusionen. Die Abbildungen 8.10 und 8.11 zeigen die Mittelwerte und Mediane der Experimente. Die durchgezogene Linie stellt die Werte der 2-Sektoren Lokalisierung dar, die gestrichelte Linie repräsentiert die 3-Sektoren-Lokalisierung. Während die Kardinalzahlen in (a) kaum Unterschiede aufweisen, zeigen sich deutlich erkennbare Differenzen bei den Positionierungsfehlern in (b). Es zeigt sich, daß die 3-Sektoren-Lokalisierung bis zu einer Verdeckung von 60 % besser arbeitet, bei höherem Verdeckungsgrad arbeitet dagegen die 2-Sektoren-Lokalisierung besser. Der Grund dafür ist, daß bei hohem Verdeckungsgrad oft

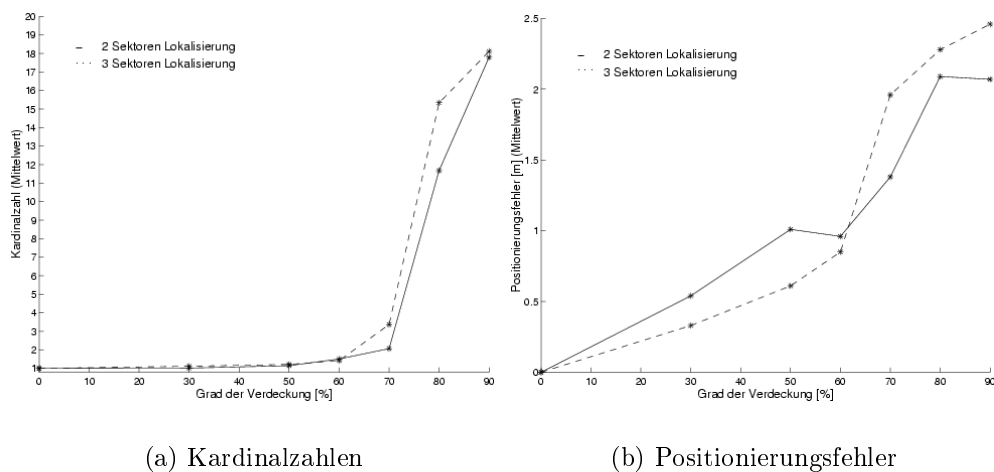


Abbildung 8.10: Mittelwerte der Lokalisierung mit 2 und 3 Sektoren, pro Verdeckungsgrad 41 Bilder.

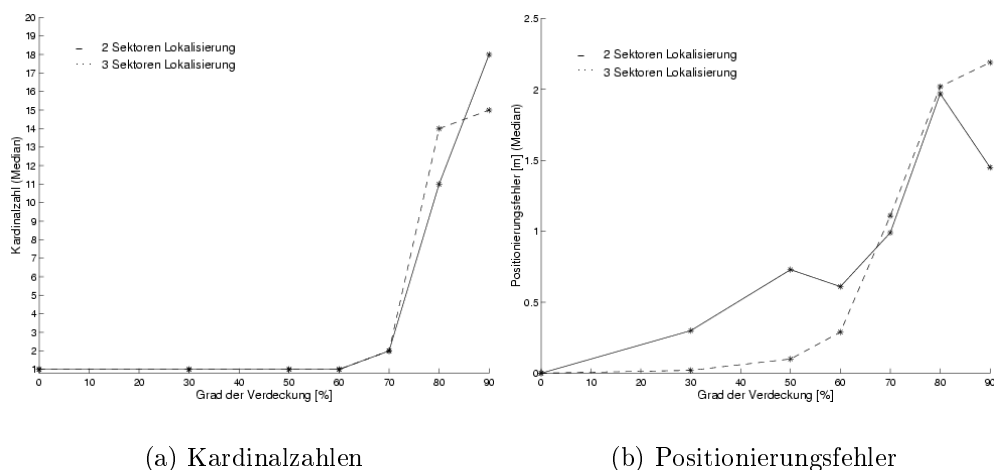


Abbildung 8.11: Mediane der Lokalisierung mit 2 und 3 Sektoren, pro Verdeckungsgrad 41 Bilder.

keine drei unverdeckten Sektoren vorhanden sind, die zur Klassifikation dienen könnten; zwei unverdeckte Sektoren finden sich dagegen häufiger.

8.3.2 Natürliche Okklusionen

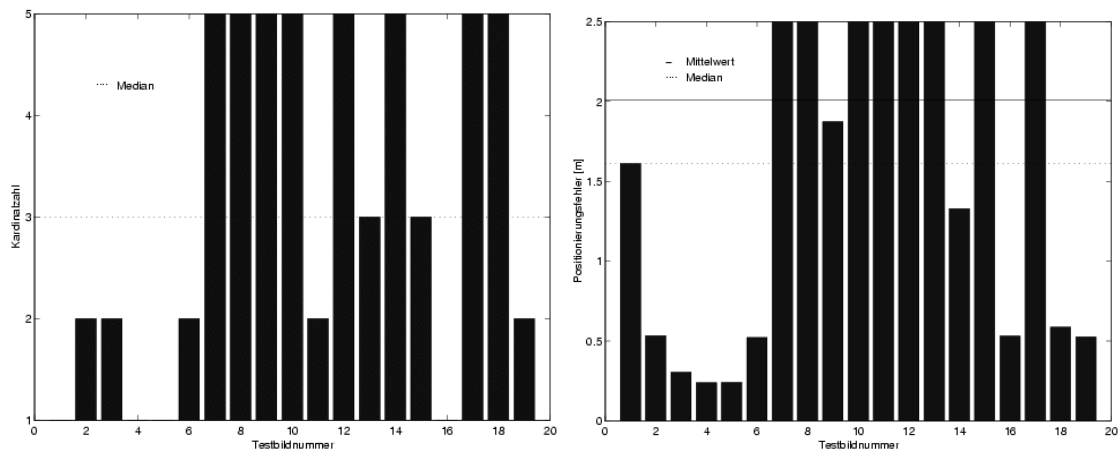
Im letzten Abschnitt wurde anhand von künstlich okkludierten Bildern getestet, wie gut die Sektoren-Lokalisierung bei verschiedenen Verdeckungsgraden arbeitet und wie die Güte der Klassifikation mit verschiedener Sektorenzahl vom Verdeckungsgrad abhängt. In diesem Abschnitt wird untersucht, ob diese Ergebnisse sich mit echten okkludierten Testbildern bestätigen, d.h. mit Bildern, die Personen und Gegenstände enthalten. Als Testdaten wurden die Bilder aus dem vorherigen Kapitel gewählt (S. 84). Bei diesen Bildern läßt sich der Verdeckungsgrad nicht genau bestimmen, es sind nur grobe Schätzungen durch Betrachten der Bilder möglich. Zudem ist der Verdeckungsgrad bei allen Bildern der Testmenge unterschiedlich, was eine generelle Aussage erschwert. Als grobe Schätzung läßt sich sagen, daß der Verdeckungsgrad der Bilder zwischen 50 und 80 % schwanken dürfte. Da sich im letzten Abschnitt gezeigt hat, daß die 3-Sektoren-Lokalisierung unter 70 % Verdeckung besser arbeitet, ab dieser Stelle aber die 2-Sektoren-Lokalisierung Vorteile hat, sollten sich auch Rückschlüsse von den Ergebnissen der Klassifikation auf den Verdeckungsgrad ziehen lassen.

Tabelle 8.2 zeigt, daß die 3-Sektoren-Lokalisierung durchschnittlich deutlich bessere Werte liefert als die Lokalisierung mit zwei Sektoren. Der mittlere Positionierungsfehler verringert sich auf fast die Hälfte.

Lokalisierungsmethode	Kardinalzahl Mittelwert	Kardinalzahl Median	Pos.fehler Mittelwert [m]	Pos.fehler Median [m]
2-Sektoren-Lokalisierung	5.32	3	2.01	1.61
3-Sektoren-Lokalisierung	4.68	2	1.15	0.64

Tabelle 8.2: Klassifikation der natürlich okkludierten Bilder mit 2- und 3-Sektoren-Lokalisierung, jeweils 19 Testbilder.

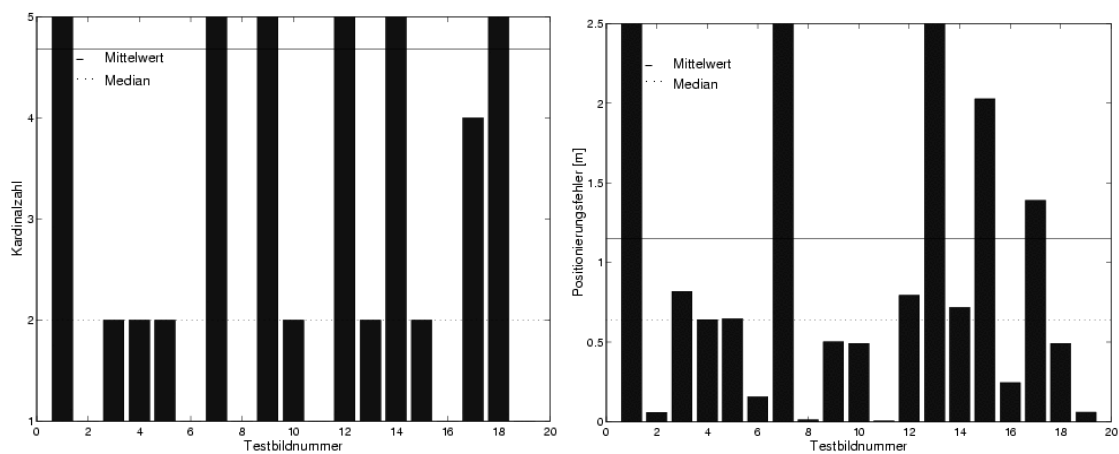
Dies zeigt sich auch sehr anschaulich in den Abbildungen 8.12 und 8.13. Man erkennt deutlich die Verbesserung, die durch die Verwendung eines zusätzlichen Sektors erreicht wird. Daraus läßt sich schließen, daß der Verdeckungsgrad der Bilder im Durchschnitt unter 70 % liegt. Daß dies jedoch nicht bei allen Bildern der Fall ist, zeigt Testbild Nr. 1: Die 2-Sektoren-Lokalisierung ergibt eine Kardinalzahl von 1, die 3-Sektoren-Lokalisierung einen Wert größer 5. Offenbar existieren zwei unverdeckte Sektoren, die eine gute Klassifikation ermöglichen, ein dritter Sektor, der eine gute Übereinstimmung mit den Trainingsdaten bot, wurde dagegen nicht gefunden. Der Verdeckungsgrad des Bildes beträgt also vermutlich mindestens 70 %.



(a) Kardinalzahlen (Mittelwert: 5.32, Median: 3)

(b) Positionierungsfehler (Mittelwert: 2.01, Median: 1.61)

Abbildung 8.12: 2-Sektoren-Lokalisierung



(a) Kardinalzahlen (Mittelwert: 4.68, Median: 2)

(b) Positionierungsfehler (Mittelwert: 1.15, Median: 0.64)

Abbildung 8.13: 3-Sektoren-Lokalisierung

Daß einige Bilder sowohl bei der Verwendung von zwei als auch von drei Sektoren keine guten Ergebnisse liefern, liegt wie bei der Klassifikation des Testlaufs an der geringen Informationsmenge der Sektoren, die zu Mehrdeutigkeiten führt. Insgesamt läßt sich sagen, daß bei realen Testbildern die 3-Sektoren-Lokalisierung vorzuziehen ist, da der Verdeckungsgrad selten höher als 70 % liegt. Am günstigsten wäre jedoch ein Verfahren, das je nach Verdeckungsgrad bestimmt, wieviele Sektoren zur Lokalisierung herangezogen werden sollen, indem all die Sektoren ausgewählt werden, die eine gute Übereinstimmung mit den Trainingsdaten liefern. Ein Vergleich dieser Ergebnisse mit den Ergebnissen der Vollbildmethode folgt im nächsten Kapitel.

Kapitel 9

Vergleich der Lokalisierungsmethoden

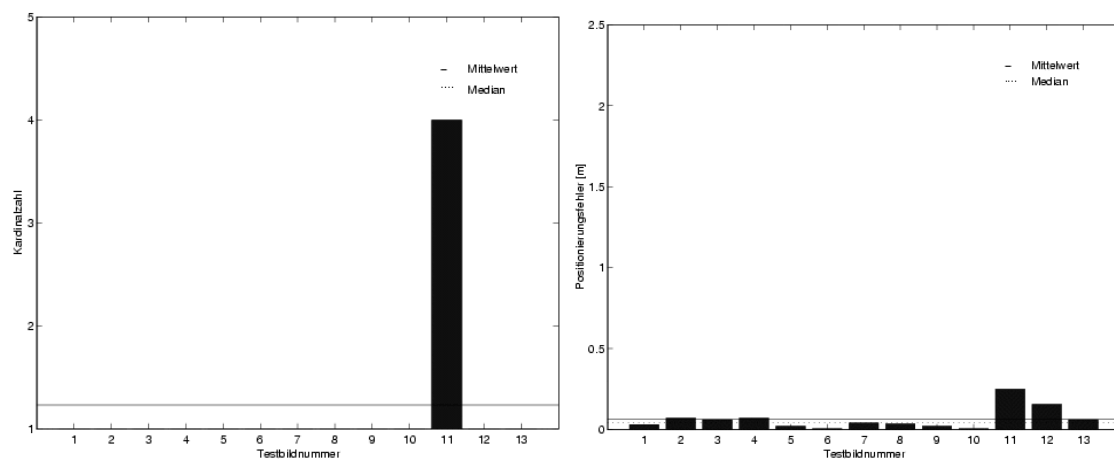
Die in den letzten zwei Kapiteln gezeigten Ergebnisse der Vollbild-Lokalisierung einerseits und der Sektoren-Lokalisierung andererseits sollen in diesem Kapitel nun direkt gegenübergestellt werden. Als Repräsentant der Sektoren-Lokalisierung wurde hier die Lokalisierung mit drei Sektoren gewählt, da sie durchschnittlich bessere Werte liefert als die Lokalisierung mit zwei Sektoren. Weiterhin werden die Vor- und Nachteile der zwei Ansätze diskutiert.

Abschnitt 9.1 vergleicht zunächst die Ergebnisse für normale Testbilder, die der Standardeinstellung entsprechen. Abschnitt 9.2 behandelt die Lokalisierung von rotierten Bildern und Abschnitt 9.3 vergleicht die Fähigkeiten der Verfahren okkludierte Bilder zu klassifizieren. Abschließend werden in Kapitel 9.4 einige Optimierungsmöglichkeiten der Sektoren-Lokalisierung vorgestellt.

9.1 Normale Bilder

In diesem Abschnitt sollen die beiden Lokalisierungsmethoden zunächst daraufhin untersucht werden, wie sie mit normalen Testbildern umgehen. Normal bedeutet in diesem Fall, daß die Bilder der Standardeinstellung entsprechen, d.h. daß sie nicht rotiert sind, keine Okklusionen enthalten, etc. (vgl. S. 58). Diese Anforderung erfüllt der Testlauf bei Tag (Abbildung 7.1). Die Abbildungen 9.1 und 9.2 stellen die Kardinalzahlen und Positionierungsfehler der Vollbild-Lokalisierung und der 3-Sektoren-Lokalisierung für diesen Testlauf gegenüber.

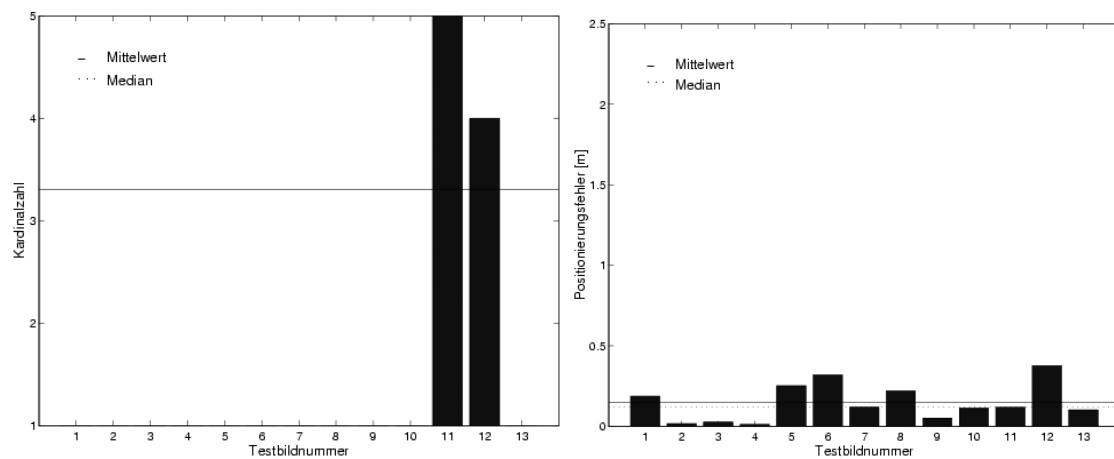
Die Kardinalzahlen in (a) weisen keinen großen Unterschied auf; die Sektoren-Lokalisierung klassifiziert zwei Bilder nicht perfekt, im Gegensatz zu einem Bild bei der Vollbild-Lokalisierung. Bei dem Testbild, das mit beiden Methoden schlecht klassifiziert wird, handelt es sich um Bild Nr. 11, das bereits in Abb. 7.5, S. 64 dargestellt wurde. Offenbar sind die Beleuchtungsvarianzen dieses Testbildes so gravierend, daß eine gute Klassifikation mit beiden Lokalisierungsmethoden nicht möglich ist. Die Werte der Positionierungsfehler in (b) zeigen, daß die Vollbild-Lokalisierung für normale Testbilder durchschnittlich besser ar-



(a) Kardinalzahl (Mittelwert: 1.23, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.07, Median: 0.04)

Abbildung 9.1: Vollbild-Lokalisierung des Testlaufs



(a) Kardinalzahl (Mittelwert: 3.31, Median: 1)

(b) Positionierungsfehler (Mittelwert: 0.15, Median: 0.12)

Abbildung 9.2: 3-Sektoren-Lokalisierung des Testlaufs

beitet. Der mittlere Positionierungsfehler der 3-Sektoren-Lokalisierung ist doppelt so hoch wie der Wert der Vollbildmethode, der Median beträgt gar den dreifachen Wert (siehe Tabelle 9.1). Dennoch kann man sagen, daß die Sektoren-Lokalisierung erfolgreich arbeitet, da ein Großteil der Bilder richtig klassifiziert wird und der mittlere Positionierungsfehler nur 15 cm beträgt.

Lokalisierungsmethode	Kardinalzahl Mittelwert	Kardinalzahl Median	Pos.fehler Mittelwert [m]	Pos.fehler Median [m]
Vollbild-Lokalisierung	1.23	1	0.07	0.04
3-Sektoren-Lokalisierung	3.31	1	0.15	0.12

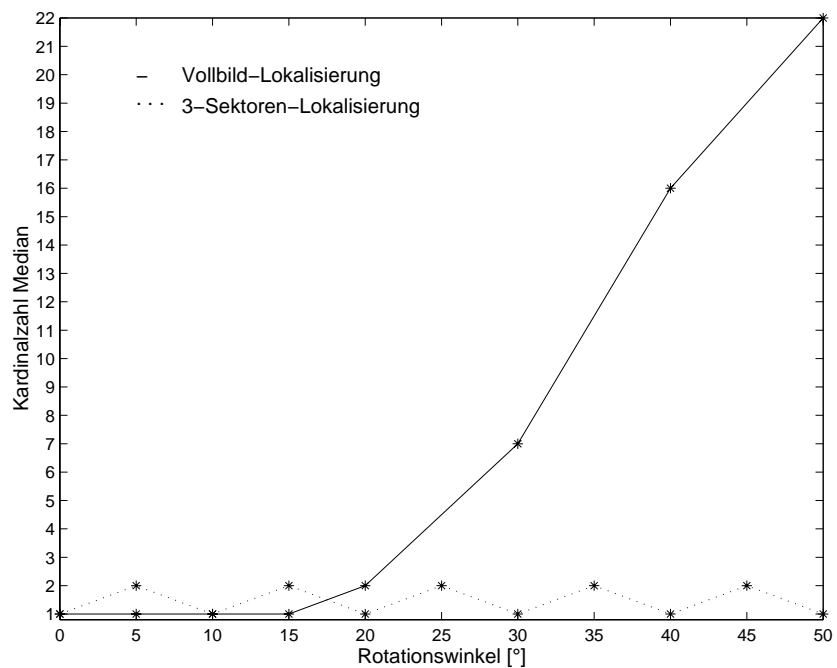
Tabelle 9.1: Klassifikation des Testlaufs, jeweils 13 Testbilder.

Insgesamt schneidet die Vollbild-Lokalisierung für normale Testbilder jedoch deutlich besser ab als die Sektoren-Lokalisierung. Der Grund ist, daß die gesamten Panoramabilder sehr viel mehr Information beinhalten als einzelne Sektoren und geringe Schwankungen in den Testdaten weniger ins Gewicht fallen. In Abschnitt 9.4 werden Möglichkeiten zur Verbesserung der Sektoren-Lokalisierung diskutiert.

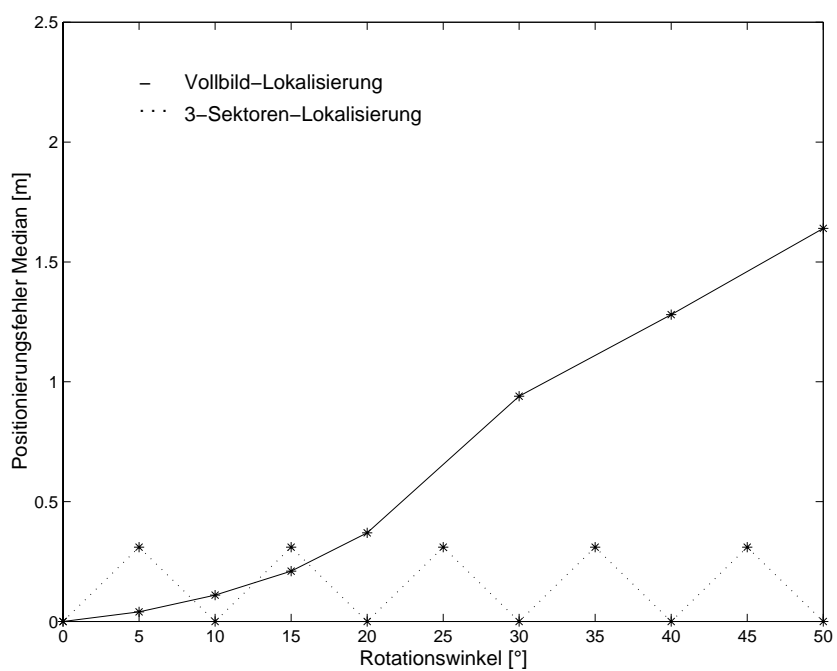
9.2 Rotationen

Dieser Abschnitt stellt die beiden Lokalisierungsmethoden im Bezug auf ihre Fähigkeit, rotierte Bilder zuzuordnen, gegenüber. In den vorherigen Kapiteln haben wir gesehen, daß die Vollbild-Lokalisierung robust gegenüber Rotationen bis etwa 15 % ist, bei höher werdenden Rotationswinkeln aber immer schlechtere Ergebnisse erzielt. Die Sektoren-Lokalisierung dagegen ist anfälliger gegenüber kleinen Rotationen, dafür spielen Rotationen um ein Vielfaches von 10 keine Rolle, da an diesen Stellen jeweils ein neuer Sektor beginnt. Abbildung 9.3 veranschaulicht diesen Sachverhalt. In (a) sind die Mediane der Kardinalzahlen rotierter Bilder bis 50° dargestellt, (b) zeigt die Mediane der Positionierungsfehler. Die durchgezogene Linie stellt jeweils die Werte der Vollbild-Lokalisierung dar, die gestrichelte Linie repräsentiert die Werte der 3-Sektoren-Lokalisierung. In beiden Grafiken erkennt man, daß die Werte der Sektoren-Lokalisierung um 5° und 15° über den Werten der Vollbild-Lokalisierung liegen. Ab einem Rotationswinkel von 20° zeigen sich dann die Vorteile der Aufteilung in Sektoren: Während die Fehler bei der Vollbild-Lokalisierung ab dieser Stelle drastisch ansteigen, bleiben die Ergebnisse der Sektorenmethode konstant. Die Mediane der Kardinalzahlen schwanken hier zwischen 1 und 2, die der Positionierungsfehler zwischen 0 und 30 cm. Diese Maximalwerte sind tatsächlich immer gleich hoch, da jeweils die gleichen Sektoren betrachtet werden, nur in unterschiedlicher Reihenfolge.

Mit den in Kapitel 9.4 vorgeschlagenen Optimierungsmöglichkeiten sollten diese Werte noch zu verbessern sein, so daß die Sektoren-Lokalisierung auch für kleine Rotationschwankungen robust ist.



(a) Kardinalzahlen



(b) Positionierungsfehler

Abbildung 9.3: Mediane rotierter Bilder von Vollbild und 3-Sektoren-Lokalisierung, pro Rotationswinkel 41 Bilder.

9.3 Okklusionen

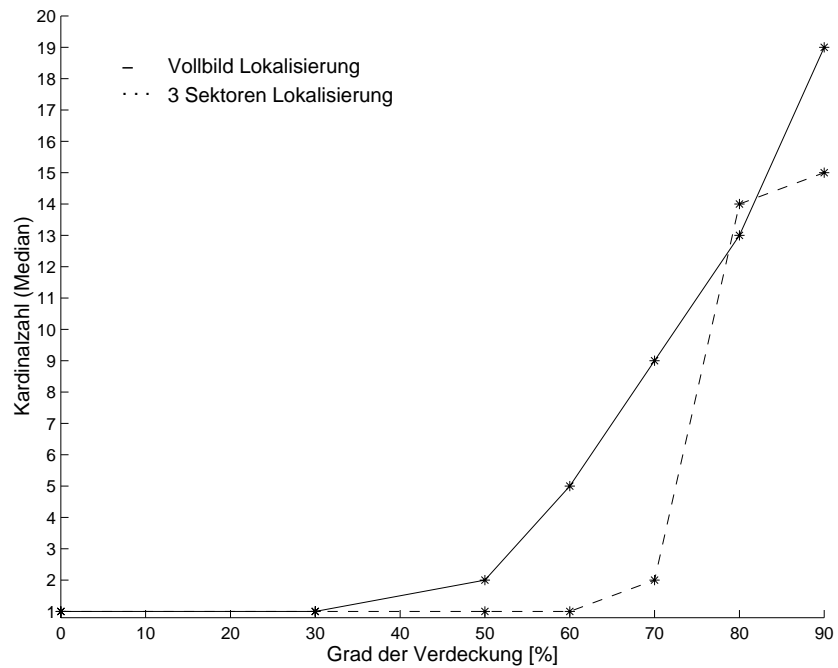
Dieser Abschnitt vergleicht die Vollbild-Lokalisierung und die 3-Sektoren-Lokalisierung bezüglich ihrer Fähigkeit, okkludierte Bilder zu klassifizieren. Zunächst findet ein Vergleich der künstlich okkludierten Bilder statt, um die Güte der Lokalisierungsmethoden abhängig vom Grad der Verdeckung testen zu können. Danach folgt ein Vergleich anhand realer okkludierter Testbilder.

9.3.1 Künstliche Okklusionen

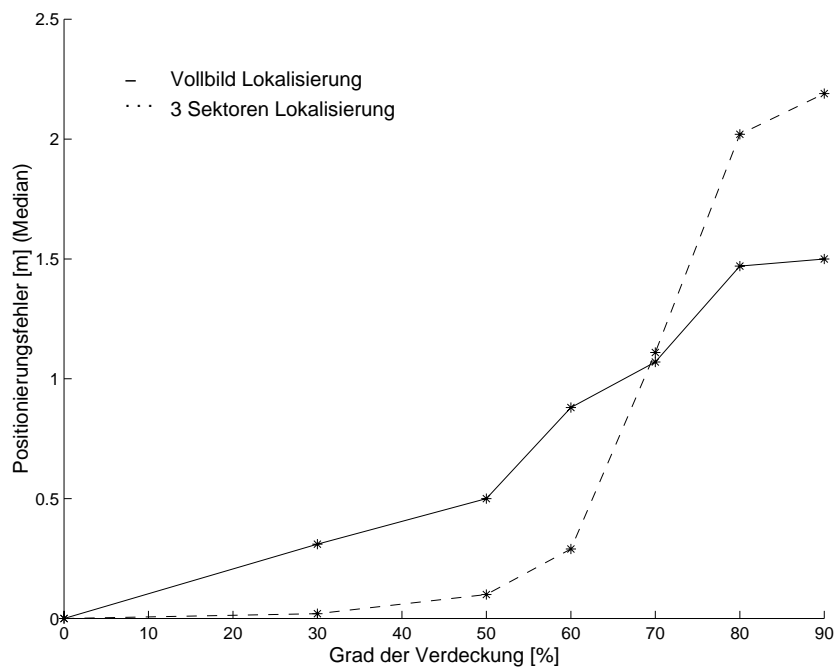
In diesem Abschnitt werden die Lokalisierungsergebnisse der Vollbild- und der 3-Sektoren-Lokalisierung bezüglich künstlicher Okklusionen dargestellt. Die Ergebnisse der Sektoren-Lokalisierung wurden bereits im vorherigen Kapitel präsentiert (Abb. 8.6 – 8.11); dieselben Experimente wurden ebenfalls mit der Vollbild-Lokalisierung durchgeführt. Abbildung 9.4 zeigt die Mediane der Kardinalzahlen und Positionierungsfehler beider Methoden bezüglich Verdeckungsgraden von 0 bis 90 %. Da die Mittelwerte keine wesentlichen Unterschiede erkennen lassen, beschränken wir uns auf diese Darstellung. Die gestrichelte Linie stellt jeweils die Werte der Sektoren-Lokalisierung dar, die durchgezogene Linie zeigt die Werte der Vollbild-Lokalisierung. Eine Betrachtung des Positionierungsfehlers in (b) zeigt, daß die Kurve der Vollbild-Lokalisierung relativ gleichmäßig ansteigt. Schon bei 30 % iger Verdeckung ist ein deutlicher Fehler erkennbar, der bis 60 % auf fast 1 m ansteigt. Bis zu dieser Stelle verläuft die Kurve der 3-Sektoren-Lokalisierung deutlich flacher. Bis zu 50 % Verdeckung liegen die Werte nah bei 0, erst ab 60 % tritt ein erwähnenswerter Fehler auf, der aber immer noch deutlich unter dem Wert der Vollbild-Lokalisierung liegt. Ab 70 % Verdeckung überschneiden sich die Kurven, bei höheren Werten liefert sogar die Vollbild-Lokalisierung die besseren Ergebnisse. Ähnliche Werte zeigt die Grafik der Kardinalzahlen in (a). Hier zeigt die Sektoren-Lokalisierung allerdings noch bei einer Verdeckung von 70 % sehr viel bessere Ergebnisse als die Vollbildmethode. Erst ab einer 80 % igen Verdeckung schneiden sich die Kurven. Der Grund für den starken Anstieg der Sektoren-Lokalisierung ab einem gewissen Verdeckungsgrad ist, daß ab dieser Stelle keine drei unverdeckten Sektoren vorhanden sind, die zur Klassifikation herangezogen werden könnten. Da die Sektoren eine recht geringe Informationsmenge beinhalten, werden sie durch Störungen stärker beeinflußt als die Vollbild-Lokalisierung. Etwas bessere Ergebnisse im Bereich sehr hoher Okklusionen bietet die 2-Sektoren-Lokalisierung, da bei dieser nur zwei unverdeckte Sektoren vorhanden sein müssen. (vgl. Kapitel 8.3). Da diese aber bei geringerem Verdeckungsgrad schlechtere Ergebnisse liefert und, wie wir in Kapitel 8.1 gesehen haben, zudem schlechter in der Lage ist mit echten Testbildern umzugehen, ist dennoch die 3-Sektoren-Lokalisierung vorzuziehen.

9.3.2 Natürliche Okklusionen

In diesem Abschnitt wird die Klassifizierung von realen okkludierten Bildern mit den zwei Lokalisierungsmethoden verglichen. Als Testdaten dienten die in Kapitel 7.3.2 vorgestellten



(a) Kardinalzahlen



(b) Positionierungsfehler

Abbildung 9.4: Mediane von Vollbild- und 3-Sektoren-Lokalisierung, pro Verdeckungsgrad 41 Bilder.

Bilder aus Umgebung 1 mit Personen und Gegenständen. Abbildungen 9.5 und 9.6 zeigen die Kardinalzahlen und Positionierungsfehler der Experimente mit der Vollbild- und mit der 3-Sektoren-Lokalisierung. Die Grafiken der Kardinalzahlen zeigen bei der Sektoren-Lokalisierung eine deutliche Verbesserung der Klassifikation. Die Positionierungsfehler unterscheiden sich dagegen kaum. Ähnliches zeigt auch Tabelle 9.2, in der die Mittelwerte und Mediane beider Lokalisierungsmethoden gegenübergestellt sind. Der Median der Kar-

Lokalisierungsmethode	Kardinalzahl Mittelwert	Kardinalzahl Median	Pos.fehler Mittelwert [m]	Pos.fehler Median [m]
Vollbild-Lokalisierung	5.63	4	0.97	0.57
3-Sektoren-Lokalisierung	4.68	2	1.15	0.64

Tabelle 9.2: Klassifikation realer okkludierter Bilder, jeweils 19 Testbilder.

dinalzahlen bei der Sektoren-Lokalisierung hat sich von 4 auf 2 halbiert. Die Positionierungsfehler liegen dagegen noch leicht über den Werten der Vollbild-Lokalisierung. Daß die Sektoren-Lokalisierung keine besseren Werte liefert, liegt an der schon beschriebenen Problematik der Zuordnung von Testbildern. Auch die unverdeckten Sektoren variieren von den Sektoren der Trainingsbilder und durch die geringe Information in den Sektoren treten Mehrdeutigkeiten auf. Im folgenden Abschnitt werden Möglichkeiten zur Verbesserung des Verfahrens diskutiert.

9.4 Diskussion

In diesem Kapitel haben wir gesehen, wie die Vollbild- und die Sektoren-Lokalisierung arbeiten. Es hat sich gezeigt, daß die Vollbildmethode auf normalen Testdaten zuverlässiger arbeitet. Aufgrund der großen Informationsmenge der Panoramabilder fallen kleine Schwankungen in den Testdaten und Rotationen bis zu 10° nicht so stark ins Gewicht wie bei den schmalen Sektorbildern. Bei größeren Rotationen und Verdeckung von Teilbereichen der Bilder stößt die Vollbild-Lokalisierung allerdings an ihre Grenzen. Hier zeigen sich die Vorteile der Sektoren-Lokalisierung. Um diese aber der Vollbild-Lokalisierung generell vorzuziehen, wäre eine Verbesserung des Verhaltens für normale Testbilder wünschenswert. Im folgenden sollen einige Verbesserungsmöglichkeiten diskutiert werden, deren Verwirklichung zukünftigen Arbeiten vorbehalten sei.

In Kapitel 8.1 haben wir gesehen, daß die Lokalisierung mit drei Sektoren deutlich bessere Ergebnisse liefert als die Verwendung von zwei Sektoren. Es liegt also nahe, daß ein Einbeziehen von mehr als drei Sektoren noch bessere Ergebnisse liefern sollte. Dabei treten allerdings Berechnungsprobleme auf, da ein wiederholtes Multiplizieren von Wahrscheinlichkeiten in der Update-Regel der Bayes'schen Formel (S. 44) sehr kleine Werte zur Folge hat, so daß durch Rundungsfehler die Wahrscheinlichkeiten zu null werden. Ein weiteres Problem bei der Verwendung von mehr Sektoren ist, daß für eine erfolgreiche Klassifikation

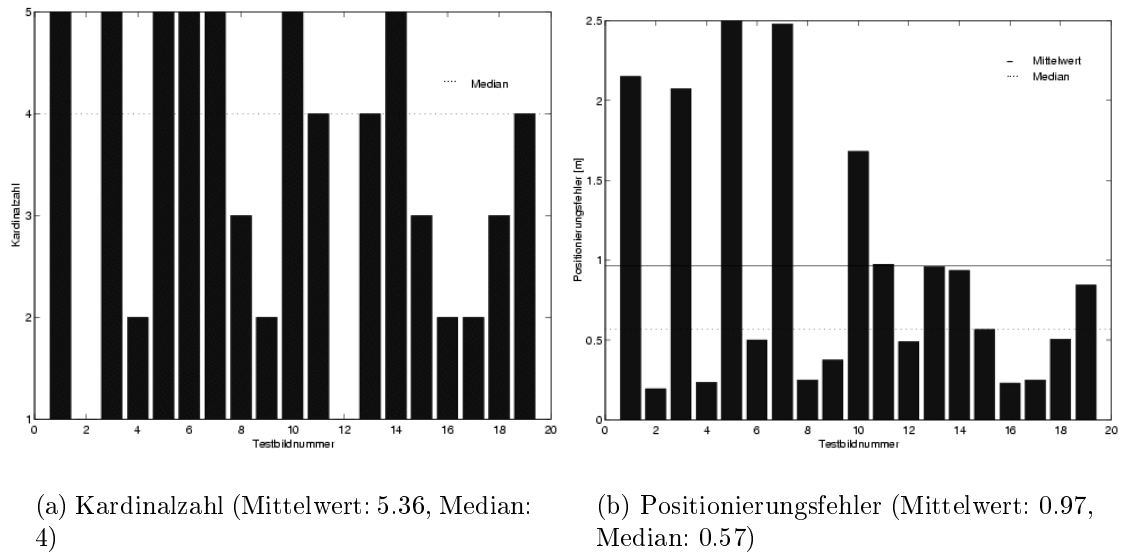


Abbildung 9.5: Vollbild-Lokalisierung realer Okklusionen

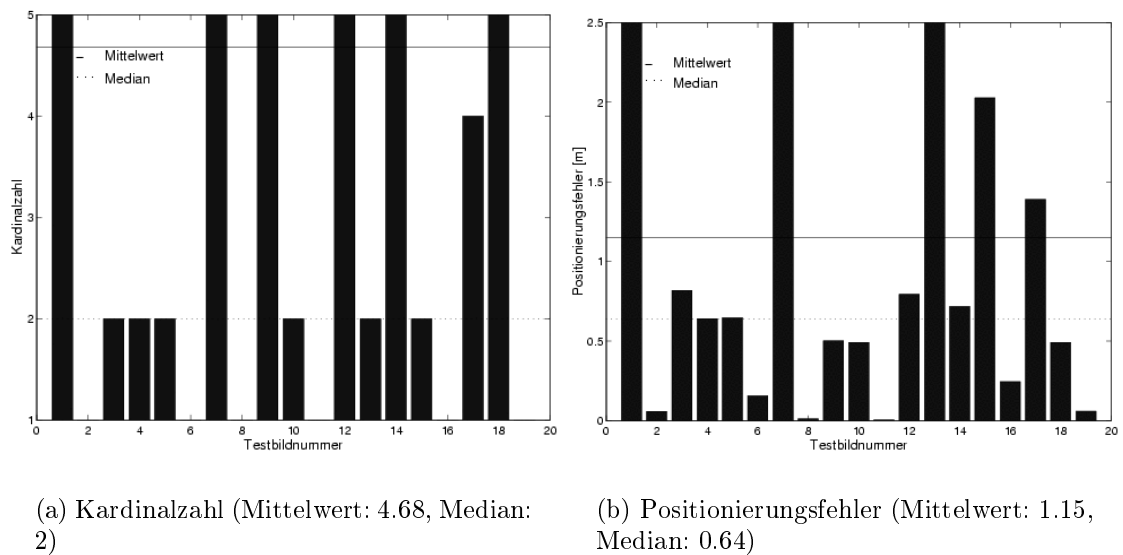


Abbildung 9.6: 3-Sektoren-Lokalisierung realer Okklusionen

mindestens so viele Sektoren unverdeckt sein müssen, wie bei der Sektoren-Lokalisierung in Betracht gezogen werden. Das ist gerade bei starken Okklusionen oft nicht der Fall. Es muß also ein Kompromiß gefunden werden zwischen dem Wunsch, die Klassifikation durch viele Sektoren zu verbessern, und dem Wunsch, mit okkludierten Bildern umgehen zu können. Eine Alternative wäre es, dynamisch so viele Sektoren einzubeziehen, wie in dem jeweiligen Testbild unverdeckt sind. Dazu müßte die Übereinstimmung eines Testsektors zu einem Trainingssektor einen bestimmten Schwellwert überschreiten, damit der Testsektor zur Klassifizierung herangezogen wird.

Eine andere Möglichkeit, das Verfahren zu verbessern, wäre, mehrere Sektorpaare (bzw. Sektortripel etc.) in Betracht zu ziehen. Bei der vorgestellten Methode wird nur die Sektorkombination betrachtet, die die höchste Wahrscheinlichkeit liefert. Wenn mehrere Sektorkombinationen, die alle für eine Position sprechen, fusioniert würden, gewichtet mit ihrer jeweiligen Wahrscheinlichkeit, könnten Mehrdeutigkeiten aufgelöst werden.

Weiterhin wäre es interessant, auszutesten, wie die Sektorbreite das Verfahren beeinflußt. Breitere Sektoren sollten bessere Klassifikationsergebnisse liefern, zu breite Sektoren erschweren allerdings wiederum die Behandlung von Okklusionen. Hier muß ein Kompromiß gefunden werden, so daß sowohl normale als auch okkludierte Testbilder zufriedenstellend klassifiziert werden.

Kapitel 10

Zusammenfassung und Ausblick

Dieses Kapitel faßt zunächst die relevanten Ideen und Konzepte sowie die erzielten Ergebnisse der Arbeit zusammen. Abschließend wird ein Ausblick über vielversprechende Erweiterungen und Ergänzungen gegeben, die im Zusammenhang mit den vorgestellten Verfahren stehen.

10.1 Zusammenfassung

In dieser Arbeit wurden zwei Verfahren zur Lokalisierung von Indoor-Robotern beschrieben. Als Sensor wurde jeweils eine omnidirektionale Kamera verwendet. Die Vorteile eines solchen Sensors liegen vor allem in der Informationsfülle, die ein solches Panoramabild bietet, und in der Flächenabdeckung der gesamten Raumhöhe. Die erste Lokalisierungsmethode arbeitet mit den gesamten Panoramabildern und wurde daher Vollbild-Lokalisierung genannt; außerdem wurde eine weitere Methode vorgestellt: die Sektoren-Lokalisierung. Dabei wurden die Panoramabilder in einzelne, sich überlappende Sektorbilder unterteilt, die zur Klassifikation genutzt wurden.

Um die Informationsfülle der Bilder geeignet nutzen zu können, ohne zu große Zeit- und Speicherressourcen zu verbrauchen, wurden die Bilder zunächst komprimiert. Als Kompressionsverfahren wurde PCA (Principal Component Analysis) gewählt, ein bekanntes Bildkompressionsverfahren. Dabei wird der Eigenraum der Kovarianzmatrix der Bilddaten erzeugt, und die Panoramabilder werden in diesen Eigenraum projiziert. Dieser besitzt die Eigenschaft, daß sich seine Dimension unter maximalem Informationserhalt drastisch verkleinern läßt. So erhielten wir an Stelle der Panoramabilder sehr viel kleinere Datenvektoren.

Die Zuordnung eines Bildes zu einer Roboterposition geschah mittels Appearance-based Matching, einer Methode, die ohne explizites Umgebungsmodell auskommt. Sie basiert darauf, daß sich Bilder benachbarter Positionen ähnlich sehen und benachbarten Punkten im Eigenraum entsprechen. Die Klassifikation wurde bei der Vollbild-Lokalisierung durch einen Nächster-Nachbar-Klassifikator realisiert.

Bei der Sektoren-Lokalisierung wurden die Sektorbilder einzeln in den Eigenraum pro-

jiziert und zur Lokalisierung genutzt. Da die Sektorbilder schmal sind und somit nur wenig Information enthalten, treten bei diesem Verfahren häufig Mehrdeutigkeiten auf. Diese konnten durch Betrachten des visuellen Kontextes der Sektoren verringert werden, welcher durch die Reihenfolge weiterer Sektoren desselben Panoramabildes repräsentiert wird. Die Klassifikation samt Berücksichtigung des Kontextes wurde durch Bayes'sches Update realisiert.

Ein Schwerpunkt wurde in dieser Arbeit darauf gelegt, die Robustheit der Verfahren zu testen und zu verbessern. So wurde untersucht, wie zwischen Bildern der Trainingspositionen interpoliert wird, welchen Einfluß Rotationen des Roboters haben, wie sich Beleuchtungsvarianzen auf die Klassifikation auswirken und wie die Methoden auf okkludierte Teilbereiche der Bilder reagieren. Um das Verhalten gegenüber Beleuchtungsvarianzen zu verbessern, wurden Tag-Nacht-Eigenräume generiert, die sowohl Tag- als auch Nachtbilder als Trainingsdaten verwenden. Mit dieser Methode wurden deutlich bessere Klassifikationsergebnisse von Nachtbildern erzielt.

Bei den Experimenten zur Sektoren-Lokalisierung wurde zunächst untersucht, wie sich die Anzahl verwendeter Sektoren auf die Klassifikationsgüte auswirkt. Dazu wurden die Experimente mit zwei und mit drei Sektoren durchgeführt. Es hat sich gezeigt, daß sich mit drei Sektoren durchschnittlich deutlich bessere Ergebnisse erzielen ließen. Weiterhin wurde anhand von Bildern, die systematisch zu einem gewissen Prozentsatz verdeckt wurden, das Verhalten des Verfahrens gegenüber Okklusionen getestet. Hierbei ergab sich, daß die Vollbild-Lokalisierung auf unverdeckten, nicht rotierten Testbildern zuverlässiger arbeitet, die Sektoren-Lokalisierung dagegen bei Okklusionen bis zu einer Verdeckung von etwa 70 % deutlich bessere Ergebnisse erzielt.

Abschließend kann gesagt werden, daß die vorgestellten Methoden eine interessante Ergänzung zu den gängigeren Lokalisierungsmethoden bieten, die auf Laser- oder Sonarsensoren basieren. Mit geeigneten Anpassungen sind die Verfahren weitgehend robust gegenüber diversen Störungen der Sensordaten. Die vorgestellte Sektoren-Lokalisierung stellt einen vielversprechenden Ansatz zur Behandlung starker Okklusionen dar, die in realen Umgebungen an der Tagesordnung sind. Es ist zu vermuten, daß diese Methode, ergänzt durch die in Kapitel 9.4 vorgeschlagenen Verbesserungsmöglichkeiten, Ergebnisse liefert, die die Vollbild-Lokalisierung auf ganzer Linie übertreffen.

10.2 Ausblick

Die in dieser Arbeit vorgestellten Verfahren stellen einen in sich geschlossenen Ansatz zur Lokalisierung von Indoor-Robotern dar. Es sind allerdings einige Optimierungs- und Ergänzungsmöglichkeiten denkbar, um die Funktionsweise der Verfahren zu verbessern. In diesem Abschnitt werden zunächst einige Optimierungsmöglichkeiten der Verfahren vorgestellt, danach wird auf verschiedene Erweiterungsmöglichkeiten eingegangen.

Zur Optimierung beider Lokalisierungsmethoden könnte beispielsweise die Zahl verwendeter Eigenvektoren variiert werden. In unseren Experimenten wurde die Zahl der Eigenvektoren auf 10 gesetzt, ausgehend von Experimenten ähnlicher Einsatzgebiete. Eine

genauere Analyse könnte zeigen, ob sich die Ergebnisse in diesem speziellen Anwendungsfall verbessern, wenn mehr Eigenvektoren verwendet werden, oder ob gar weniger Eigenvektoren ausreichen, um die gleichen Ergebnisse zu erzielen.

Weitere Optimierungsmöglichkeiten finden sich im Bereich der Sektoren-Lokalisierung. Im vorherigen Kapitel wurden bereits einige vielversprechende Ansätze diskutiert; so könnte beispielsweise die Breite der Sektoren variiert oder die Anzahl verwendeter Sektoren vergrößert werden. Alternativ könnte die Sektorenzahl dynamisch bestimmt werden, um sich an verschiedene Verdeckungsgrade anpassen zu können. Weiterhin wäre es möglich, alle Wahrscheinlichkeiten, die für eine Position sprechen, zu einem Wahrscheinlichkeitswert zu kombinieren.

Es folgen einige Ergänzungsmöglichkeiten der Lokalisierungsmethoden. Allerdings könnten Experimente, die auf den vorgestellten Trainingsdaten basieren, nicht ohne weiteres stattfinden, da inzwischen ein Gebäude an der Stirnseite des Flurs, an der sich das Fenster befand, angebaut wird. Für neue Experimente würden folglich neue Trainingsdaten benötigt.

Die Experimente dieser Arbeit fanden jeweils in einem der drei Abschnitte einer Büroumgebung statt. Wenn die Verfahren in ein globales Lokalisierungsverfahren eingebunden würden, könnte die zu verwendende Umgebung aus den Daten vorheriger Lokalisierungsschritte ermittelt und automatisch zwischen den Umgebungen gewechselt werden.

Weiterhin könnten einzelne Lokalisierungsschritte einer Testphase aufeinander aufbauen, so daß die Wahrscheinlichkeit, an einer Position zu sein, von der vorherigen Positionsschätzung abhängt. In dieser Arbeit wurden die Positionsschätzungen unabhängig von vorherigen Schätzungen generiert; in realen Einsatzgebieten könnten die bisherigen Ergebnisse durch Bayes'sches Update in die Positionsschätzungen miteinfließen, so daß die Ergebnisse im Laufe einer Testphase deutlich verbessert und Mehrdeutigkeiten aufgelöst werden könnten.

Interessant wäre auch die Einbindung der Lokalisierungsmethoden in bestehende Lokalisierungs-Systeme, die die Fusion verschiedener Sensoren gewährleisten. Dabei wird die Positionsschätzung aus den Informationen der Daten mehrerer Sensoren kombiniert. Ein Beispiel einer solchen Architektur ist die Rhino-Architektur (Burgard *et. al* [BBC⁺95]), die hauptsächlich mit Sonar- und Lasersensoren arbeitet, aber auch die Verwendung weiterer Sensoren ermöglicht.

Die Ergebnisse der in dieser Arbeit vorgestellten Lokalisierungsmethoden bieten vielversprechende Ansätze zur Einbindung in bestehende Robotik-Systeme und lassen deutliche Verbesserungen der Performance in realem Einsatz erhoffen.

Literaturverzeichnis

- [AIYT98] N. Aihara, H. Iwasa, N. Yokoya, and H. Takemura. Memory-based self-localization using omnidirectional images. In *Proc. International Conference on Pattern Recognition*, pages 1799–1803, 1998.
- [Ark98] R. C. Arkin. *Behavior-based Robotics*. The MIT Press, Cambridge, MA, 1998.
- [BBC⁺95] J. Buhmann, W Burgard, A.B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The Mobile Robot Rhino. *AI Magazin*, 16:1, 1995.
- [BCF⁺00] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000.
- [BFHS96] Wolfram Burgard, Dieter Fox, Daniel Hennig, and Timo Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the National Conference on Artificial Intelligence*, 1996.
- [BHK97] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.
- [Bis95] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, UK, 1995.
- [BL88] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.
- [BS95] A. J. Bell and T. J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
- [CGR95] R. Cassinis, D. Grana, and A. Rizzi. A perception system for mobile robot localization. In *WAAP95 Workshop congiunto su Apprendimento Automatico e Percezione, Ancona (Italy)*, 1995.

- [CGR96] R. Cassinis, D. Grana, and A. Rizzi. Using colour information in an omnidirectional perception system for autonomous robot localization. In *Proceedings of EUROBOT*, 1996.
- [Com94] P. Comon. Independent component analysis - A new concept? *Signal Processing*, 36:287–314, 1994.
- [Das94] B. Dasarathy. *Decision Fusion*. IEEE Computer Society Press, Los Alamitos, CA, 1994.
- [Dem98] James Demmel. New high accuracy algorithms for the eigenvalues and svd of structured matrices. Workshop on Complexity of Continuous and Algebraic Mathematics at the MSRI: mathematical sciences research institute, <http://msri.org/publications/ln/msri/1998/focm/demmel/1/index.html>, November 1998.
- [DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons Inc., New York, NY, 2001.
- [DJ00] Gregory Dudek and Michael Jenkin. *Computation Principles of Mobile Robotics*. Cambridge University Press, 2000.
- [DWM93] M. Dill, R. Wolf, and M. Heisenberg. Visual pattern recognition in drosophila involves retinotopic matching. *Nature*, 365:751–753, 1993.
- [EK98] R. Epstein and N. Kanwisher. A cortical representation of the local visual environment. *Nature*, 392:598–601, 1998.
- [Fel50] W. Feller. *An Introduction to Probability Theory and its Applications*. 1950. See the discussion of the birthday paradox in Section 2.3.
- [Fis36] R. A. Fisher. The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.
- [Fis86] Gerd Fischer. *Lineare Algebra*. Vieweg Verlag, 1986.
- [Fuk90] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, NY, 1990.
- [GW92] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.
- [Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, pages 417–441, 498–520, 1933.
- [IKH98] M. Ishikawa, S. Kawashima, and N. Homma. Memory-based location estimation and navigation using Bayesian estimation. In *Proc. International Conference on Neural Information Processing*, pages 112–117, 1998.

- [JC98] S.P.D. Judd and T.S. Collett. Multiple stored views and landmark guidance in ants. *Nature*, 392:710–714, 1998.
- [JKS95] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill, New York, NY, 1995.
- [JL99] M. Jogan and A. Leonardis. Panoramic eigenimages for spatial localization. In *Proc. International Conference on Computer Analysis of Images and Patterns*, pages 558–567, Ljubljana, Slovenia, 1999.
- [JL00] M. Jogan and A. Leonardis. Robust localization using panoramic view-based recognition. In *Proc. International Conference on Pattern Recognition*, pages 136–139, Barcelona, Spain, 2000.
- [Kar47] K. Karhunen. Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Mathematica-Physica*, 37:3–79, 1947.
- [KBVM99] B. J. A. Kröse, R. Bunschoten, N. Vlassis, and Y. Motomura. Appearance based robot localization. In G Kraetzschmar, editor, *Proc. IJCAI-99 Workshop on adaptive spatial representations of dynamic environments*, pages 53–58, 1999.
- [KH97] F. Kirchner and J. Hertzberg. A prototype study of an autonomous robot platform for sewerage system maintenance. *J. Autonomous Robots* 4(4), pages 319–331, 1997.
- [Kow79] Hans-Joachim Kowalsky. *Lineare Algebra*. de Gruyter, 1979.
- [Kur] Kurt-2 an universal platform for robot applications.
<http://ais.gmd.de/BAR/KURT2/>.
- [Loe63] M. Loeve. *Probability Theory*. Van Nostrand, New York, NY, 1963.
- [LW98] C. Liu and H. Wechsler. Face recognition using evolutionary pursuit. In *Proc. European Conference on Computer Vision*, 1998.
- [MIII99] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Visual navigation using omnidirectional view sequence. In *Proc. International Conference on Intelligent Robots and Systems*, pages 317–322, 1999.
- [MKS97] S. Maeda, Y. Kuno, and Y. Shirai. Active navigation vision based on eigen-space analysis. In *Proc. International Conference on Intelligent Robots and Systems*, pages 1018–1023. Grenoble, France, 1997.
- [MLR⁺01] Ralf Möller, Dimitrios Lambrinos, Thorsten Roggendorf, Rolf Pfeifer, and Rüdiger Wehner. Insect strategies of visual homing in mobile robots. In Barbara Webb and Thomas R. Consi, editors, *Biorobotics: Methods and Applications*. MIT-Press/AAAI Press, Cambridge, MA, 2001.

- [MN95] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.
- [Mor88] Hans Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.
- [Mur00] R. Murphy. *Introduction to AI robotics*. MIT Press, Cambridge, MA, 2000.
- [nav] <http://www.neuronics.ch>.
- [Nou98] Illah Nourbakhsh. Chapter 3. In R. Peter Bonasso David Kortenkamp and Robin Murphy, editors, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*. AAAI Press/ The MIT Press, 1998.
- [Pal99] Lucas Paletta. *Learning Active Object Recognition*. PhD thesis, Institute for Computer Graphics and Vision, Graz, Austria, 1999.
- [Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.
- [PFH01a] Lucas Paletta, Simone Frintrop, and Joachim Hertzberg. Localization by spatial context analysis. In *Proceedings of the 25th Workshop of the AAPR (ÖAGM 2001)*, pages 55–65, Berchtesgaden, Germany, June 2001.
- [PFH01b] Lucas Paletta, Simone Frintrop, and Joachim Hertzberg. Robust localization using context in omnidirectional imaging. In *Proceedings of the IEEE/RAS International Conference on Robotics and Automation (ICRA 2001)*, pages 2072–2077, Seoul, South Korea, May 2001.
- [PFH01c] Lucas Paletta, Simone Frintrop, and Joachim Hertzberg. Visual localization using bayesian decision fusion on omnidirectional sensing. In *Proceedings of the SPIE Sensor Fusion: Architectures, Algorithms, and Applications V, Aerosense 2001*, volume 4385, pages 58–66, Orlando, Florida, USA, April 2001.
- [PFTV88] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 1988.
- [PS99] R. Pfeifer and C. Scheier, editors. *Understanding Intelligence*. MIT Press, Cambridge, MA, 1999.
- [RHW86] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- [Rö98] T. Röfer. *Panoramic Image Processing and Route Navigation*. PhD thesis, Universität Bremen, Germany, 1998.

- [Thr98] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1), 1998.
- [TP91] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.
- [Web99] Andrew Webb. *Statistical Pattern Recognition*. Defence Evaluation and Research Agency, UK, 1999.
- [WGLSV00] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omnidirectional vision for robot navigation. In *Proc. IEEE Workshop on Omnidirectional Vision*, 2000.
- [WPB95] David Wettergreen, Henning Pangels, and John Bares. Behavior-based Gait Execution for the Dante II Walking Robot. In *Proceedings of IROS '95*, volume 3, pages 274 – 279, August 1995.

Erklärung

Name: Simone Frintrop
Mat.-Nr: 945132

Hiermit erkläre ich, daß ich die vorliegende Arbeit „Robuste Roboterlokalisierung mit omnidirektionaler Bildsensorik“ selbständig durchgeführt habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Bonn, den 31. Oktober 2001