

Technische Universität Kaiserslautern

DISSERTATION

**Bildbasierte Erzeugung
kollisionsfreier
Transferbewegungen für
Industrieroboter**

Zur Erlangung des akademischen Grades eines Doktors der
Ingenieurwissenschaften (Dr.-Ing.)

des Fachbereichs Informatik der Technischen Universität Kaiserslautern

vorgelegt von Dirk Ebert aus Freiburg im Breisgau

Datum der Einreichung: 23. Juni 2003

Danke

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe „Eingebettete Systeme und Robotik“ des Fachbereichs Informatik an der Technischen Universität Kaiserslautern in der Zeit vom März 2000 bis Juni 2003.

Meinem Doktorvater, Herrn Prof. Dr. rer. nat. Dominik Henrich, gilt mein besonderer Dank für seine Unterstützung und Betreuung meiner Arbeit. Durch die vielen anregenden und fruchtbaren Diskussionen hat er wesentlich zum Erfolg beigetragen.

Meinen Kollegen in der Arbeitsgruppe danke ich für ihre Anregungen, Diskussionen und Kritik. Besonderer Dank geht an Tun Schlechter, der mit seiner Neigung zur Theoretischen Informatik einen Gegenpol zu meinem eher ingenieurmäßigen Vorgehen bildete.

Ohne die tatkräftige Hilfe von Studenten, die sich in ihren Projekt- und Diplomarbeiten oder als Hiwi mit Teilproblemen beschäftigten, wäre diese Arbeit nicht möglich gewesen. Besonderer Dank geht an Thorsten Gecks, der mit seiner Erfahrung als C++ Programmierer viel zur Realisierung des Systems beigetragen hat und nun als Kollege die aufgeworfenen Fragen weitergehend untersucht, und an Frank Heinzen für die Hilfe bei der Durchführung vieler Experimente. Dank geht auch an Evelina Chamarova, Thomas Diehl, Marco Fandel, Marcel Heintz, Jan Koch, Jens Körte, Christoph Urbanczik und Pascal Wenzel.

Zusammenfassung

Aus Sicherheitsgründen müssen zur Zeit die Arbeitsräume von Industrierobotern und Menschen durch Schutzeinrichtungen getrennt werden. Dies kommt daher, dass die Roboter ihre Umwelt nicht wahrnehmen können. Für viele Bereiche ist es wünschenswert, Mensch und Roboter im selben Arbeitsraum arbeiten zu lassen.

Die für einen Menschen im Arbeitsraum gefährlichste Bewegung ist die freie Transferbewegung, denn dabei werden schnelle Bewegungen über größere Distanzen ausgeführt. Die betrachtete Aufgabe besteht daher darin, die Transferbewegungen eines Manipulators kollisionsfrei durchzuführen. Von einer gegebenen Startkonfiguration aus soll der Manipulator auf einer möglichst kurzen, kollisionsfreien Bahn in eine gegebene Zielkonfiguration fahren.

Das realisierte System gliedert sich in die vier Hauptkomponenten Bildverarbeitung, Robotermodellierung, Kollisionserkennung und Bahnplanung.

Aufgabe der Bildverarbeitung ist, ein Silhouettenbild der aktuellen Szene zu erstellen. Ein Silhouettenbild ist ein binäres Bild, in dem alle Vordergrundpixel gesetzt und alle Hintergrundpixel gelöscht sind. Die robuste und schnelle Erstellung von Silhouettenbildern aus Grauwertbildern ist ein relativ schwieriges Problem.

Die Aufgabe der Robotermodellierung besteht darin, die Kollisionserkennung mit Wissen über die Geometrie des Roboters zu versorgen. Im realisierten System wird eine silhouettenbasierte Kollisionserkennung verwendet. Daher muss hier die Robotermodellierung zu einer angefragten Konfiguration ein Silhouettenbild des Roboters liefern.

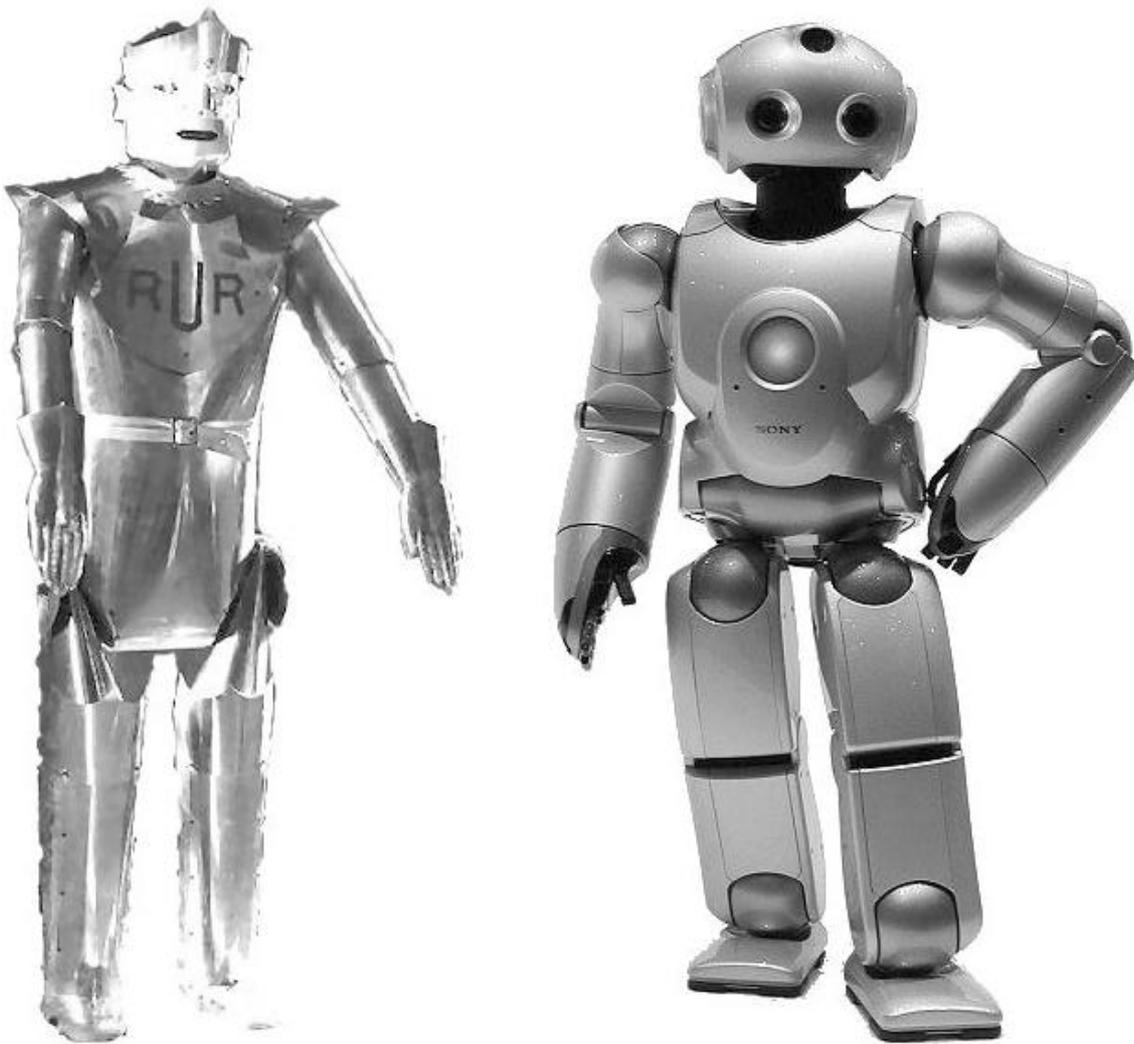
Die Aufgabe der Kollisionserkennung ist, zu entscheiden, ob eine bestimmte Konfiguration zu einer Kollision mit einem Objekt führen würde, wenn der Roboter in dieser Konfiguration wäre. Dabei wird lediglich diese angefragte Konfiguration betrachtet, egal ob es möglich ist, diese Konfiguration aus der gegenwärtigen Konfiguration zu erreichen. Der verwendete Kollisionstest baut keine Repräsentation des 3D-Arbeitsraumes auf, sondern arbeitet ausschließlich auf den Silhouettenbildern von Szene und Roboter. Dieser Kollisionstest basiert auf dem Schnitt von Szenensilhouette, aktueller und zu testender Robotersilhouette.

Die Bahnplanung hat die Aufgabe, ausgehend von einer gegebenen Startkonfiguration, eine kollisionsfreie Bahn zu einer gegebenen Zielkonfiguration zu finden. Dieses Problem ist in der Robotik gut untersucht. Bislang wurden jedoch hauptsächlich Probleme betrachtet, in denen der betrachtete Raum statisch und vollständig bekannt ist. Im Anwendungsszenario ist die Umgebung unbekannt und ändert sich ständig. Dies stellt besondere Anforderungen an die Bahnplanung. In der klassischen Bahnplanung kann dem Ergebnis der Kollisionserkennung blind vertraut werden. Bei einem System, das die Umwelt mit Sensoren erfasst, ist dies nicht so, da in der Sensorerfassung immer Fehler auftreten können.

Das oben beschriebene Systemkonzept wurde realisiert. Die verwendete Hardware besteht aus einem konventionellen 6-Achs Industrieroboter, Grauwertkameras und einem Standard-PC zur Bildverarbeitung und Bahnplanung. Die Versuche zeigen, dass der Ansatz zur Lösung der gestellten Aufgabe geeignet ist. Mögliche Erweiterungen des Systems sind die direkte Mensch-Roboter-Kooperation oder das gemeinsame Arbeiten an Werkstücken.

„It is difficult to say what is impossible, for the dream of yesterday is the hope of today and the reality of tomorrow.“

Robert H. Goddard



*Abbildung 1: Links: R.U.R. aus dem gleichnamigen Theaterstück, 1921 [1]
Rechts: Sony Dream Robot SDR-4X, 2002 [2]*

Inhaltsverzeichnis

Kapitel 1 Einleitung	1
1.1 Motivation.....	1
1.2 Ziele	3
1.3 Aufgabenstellung.....	4
1.4 Abgrenzung.....	5
1.5 Kapitelübersicht.....	6
Kapitel 2 Stand der Forschung.....	7
2.1 Kriterien zur Klassifizierung und Bewertung der Systeme	7
2.1.1 Erkennungsstrategie	7
2.1.2 Reaktionsstrategie	9
2.2 Klassifizierung der existierenden Systeme	12
2.2.1 Systeme mit Kollisionsentschärfung.....	12
2.2.2 Systeme mit Geschwindigkeitsregelung	13
2.2.3 Systeme mit lokaler Ausweichbewegung	16
2.3 Schlussfolgerungen.....	21
Kapitel 3 System zur sicheren Transferbewegung	22
3.1 Konkretisierung der Aufgabenstellung.....	22
3.2 Systemaufbau.....	23
3.2.1 Zusammenhang der Systemkomponenten.....	23
3.2.2 Zusammenfassung der Systemkomponenten	24
3.3 Auswahl der Systemkomponenten	25
3.3.1 Externe Sensorik	25
3.3.2 Umweltmodell.....	33
3.3.3 Robotermodell.....	35
3.3.4 Kollisionserkennung	36
3.3.5 Kollisionsvermeidung	37
3.4 Schlussfolgerungen.....	41
3.4.1 Systemkonzept	42
3.4.2 Bildbasierter Ganzarmkollisionstest	44
3.4.3 Vorgehen	44
3.4.4 Mathematische Formulierung des Problems.....	45
Kapitel 4 Silhouettenerzeugung von Szene und Roboter.....	48
4.1 Erzeugung von Szenensilhouetten aus Kamerabildern.....	48
4.1.1 Stand der Forschung.....	49
4.1.2 Bewertung der Silhouettenerzeugung	50
4.1.3 Ansatz.....	51
4.1.4 Merkmalsbilderstellung.....	52

4.1.5 Differenzbilderstellung	56
4.1.6 Referenzbilderstellung	59
4.1.7 Kamerakalibrierung	62
4.2 Silhouettenerzeugende Robotermodellierung.....	64
4.2.1 Bilddatenbank	64
4.2.2 Modellberechnung	65
4.2.3 Anpassung eines Modells.....	66
4.2.4 Modellierung von Konfigurationsfolgen	66
4.2.5 Ergebnis	69
Kapitel 5 Silhouettenbasierter, globaler Ganzarmkollisionstest...70	
5.1 Stand der Forschung	70
5.2 Kollisionserkennung im Bildraum der Kameras	72
5.2.1 Formale Beschreibung der Kollisionserkennung.....	72
5.2.2 Grundalgorithmus	73
5.2.3 Verfahren für Szenen mit Roboter	75
5.2.4 Unkalibrierte Kameras	77
5.2.5 Kalibrierte Kameras	77
5.3 Vergleich mit der Rekonstruktion des Arbeitsraums.....	79
5.4 Experimente.....	81
5.4.1 Versuchsaufbau	81
5.4.2 Bewertung der Kollisionserkennung.....	82
5.4.3 Versuchsdurchführung	82
5.4.4 Unbewegliches Hindernis	84
5.5 Gleichzeitiger Kollisionstest für mehrere Konfigurationen	86
5.5.1 Gleichzeitiges Testen von Punkt-zu-Punkt Bewegungen	86
5.5.2 Tabellenbasierte Rekonstruktion des gesamten Konfigurationsraumes ..	89
5.6 Berücksichtigung statischer Hindernisse	93
5.7 Zusammenfassung	93
Kapitel 6 Systemintegration	94
6.1 Bahnplanung	94
6.1.1 Verwendete Bahnplaner	94
6.1.2 Mehrstufige Bahnplanung.....	94
6.2 Erzeugung kontinuierlicher Bewegungen.....	97
6.2.1 Problem	97
6.2.2 Sequentieller Kontrollfluss	97
6.2.3 Paralleler Kontrollfluss	97
6.2.4 Experimente	98
6.3 Sicherheit	100
6.3.1 Stand der Forschung.....	100

6.3.2 Risiko- und Ursachenanalyse.....	101
6.3.3 Problematik der Sensorfehler.....	102
6.3.4 Erkennung von Sensorfehlern.....	102
6.3.5 Fehler durch unerwartete Umweltzustände.....	103
6.3.6 Kamerafehler.....	104
6.3.7 Erkennung von Bildrauschen und Standbildern.....	105
6.3.8 Falsche Kamerapositionierung.....	106
6.3.9 Zusammenfassung.....	107
6.4 Verwendung des Systems.....	108
6.4.1 Ausführungsablauf des Befehls zur sicheren Transferbewegung.....	108
6.4.2 Umgang mit Konfigurationen außerhalb des Planungsraumes.....	109
Kapitel 7 Experimente.....	111
7.1 Versuchsaufbau.....	111
7.2 Versuchsdurchführung.....	113
7.2.1 Passieren in der Roboterzelle.....	114
7.2.2 Wartung der Maschine.....	115
7.2.3 Aufräumen in der Arbeitszelle.....	117
7.2.4 Ergebnis.....	118
7.3 Diskussion der erzielten Ergebnisse.....	118
7.3.1 Einschränkungen des Systems.....	118
7.3.2 Leistung des Systems.....	120
Kapitel 8 Zusammenfassung und Ausblick.....	121
8.1 Zusammenfassung.....	121
8.2 Ausblick.....	122
Anhang A: Literaturverzeichnis.....	124
Anhang B: Sonstige Quellen.....	130
Lebenslauf des Verfassers.....	131

Kapitel 1

Einleitung

1.1 Motivation

Seit langem besteht bei den Menschen der Wunsch nach einem mechanischen Arbeiter, der den Menschen von gefährlichen, anstrengenden oder lästigen Tätigkeiten befreit. Dieser Wunsch wird bereits im Begriff „Roboter“ sichtbar, der auf dem slawischen Wortstamm *robot* (Fronarbeit) basiert.¹ Eine Grundvoraussetzung für einen wirklich hilfreichen Arbeiter ist, dass er in der Nähe von Menschen und gemeinsam mit Menschen arbeiten kann.

Den in der industriellen Produktion eingesetzten Manipulatoren fehlt in der Regel die Fähigkeit, ihre Umwelt umfassend wahrzunehmen. Die Roboter besitzen nur diejenigen Sensoren, die sie für die Erfüllung ihrer Aufgabe benötigen. Dies hat den Nachteil, dass die Roboter Hindernisse, die sich in ihrem Arbeitsraum befinden, nicht wahrnehmen können. Daher können bei der Bewegung von Industrierobotern in unvollständig bekannten Umgebungen Kollisionen auftreten. Dies gilt insbesondere dann, wenn bewegliche Hindernisse, wie zum Beispiel Menschen, im Arbeitsraum sind. Je nach Masse und Geschwindigkeit des Roboterarms kann die Kollision zu schweren Verletzungen bis hin zum Tod führen.

Die aktuellen Sicherheitsvorschriften [ISO10218] schreiben daher vor, dass Industrieroboter in abgegrenzten Bereichen („Zellen“) arbeiten müssen. Diese Bereiche müssen so abgesichert sein, dass der Roboter deaktiviert wird, sobald die Abgrenzung während des Automatikbetriebes durchbrochen wird.

Allerdings ist eine strikte Abschottung des Roboters von der Umwelt bei vielen Aufgaben nicht wünschenswert oder nicht möglich. Industrieroboter und Mensch haben unterschiedliche Stärken. Industrieroboter sind schnell, stark und positionsgenau. Menschen dagegen sind geschickt bei komplizierten Montagevorgängen und können sehr flexibel auf ungeplante Situationen reagieren. Um diese Stärken optimal zu kombinieren, ist in vielen Fällen eine enge Zusammenarbeit zweckmäßig.

Dies sind zum Beispiel Hol- und Bringdienste, bei denen der Roboter das benötigte Werkstück oder Werkzeug zum menschlichen Arbeiter bringt. Auf diese Weise kann die effektive Arbeitszeit des Arbeiters vergrößert werden und er muss seine Konzentration

¹ Der Begriff Roboter wurde durch den tschechischen Schriftsteller Karel Capek im Jahr 1920 in seinem Stück "R.U.R." (Rossum's Universal Robots) eingeführt. Die Erstaufführung war 1921. [3], [4]

nicht durch überflüssige Tätigkeiten unterbrechen. Eine besondere Form der Hol- und Bring-Dienste realisiert das System, das in [Schraft98] beschrieben ist. Der Roboter führt die Funktion eines Barkeepers aus. Der Roboter kann Sektflaschen öffnen, den Sekt in Gläser füllen und die gefüllten Gläser an Übergabepunkten abstellen. Beispiele für Systeme zur Erfüllung von Hol- und Bringdiensten sind in Abbildung 2 zu sehen.



Abbildung 2: Links: System zum Bringen von Werkstücken (aus [5])
Rechts: Sektroboter (aus [Schraft98])

Ein weiteres Einsatzgebiet ist das Halten von schweren Werkstücken derart, dass der menschliche Arbeiter in ergonomisch günstiger Haltung arbeiten kann. In Abbildung 3 ist ein Demonstrationssystem für diese Aufgabe zu sehen.



Abbildung 3: System zum Halten von schweren Werkstücken (aus [6])

Weitere Gründe für das Durchbrechen von trennenden Schutzeinrichtungen sind die erhöhte Leistungsfähigkeit und Flexibilität durch die engere räumliche Zusammenarbeit von Mensch und Roboter. Einige Aufgaben könnten im Prinzip auch durch Systeme mit trennenden Schutzeinrichtungen gelöst werden, der Ablauf wird aber durch die Aufhebung der Trennung einfacher.

Ein Beispiel für solche Aufgaben sind Einlegestationen, bei denen der Mensch die komplizierte Aufgabe des „Griffs in die Kiste“ übernimmt und die eigentliche Bearbeitung durch den Roboter erfolgt. Ein anderes Beispiel ist die Prozesszustandsüberwachung von Schweißarbeiten des Roboters durch direkte optische Kontrolle seitens des menschlichen Arbeiters. Kommerzielle Systeme für beide Aufgaben sind in Abbildung 4 zu sehen.

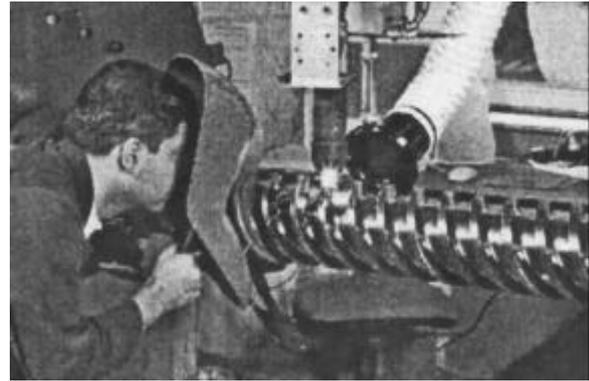
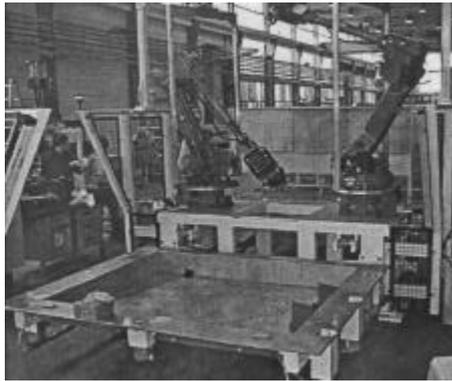


Abbildung 4: Links: Durch Arbeiter bestückte Einlegestation (aus [Zettl02])
Rechts: Schweißüberwachung durch Arbeiter (aus [Morhard02])

Aus diesen Gründen gibt es zur Zeit das Bestreben, Roboterarbeitszellen ohne trennende Schutzeinrichtungen (OTS) zu entwickeln.

1.2 Ziele

Die verschiedenen Arten der Interaktion zwischen Mensch und Roboter lassen sich als Multiagentenproblem beschreiben. In [Ferber99] werden die verschiedenen Arten anhand der Ziele und Leistungsfähigkeit der Agenten und der zu Verfügung stehenden Ressourcen unterschieden. Diese Klassifizierung ist in Tabelle 1 dargestellt und wird im Folgenden auf die Aufgabenstellung angewendet.

Ziele der Agenten verträglich?	Zur Verfügung stehende Ressourcen ausreichend?	Fähigkeit der Agenten zur individuellen Zielerreichung?	Art der Situation	Klasse der Situation
+	+	+	Unabhängigkeit	Indifferenz
+	+	-	einfache Kollaboration	Kooperation
+	-	+	Behinderung	
+	-	-	koordinierte Kollaboration	
-	+	+	rein individuelle Konkurrenz	Antagonismus
-	+	-	rein kollektive Konkurrenz	
-	-	+	individuelle Ressourcenkonflikte	
-	-	-	kollektive Ressourcenkonflikte	

Tabelle 1: Klassifizierung der Interaktionsarten. Die sichere Mensch-Roboter-Koexistenz gehört zur Klasse „Behinderung“. (nach [Ferber99])

Eine Grundvoraussetzung für das gemeinsame Arbeiten von Mensch und Roboter ist das Garantieren der Sicherheit für den Menschen, der durch potentielle Kollisionen mit dem Roboter gefährdet ist. Die Gefährdung des Menschen liegt an der unzureichenden Ressource „Raum“. Unter der Annahme, dass die Ziele von Mensch und Roboter verträglich sind, ist die Art der Situation entweder „Behinderung“, falls die Leistungsfähigkeit von Mensch und Roboter groß genug ist, die eigenen Ziele zu erreichen oder

„koordinierte Kollaboration“, falls die Leistungsfähigkeit nicht ausreicht. Das Lösen der Situation „Behinderung“ ist die Voraussetzung für das Lösen der schwierigeren Situation „koordinierte Kollaboration“.

Das Ziel dieser Arbeit lässt sich in der Terminologie der Multiagentensysteme beschreiben als „Lösung der Situation ‚Behinderung‘ bezüglich der knappen Ressource ‚Raum‘ mit den Agenten ‚Mensch‘ und ‚Roboter manipulator‘“. Ein System, das diese Situation löst, ermöglicht die sichere Mensch-Roboter-Koexistenz.

1.3 Aufgabenstellung

Die Gefährdung des Menschen durch den Roboter ist je nach Bewegungsart des Roboters unterschiedlich. Die Gefährdung kann als Risiko R betrachtet werden, bei dem mit einer bestimmten Eintrittswahrscheinlichkeit P ein Schaden der Höhe S eintritt.

$$R = P * S \quad (1)$$

Die Eintrittswahrscheinlichkeit P hängt von zwei wesentlichen Faktoren ab. Zum einen von der Wahrscheinlichkeit P_{Raum} , dass sich der Mensch im vom Roboter überstrichenen Raum aufhält und von der Wahrscheinlichkeit P_{Ausweich} , mit der der Mensch eine drohende Kollision aktiv vermeiden kann.

Damit lässt sich das Risiko folgendermaßen abschätzen:

$$R = P_{\text{Raum}} * (1 - P_{\text{Ausweich}}) * S \quad (2)$$

Die Wahrscheinlichkeit P_{Raum} , dass Mensch und Roboter den selben Raumteil beanspruchen, hängt stark von den Bewegungsmöglichkeiten des Roboters, der Gestaltung der Arbeitszelle und den Aufgaben von Mensch und Roboter ab. Im allgemeinen ist sie unter ansonsten gleichen Bedingungen, um so größer, je raumgreifender die Bewegung des Roboters ist.

Da der Mensch eine begrenzte Reaktionsfähigkeit hat, fällt es ihm bei zunehmender Robotergeschwindigkeit schwerer, eine drohende Kollision aktiv zu vermeiden, zum Beispiel durch Ausweichen oder Betätigen eines Not-Aus-Knopfes. Aus diesem Grund ist die Wahrscheinlichkeit P_{Ausweich} um so kleiner, je größer die Robotergeschwindigkeit ist.

Die Schadenshöhe S hängt von vielen Faktoren ab. Abhängig von der maximalen Kraft des Roboters, dem Typ des Werkzeugs und den Eigenschaften der Umgebung ist die Schadenshöhe stark unterschiedlich. Im Fall eines Schadens durch Kollision, wächst die Schadenshöhe, unter sonst gleichen Bedingungen, mit der Bewegungsenergie des Roboters. Damit wird S um so größer, je größer die Robotergeschwindigkeit ist.

In der Abbildung 5 ist eine Unterteilung von Roboteroperationen anhand der Interaktion des Roboters bzw. des von ihm gehaltenen Werkstücks oder Werkzeugs mit der Umwelt dargestellt. Es wird unterschieden zwischen Roboteroperationen, bei denen sich die Kontaktsituation des Roboter-Werkstück-Systems verändert und solchen, bei denen das System in Kontakt zu weiteren Objekten der Umwelt auftritt.

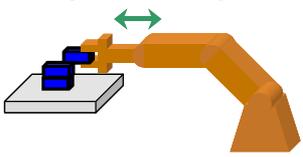
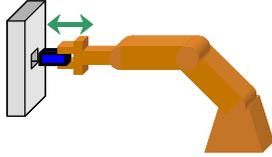
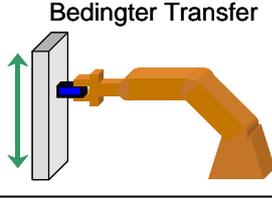
	Werkstück (zeitweise) ohne Umweltkontakt	Werkstück mit Umweltkontakt	
Bewegung mit Änderung der Kontakt-situation	Aufgreifen / Ablegen 	De-/Montieren 	Bewegung auf engem Raum
Bewegung ohne Änderung der Kontakt-situation	Freier Transfer 	Bedingter Transfer 	Bewegung über große Entfernung
	Schnelle Bewegung	Langsame Bewegung	

Abbildung 5: Unterteilung der verschiedenen Roboteroperationen nach Ausführungsgeschwindigkeit und überstrichenem Raum.

Die obigen Überlegungen führen zu dem Schluss, dass die Gefährdung des Menschen bei schnellen Bewegungen größer ist als bei langsamen und bei raumgreifenden Bewegungen größer als bei Bewegungen auf engem Raum.

Bei allen Bewegungen, außer der freien Transferbewegung, fährt der Roboter vergleichsweise langsam oder über kurze Entfernungen. Die für einen Menschen im Arbeitsraum gefährlichste Bewegung ist die freie Transferbewegung, denn dabei werden schnelle Bewegungen über größere Distanzen ausgeführt.

Die betrachtete Aufgabe besteht darin, die Transferbewegungen eines Manipulators kollisionsfrei durchzuführen. Von einer gegebenen Startkonfiguration aus, soll der Manipulator auf einer möglichst kurzen, kollisionsfreien Bahn in eine gegebene Zielkonfiguration fahren. Falls notwendig, soll der Manipulator um auftretende Hindernisse herumfahren. Wenn es keinen Weg zum Ziel gibt, soll der Manipulator zu einer freien Konfiguration in der Nähe des Ziels fahren und dort stehen bleiben. Die Anzahl und Position der Hindernisse sowie deren Bewegungsverhalten wird vorerst nicht eingeschränkt.

Im Prinzip ist eine Transferbewegung nur eine Bewegung des Roboters ohne gegriffene Objekte. Bewegungen mit gegriffenen Objekten werden als Transportbewegungen bezeichnet. Für kleine Objekte sind Transport- und Transferbewegung praktisch gleich zu behandeln. Lediglich der Transport von größeren Objekten muss gesondert betrachtet werden. Im Folgenden wird zwischen Transfer- und Transportbewegung nur dort unterschieden, wo es notwendig ist. Ansonsten wird der Begriff Transferbewegung verwendet.

1.4 Abgrenzung

Die Gewährleistung der Sicherheit für den Menschen ist Voraussetzung für ein gemeinsames Arbeiten von Roboter und Mensch in einem Arbeitsraum. Zur erfolgreichen Umsetzung von praxisnahen Aufgaben sind aber häufig noch zusätzliche Fähigkeiten des Roboters notwendig, zum Beispiel die Fähigkeit mit dem Menschen zu kooperieren oder aktiv die Umwelt zu verändern. Es ist nicht Ziel, diese Zusatzfähigkeiten zu untersuchen oder zu entwickeln.

Die Stichworte „Bildbasierte“ und „Roboterbewegungen“ tauchen auch im Bereich des Visual Servoing gemeinsam auf. Ein Beispiel hierfür ist [Noborio01]. Dabei wird die Bewegung eines Roboters mit Kameras überwacht. Ziel ist es, den Roboter so zu bewegen, dass in den Bildern bestimmte Merkmale an bestimmten Positionen zu finden sind. Im Unterschied dazu sind in der in dieser Arbeit behandelten Aufgabenstellung Vorhersagen über die Merkmale in den Bildern nicht möglich, da keine Annahmen über die Hindernisse und deren Verteilung gemacht werden. Außerdem liegt beim Visual Servoing typischerweise das Hauptaugenmerk auf ein korrektes Positionieren des Werkzeugmittelpunktes, damit Montageoperationen erfolgreich durchgeführt werden können. Bei der kollisionsfreien Transferbewegung ist aber der gesamte Roboter von Interesse.

Die Arbeit konzentriert sich auf stationäre Roboter, mobile Roboter werden nicht betrachtet.

1.5 Kapitelübersicht

Die Arbeit ist folgendermaßen aufgebaut: In Kapitel 2 wird der Stand der Forschung beschrieben. Die vorhandenen Systeme werden klassifiziert anhand der Art und Weise, wie sie eine drohende Kollision erkennen und wie sie auf diese Gefahr reagieren. In Kapitel 3 wird der prinzipielle Aufbau eines Systems zur Sicherung der Transferbewegung gezeigt. Ausgehend von einer Bewertung der möglichen Realisierungen der einzelnen Komponenten wird die gewählte Realisierung beschrieben.

Das realisierte System verwendet binäre Bilder der Szene und des Roboters. Die Erzeugung dieser Bilder ist in Kapitel 4 beschrieben. Diese binären Bilder werden vom zentralen Element des realisierten Systems, einem globalen Ganzarmkollisionstest verwendet. Dieser Kollisionstest wird in Kapitel 5 beschrieben. Im Kapitel 6 wird das verwendete Verfahren zur Suche nach einer kollisionsfreien Bahn, die Erzeugung von kontinuierlichen Roboterbewegungen und die Methoden zur Sicherheit und Wartung des Systems vorgestellt.

Im Kapitel 7 werden die Experimente mit dem realisierten System gezeigt, sowie die Leistungsfähigkeit und die Schwächen des Systems diskutiert. Im Kapitel 8 wird die Arbeit zusammengefasst und ein Ausblick auf zukünftige Arbeiten gegeben.

Kapitel 2

Stand der Forschung

2.1 Kriterien zur Klassifizierung und Bewertung der Systeme

Die bislang realisierten Systeme unterscheiden sich stark voneinander. Dies betrifft sowohl die Sensorik als auch die verwendeten Algorithmen. Als Folge davon ist es schwierig, die verschiedenen Systeme in Klassen zu unterteilen, die eine Aussage über die Leistungsfähigkeit der Systeme ermöglichen.

Eine gute Klassifizierung der Systeme sollte die folgenden Kriterien erfüllen:

1. Anordnung der Systeme nach Gemeinsamkeiten
2. Unabhängigkeit von der Realisierung (verwendete Sensoren, Umweltmodellierung)
3. Wenn möglich, eine Rangordnung nach Leistungsfähigkeit bilden.

In [Heiligensetzer02b] werden die Systeme zur Roboter-Mensch-Roboter-Interaktion anhand des Grades der Umweltmodellierung und der Kommandierbarkeit klassifiziert. Dies ist problematisch, da die Umweltmodellierung stark von den verwendeten Sensoren abhängt und nicht direkt die Leistung eines Systems beschreibt.

Für die Leistungsfähigkeit ist nur entscheidend, wie das System die Notwendigkeit einer Reaktion erkennt und wie es reagiert. Aus diesem Grund lassen sich die Systeme, unabhängig von den großen Unterschieden in den verwendeten Sensoren und Algorithmen, anhand der zu Grunde liegenden Strategien zur Erkennung von und Reaktion auf drohende Kollisionen in verschiedene Klassen unterteilen. Dies wird im folgenden ausgeführt.

2.1.1 Erkennungsstrategie

Zur Bewertung der verschiedenen Erkennungsstrategien müssen die zwei möglichen Fehlerarten hinsichtlich des entstehenden Schadens gewichtet werden. Zum einen kann eine drohende Kollision nicht als solche erkannt werden (Fehler 1. Art, Kollision), zum andern kann eine Kollision signalisiert werden, obwohl in Wirklichkeit keine Gefährdung besteht (Fehler 2. Art, Fehlalarm).

Die Erkennungsstrategie muss sicherstellen, dass das System drohende Kollisionen als solche erkennt, da der Schaden einer Kollision sehr groß sein kann. Gleichzeitig soll a-

ber auch die Zahl der „Fehlalarme“ möglichst gering sein. Jede erkannte Kollision löst die verwendete Reaktionsstrategie aus. Im allgemeinen ist die Ausführung dieser Reaktionsstrategie aufwändiger als das einfache Weiterfahren. Damit beeinträchtigt jeder Fehlalarm die Arbeitsleistung des Systems.

Zur Erkennung einer drohenden Kollision werden verschiedene Verfahren verwendet, wie Abstandsberechnung und Kollisionstest jeweils mit und ohne eine Projektion der Bewegung der erkannten Hindernisse und des Roboters in die Zukunft. Entscheidend für die Leistungsfähigkeit der Erkennung ist, welche Punkte des Roboters auf Kollision mit Hindernissen untersucht werden.² Anhand dieses Merkmals lassen sich die Erkennungsstrategien unterteilen in Strategien:

1. Schutzraumüberwachung (*SÜ*)
2. Einzelpunkterkennung (*EE*)
3. Mehrpunkterkennung (*ME*)
4. Ganzarmerkennung (*GE*)

Schutzraumüberwachung

Bei Systemen mit Schutzraumüberwachung (Abkürzung *SÜ*) wird kein Punkt des Roboters, sondern der vom Roboter (möglicherweise) überstrichene Arbeitsraum überwacht. Da sich der Roboter innerhalb dieses Raumes bewegt, müsste eine Erkennung eigentlich zwischen Roboter und Hindernissen unterscheiden können. Um den damit verbundenen hohen Aufwand zu vermeiden, werden bei manchen Systemen nur die Grenzen des Schutzraumes überwacht. Zur Überwachung der Grenzen können vergleichsweise einfache Sensoren wie Lichtschranken verwendet werden.

Der überwachte Schutzraum kann je nach Erfordernis statisch sein oder dynamisch verändert werden. Im statischen Fall übernimmt der sensorüberwachte Schutzraum die Funktion einer abgetrennten Zelle. Im Falle des dynamisch veränderten Schutzraumes genügt allerdings eine reine Überwachung der Grenzen auf Überschreitung nicht, sondern der gesamte Raum muss überwacht werden.

Einzelpunkterkennung

Bei Systemen mit Einzelpunkterkennung (Abkürzung *EE*) wird nur ein einzelner Punkt des Roboters überprüft. In der Regel ist dies der Werkzeugmittelpunkt (Tool-Center-Point, TCP).

Mehrpunkterkennung

Bei Systemen mit Mehrpunkterkennung (Abkürzung *ME*) werden mehrere Punkte des Roboters überprüft. Häufig werden die Eckpunkte der Segmente des Roboters genommen. Die Idee ist, dass diese Punkte am kollisionsgefährdetsten sind. Auch ein Mensch

² Im Prinzip gilt dies auch für die Hindernisse. Allerdings sind bei der sensorbasierten Hindernisvermeidung die Hindernisse prinzipiell nicht vorher bekannt und müssen mit den Sensoren erfasst werden. Keines der bekannten Systeme reduziert die erfassten Sensordaten derart, dass die reduzierte Repräsentation die gemessenen Hindernisse nicht mehr vollständig umfasst. In bezug auf die Hindernisse ist daher eher die Frage relevant, welche Objekte als Hindernisse erkannt werden können. Dies ist jedoch stark von den gewählten Sensoren abhängig und damit ein Widerspruch zur Forderung 2 nach den Kriterien einer Klassifikation der Systeme.

kollidiert häufiger mit dem Ellbogen oder der Schulter an ein Hindernis, als mit der Mitte des Oberarms.

Ganzarmerkennung

Bei der Ganzarmerkennung (Abkürzung *GE*) wird das gesamte Volumen des Roboters auf Kollision mit der Umwelt überprüft. Auf diese Weise ist sichergestellt, dass keine Kollisionsgefahr unerkannt bleibt. Wenn hinreichend viele Kontrollpunkte auf dem Roboter verteilt werden, geht die Mehrpunkterkennung in die Ganzarmerkennung über.

Bei einigen Systemen wird mit einem einzigen Sensor ein größerer Roboterbereich auf Kollision untersucht. Diese Erkennung ist, je nachdem ob der gesamte Roboter oder nur ein Teil überwacht wird, äquivalent zu einer Ganzarm- oder Mehrpunkterkennung.

Bewertung der Erkennungsstrategien

Die Schutzraumüberwachung erkennt bei geeigneter Dimensionierung des Schutzraumes jede mögliche Kollision. Allerdings ist bei entsprechender Dimensionierung die Zahl der Fehlalarme sehr groß. Die Schutzraumstrategie ermöglicht nicht, dass sich Mensch und Roboter im selben Raumbereich aufhalten. Daher ist diese Strategie für die Mensch-Roboter-Koexistenz alleine nicht ausreichend. Die Schutzraumstrategie kann dazu genutzt werden, kleine Raumbereiche zu überwachen, in denen Mensch und Roboter nie gleichzeitig sein können, wie zum Beispiel enge Durchgänge.

Die Einzelpunkterkennung hat zwar ein deutlich verringertes Risiko für Fehlalarme, aber die Zahl der Kollisionen, die nicht durch die Erkennung verhindert werden, ist unter Umständen sehr groß. Der Nachteil dieser Strategie ist, dass der Roboter mit andern Teilen, zum Beispiel dem „Ellbogen“, mit der Umwelt kollidieren kann. Unter speziellen Bedingungen kann es jedoch genügen, einen einzigen Punkt des Roboters zu überwachen.

Bei der Mehrpunkterkennung ist das Risiko für nicht erkannte Kollisionen kleiner als bei der Einzelpunkterkennung. Trotzdem können natürlich auch die unüberwachten Roboterpunkte in ungünstigen Fällen mit Hindernissen kollidieren.

Die Ganzarmerkennung bietet die optimale Kombination aus Erkennungssicherheit und geringer Zahl der Fehlalarme. Der Nachteil dabei ist, dass das Überprüfen des ganzen Roboters sehr zeit- und rechenaufwendig sein kann.

Insgesamt können die Erkennungsstrategien bei der Mensch-Roboter-Koexistenz nach zunehmender Leistungsfähigkeit von Schutzraumüberwachung über Einzelpunkterkennung und Mehrpunkterkennung zur Ganzarmerkennung geordnet werden.

2.1.2 Reaktionsstrategie

Die Reaktionsstrategie muss die Verletzung des Menschen und die Beschädigung des Roboters vermeiden. Gleichzeitig muss sie sicherstellen, dass Mensch und Roboter so effizient wie möglich arbeiten.

Die Reaktionsstrategien lassen sich unterteilen in:

1. Kollisionsentschärfung (*KE*)
2. Geschwindigkeitsregelung (*GR*)
3. Lokale Ausweichbewegung (*LA*)
4. Globale Ausweichbewegung (*GA*)

Kollisionsentschärfung

Die Strategie der Kollisionsentschärfung (Abkürzung *KE*) ist, dafür zu sorgen, dass Kollisionen für den Menschen ungefährlich sind. Dies kann durch eine Begrenzung der kinetischen Energie, der aufgewendeten Kraft und durch die Nachgiebigkeit des Roboters erreicht werden.

Die Begrenzung der im Robotersystem gespeicherten kinetischen Energie erfolgt durch eine Begrenzung der Robotergeschwindigkeit. Diese Begrenzung basiert auf einem dynamischen Modell des Roboters. Die Begrenzung der aufgewendeten Kraft kann durch Kraftsensoren, Messen der Motorströme oder durch mechanische Beschränkungen der Motoren und Getriebe geschehen. Die Nachgiebigkeit des Roboters kann durch deformierbare Segmente, nachgiebige Motoren und Getriebe oder durch zusätzliche nachgiebige Elemente erreicht werden.

Geschwindigkeitsregelung

Die Strategie der Geschwindigkeitsregelung (Abkürzung *GR*) besteht darin, dass der Roboter die Geschwindigkeit, mit der er auf der vorgegebenen Bahn fährt, je nach Abstand zu den Hindernissen ändert. Wird die Entfernung zu klein, wird die Bewegung gestoppt, bis die Gefahr beseitigt ist. Die einfachste Form der Geschwindigkeitsregelung ist binäres Anhalten / Weiterfahren.

Diese einfachste Form der Geschwindigkeitsregelung ergibt zusammen mit der Schutzraumüberwachung die klassische Sicherheitsregelung von abgetrennten Industrieroboterzellen, bei der der Roboter außer Betrieb gesetzt wird, sobald ein Eintritt in den Schutzraum festgestellt wird.

Globale- und lokale Ausweichbewegung

Ein System, das die Reaktionsstrategie der Ausweichbewegung verfolgt, erkennt die Gefahr der Kollision durch seine Sensoren und verändert die Roboterbewegung so, dass keine Kollisionsgefahr mehr besteht. Im Fall, dass das Ziel der Bewegung nicht erreicht werden kann, muss angehalten werden. Diese Strategie gibt es in zwei Ausprägungen, der lokal geplanten und der global geplanten Ausweichbewegung.

Bei der lokal geplanten Ausweichbewegung (Abkürzung *LA*) wird nur die Information aus der unmittelbaren Umgebung des Roboters verwendet.

Bei der global geplanten Ausweichbewegung (Abkürzung *GA*) wird der gesamte Raum betrachtet.

Eine Mischform zwischen der Strategie der Geschwindigkeitsregelung und der lokalen Ausweichbewegung ist die komponentenweise Geschwindigkeitsregelung. Dabei wird die Geschwindigkeit des Roboters in mehrere unabhängige Komponenten zerlegt und nur der Betrag einer oder mehrerer dieser Komponenten geregelt. Beispielsweise ermöglicht die Regelung derjenigen Komponente der Robotergeschwindigkeit, die direkt auf das nächstgelegene Hindernis zeigt, ein Gleiten entlang dieses Hindernisses. Da die ursprünglich geplante Bahn bei der komponentenweisen Geschwindigkeitsregelung verlassen wird, fällt sie in die Gruppe der lokal geplanten Ausweichbewegungen.

Bewertung der Reaktionsstrategien

Die Strategie der Kollisionsentschärfung *KE* hat den Nachteil, dass keine herkömmlichen Industrieroboter verwendet werden können, beziehungsweise diese verändert wer-

den müssen. Es gibt Probleme beim Bewegen großer Lasten, da diese die bei einer Kollision auftretenden Kräfte stark beeinflussen. Zudem ist die maximal zulässige Fahrgeschwindigkeit eng begrenzt. Durch die Nachgiebigkeit lassen sich Positionen nur ungenau ansteuern. Die Positionsungenauigkeit kann zwar durch eine sensorbasierte Regelung, zum Beispiel Visual Servoing, ausgeglichen werden, aber dadurch besteht die Gefahr, dass gegen ein kollidierendes Objekt geregelt wird. Dadurch geht die Nachgiebigkeit wieder verloren und bei entsprechender Stärke des Roboters kann ein Mensch verletzt werden. Letztendlich gehen bei der Kollisionsentschärfung die Stärken des klassischen Industrieroboters, nämlich hohe Geschwindigkeit und Präzision, verloren. Das Hauptproblem ist aber, dass die Kollision nicht vermieden wird. Je nach Werkzeug (z.B. Schneidbrenner, Skalpell) können schon leichte Kollisionen schwere Verletzungen des Menschen bewirken. Die Strategie der Kollisionsentschärfung ist daher für eine sichere Mensch-Roboter-Koexistenz nicht ausreichend.

Die Strategie der Geschwindigkeitsregelung *GR* hat den Vorteil, dass der Roboter sich immer auf einer bekannten Bahn bewegt. Dadurch ist keine aufwendige Online-Bahnplanung notwendig. Die Sensorik muss lediglich erkennen, ob in Fahrtrichtung des Roboters ein Hindernis vorhanden ist. Dies kann kostengünstig mit auf dem Roboterarm angebrachten Sensoren realisiert werden. Der Nachteil der Geschwindigkeitsregelung ist, dass der Roboter immer dann anhält, sobald ein Hindernis auf der vorberechneten Bahn ist. Auch wenn die Zielposition frei ist und ein Weg dorthin existiert, wird die Arbeit des Roboters unterbrochen, bis das Hindernis beseitigt ist. Dies bedeutet, dass bei der Mensch-Roboter-Koexistenz allein dem Menschen aufgebürdet wird, sich auf die Bedürfnisse des Roboters einzustellen und ihm den Weg frei zu räumen.

Die Strategie der lokal geplanten Ausweichbewegung *LA* hat den Vorteil, dass der Roboter einfache Hindernisse auf seinem Weg umfahren kann. Selbst einfache Ausweichbewegungen, wie die komponentenweise Geschwindigkeitsregelung, reduzieren die Standzeiten des Roboters im Vergleich zur Geschwindigkeitsregelung. Nachteilig ist, dass der Roboter immer noch in Sackgassen stehen bleiben kann, auch wenn es einen Weg zum Ziel gibt.

Die Strategie der global geplanten Ausweichbewegung *GA* hat den Vorteil, dass der Roboter immer zum Ziel kommt, wenn es einen Weg gibt. Dadurch wird der Roboter maximal effizient eingesetzt. Der Nachteil dieser Strategie ist, dass die Sensorik den gesamten Bewegungsraum des Roboters abdecken muss und die Bahnplanung eine erhebliche Rechenkapazität beansprucht.

Es gibt eine Hierarchie der Leistungsfähigkeit der verschiedenen Reaktionsstrategien. Am leistungsfähigsten ist die global geplante Ausweichbewegung, da sie sicherstellt, dass keine Kollision auftritt und gleichzeitig der Roboter immer dann weiterfahren kann, wenn ein Weg existiert. Bei der lokal geplanten Ausweichbewegung kann es sein, dass ein Weg existiert, aber nicht gefunden wird und der Roboter daher überflüssige Wartezeiten hat. Noch weniger leistungsfähig ist die Geschwindigkeitsregelung, da der Roboter immer dann aufhört zu arbeiten, sobald der fest vorgegebene Weg durch ein Hindernis blockiert ist. Die Leistungsfähigkeit der Kollisionsentschärfung hängt wesentlich davon ab, wie das System nach einer Kollision reagiert. Im einfachsten Fall, wird ein Not-Halt ausgelöst. Dadurch kann das System erst weiterarbeiten, wenn die Kollision von Hand beseitigt und das System neu gestartet wurde. In diesem Fall ist die Kollisionsentschärfung am wenigsten leistungsfähig.

Es ist möglich, in einem System alle Reaktionsstrategien gleichzeitig zu realisieren. Allerdings ist dies nicht immer sinnvoll, da nicht nur die Stärken, sondern auch die Schwächen kombiniert werden. Ein System mit Kollisionsentschärfung leidet unter der

reduzierten Geschwindigkeit oder Positionsgenauigkeit auch dann, wenn eine global geplante Ausweichbewegung eine Kollision unwahrscheinlich macht.

2.2 Klassifizierung der existierenden Systeme

Anhand der Kombination von Erkennungs- und Reaktionsstrategie lassen sich die bislang realisierten Systeme in 16 Klassen einordnen. Dabei können verschiedene Strategien gleichzeitig realisiert werden. Systeme, die mehrere Strategien kombinieren, werden in allen zutreffenden Klassen aufgeführt. Andererseits kann es sein, dass die betrachtete Aufgabenstellung abweicht. Zum Beispiel nehmen einige Autoren eine statische Hindernisumgebung an. Bei diesen Systemen wird aus den beschriebenen Eigenschaften abgeleitet, welche Klasse technisch möglich wäre.

Im folgenden werden einige der bislang realisierten Systeme vorgestellt. Die Systeme sind nach der verwendeten Reaktionsstrategie zusammengefasst.

In der Regel werden bei der Kollisionsentschärfung die Kräfte bzw. Momente an den Gelenken des Roboters gemessen und keine externen Sensoren verwendet. Dies macht die Klassifizierung der zugehörigen Erkennungsstrategie problematisch, da sich die Systeme gerade dadurch auszeichnen, dass kein Punkt des Roboters auf Kollision mit der Umwelt getestet wird. Daher wird für die Kollisionsentschärfung eine andere Einordnung verwendet. Wenn die Kollisionsentschärfung den gesamten Roboter abdeckt, wird das System in die Kategorie Ganzarmerkennung eingeordnet. Deckt sie nur einen Teil des Roboters ab, dann erfolgt die Einordnung in die Kategorie Mehrpunkterkennung. Wird nur das Werkzeug entschärft, folgt die Einordnung in die Klasse Einzelpunkterkennung.

2.2.1 Systeme mit Kollisionsentschärfung

In [Heinzmann99] und [Zelinsky99] wird ein System präsentiert, das eine Kollisionsentschärfung realisiert. Die Entschärfung der Kollision wird durch einen leichten Arm, die Verwendung von Getrieben, die nur begrenzte Momente übertragen, und eine Begrenzung der Einschlagskraft erreicht. Die Begrenzung der Einschlagskraft erfolgt durch das Berechnen eines dynamischen Modells. Damit auch schwerere Gegenstände transportiert werden können, wird die Einwirkung der Schwerkraft kompensiert. Das System verwendet die Motorströme als Kraftsensoren. Dieser Teil des Systems fällt in die Kategorie *GE/KE*. Gleichzeitig wird die kartesische Geschwindigkeit von Ellbogen und Hand des Roboters überwacht. Diese Überwachung basiert auf den Drehgeschwindigkeiten der Gelenke, die von analogen Schaltkreisen überwacht werden. Wenn die Geschwindigkeit zu groß ist, wird der Roboter abgeschaltet. Dieser Teil des Systems fällt in die Kategorie *ME/GR*.

In [Lim00] wird ein Ansatz präsentiert, bei dem ein starrer Roboter durch eine nachgiebige und bewegliche Basis ergänzt wird. Der prinzipielle Aufbau und das Versuchsmo-
dell sind in Abbildung 6 zu sehen. Es werden verschiedene Szenarien in Simulation und Experiment untersucht, bei denen der Roboter mit Hindernissen und Menschen kollidiert. Die dabei auftretenden Kräfte wurden berechnet bzw. gemessen. Das System verfügt über zwei Kraft-Momenten-Sensoren. Einer sitzt in der Basis des Roboters, der andere ist mit einer Kontaktfläche auf dem Manipulator verbunden. Der Kraft-Momenten-Sensor in der Basis wird allerdings nur verwendet, um eine passiv verformbare Basis und eine passiv bewegliche Plattform zu realisieren und wird nicht zur eigentlichen Kollisionserkennung verwendet. Durch eine spezielle Regelung, die in [Lim99] vorgestellt

wurde, versucht das System einerseits die Kontaktkräfte in einem niedrigen Bereich und gleichzeitig den Endeffektor des Roboters auf der gewünschten Bahn zu behalten. Dies funktioniert allerdings nur bei redundanten Robotern. Der Kraftsensor in der Kontaktfläche wird nur zur Messung der Kontaktkraft benutzt, es findet keine Reaktion auf einen steigenden Kraftverlauf statt. Dieses System fällt in die Kategorie *GE/KE*.

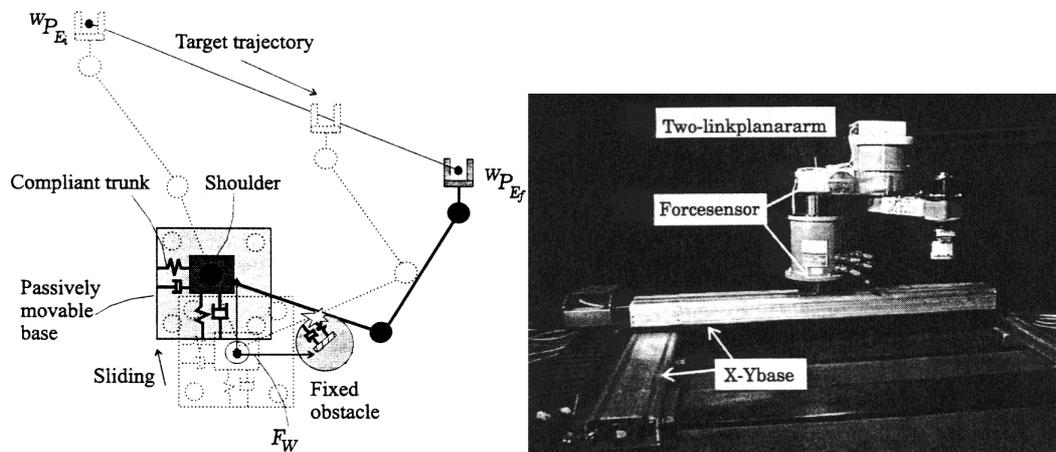


Abbildung 6: Aufbau und Versuchsmodell des „Passively Movable Human-Friendly Robots with Passive viscoelastic Trunk“ (aus [Lim00])

2.2.2 Systeme mit Geschwindigkeitsregelung

In [Meisel91], [Meisel94] und [Ameling96] wird ein System zur Hinderniserkennung vorgestellt. Dieses System verwendet mehrere stationäre CCD-Kameras, um ein passives Multi-Lichtschrankenverfahren zu realisieren. Dabei werden die Abbildungsstrahlen, die vordefinierte Raumpunkte kreuzen, auf Veränderungen überprüft. In Abbildung 7 ist die mögliche Anwendung auf einen Portalroboter mit eingezeichneten Positionen von Kameras und Überwachungsraumpunkten dargestellt. In der Einrichtungphase werden zuerst die Kameras bezüglich des Weltkoordinatensystems kalibriert, dann die zu überwachenden Raumpunkte festgelegt und deren Bildpunkte in den Kameras bestimmt. Im Betrieb werden die Ist-Merkmale der Bildpunkte mit den Merkmalen im freien Zustand verglichen. Wenn die Abweichung einen festgelegten Schwellenwert überschreitet, wird der zugehörige Abbildungsstrahl als unterbrochen angenommen.

Das System wurde nicht vollständig realisiert. Eine Anwendungsmöglichkeit ist die Raumüberwachung mit Nothalt des Roboters *SÜ/GR*. Kurz angesprochen wurde die Möglichkeit der hindernisvermeidenden Bahnplanung für den Werkzeugmittelpunkt. Je nach Bahnplanungsverfahren läge dann die Klasse *EE/LA* oder *EE/GA* vor. Bei allen Einsatzzwecken müsste das Problem des in den Erfassungsbereich hineinragenden Roboters gelöst werden. Dies wird jedoch nicht erwähnt. Dieses Problem tritt bei der Schutzraumüberwachung nur in einfacherer Form auf, da dabei angenommen werden kann, dass der Roboter das einzige Objekt innerhalb des Bereiches ist und alle Hindernisse von außen in den überwachten Bereich eindringen. Aus diesem Grund wird dieses System in die Kategorie *SÜ/GR* eingeordnet.

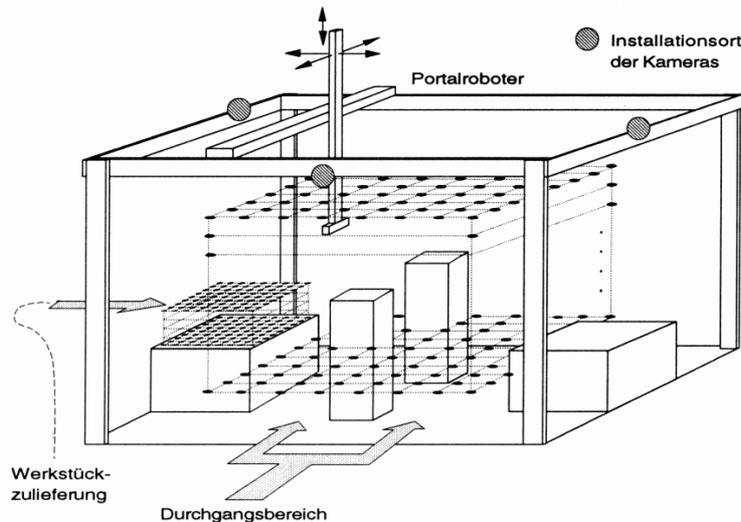


Abbildung 7: System zur Überwachung von vordefinierten Raumpunkten (aus [Meisel94])

In [Zettl02] wird ein System vorgestellt, bei dem zwei Industrieroboter Teppichböden schneiden. Dieses System ist in Abbildung 8 zu sehen. Zwei Einlegestationen werden im Wechsel von Menschen mit neuem Material bestückt und von den Robotern bearbeitet. Beide Stationen sind mit 2D-Laserscannern ausgerüstet, die einen Eintritt feststellen können, allerdings wird nur die gerade von den Robotern benutzte Station überwacht. Wenn der Schutzraum dieser Station verletzt wird, werden die Roboter angehalten. Damit gehört dieses System in die Klasse $SÜ/GR$.

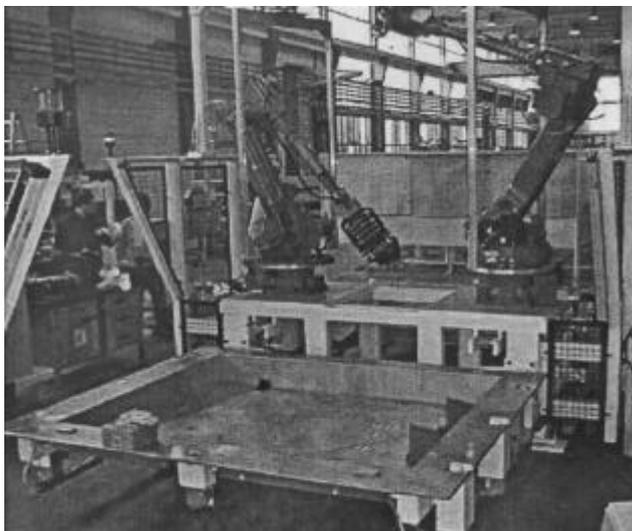


Abbildung 8: System zum Wasserstrahlschneiden von Teppichböden mit sensorüberwachter Einlegestation (aus [Zettl02])

In [Morhard02] wird ein System vorgestellt, das einem Menschen gestattet, sich im Normalbetrieb innerhalb der Roboterzelle aufzuhalten. Eine Anwendung des Systems besteht in der Prozessüberwachung von Schweißarbeiten. Hierzu wird die Grenze des Arbeitsraumes durch Lichtschranken überwacht. Das Betreten und Verlassen des Arbeitsraumes erfordert eine Anmelde- bzw. Abmeldeprozedur. Während sich der Mensch im Arbeitsraum aufhält, ist die kartesische Bewegungsgeschwindigkeit des Roboters beschränkt. Dieser Teil des Systems fällt in die Kategorie GE/KE . Bei unangemeldetem Eintreten oder Verlassen des Schutzraumes wird der Roboter stillgelegt. Damit fällt dieser Teil des Systems in die Kategorie $SÜ/GR$.

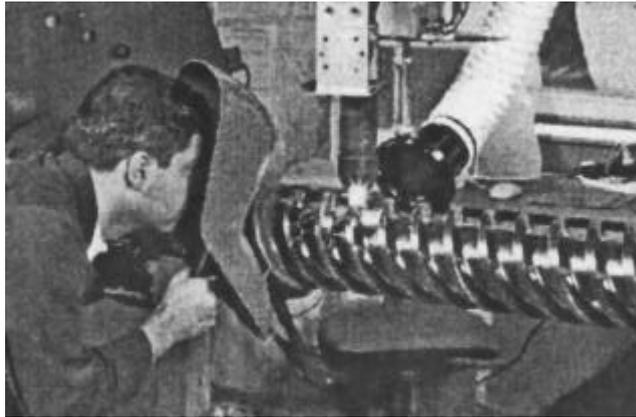


Abbildung 9: System zur Prozessüberwachung von Schweißarbeiten durch den Menschen (aus [Morhard02])

In [Thiemermann02] und [Spiengler02] wird ein System präsentiert, das die gemeinsame Montage von Objekten durch Mensch und Roboter ermöglicht. Das System ist im linken Teil von Abbildung 10 zu sehen. Über dem gemeinsamen Arbeitsplatz sind Kameras angebracht. Die Geschwindigkeit des Roboters wird über den minimalen Abstand zwischen Roboter und Mensch geregelt. In diese Regelung können auch Prozesszustände eingehen. Durch die Bildverarbeitung werden die Hände eines Menschen anhand der charakteristischen Farbkomponenten erkannt. Welche und wie viele Punkte des Roboters in die Abstandsberechnung eingehen, ist unbekannt. Die auffälligen Markierungen an der Oberseite des Armes legen die Vermutung nahe, dass diese Punkte zur Erfassung des Roboters dienen. In diesem Fall würden Kollisionen durch das Armsegment vermieden und das System fiel in die Klasse *ME/GR*.



Abbildung 10: Modellarbeitsplatz zur gemeinsamen Montage von Objekten durch menschlichen Arbeiter und Roboter (links, Foto: Autor)
Sicht von oben mit exponierten Punkten von Mensch und Roboter und minimalem Abstand (rechts, aus [Thiemermann02])

In [Heiligensetzer02a] wird ein Kleinroboter vorgestellt, der ohne zusätzliche Schutz-einrichtungen arbeiten soll. Dazu wird der Roboter mit einem kapazitiven Sensor in der Nähe des Werkzeuges ausgestattet. Das vorderste Armsegment wird mit einer berührungsempfindlichen Matte verkleidet. Die Motorströme werden als Kraftsensoren genutzt. Zusätzlich wird die kinetische Energie des Roboters beschränkt. Die Beschränkung erfolgt durch die Mitführung eines dynamischen Modells des Roboters und der Regelung der Geschwindigkeit, so dass die Bewegungsenergie des gesamten Roboters innerhalb gegebener Grenzwerte bleibt. Daher fällt dieser Teil des Systems in die Klasse *GE/KE*. Der kapazitive Sensor schützt im wesentlichen den Werkzeugbereich. Damit wird durch einen einzelnen Sensor eine Mehrpunkterkennung *ME* realisiert. Ebenso liefert die berührungsempfindliche Matte nur ein Ausgangssignal, das aber den Kontaktzu-

stand des ganzen Armsegmentes beschreibt. Auch damit wird eine Mehrpunkterkennung *ME* realisiert. Über die Reaktionsstrategie wird keine Aussage getroffen. Das Problem ist, dass sowohl der kapazitive Sensor als auch die berührungsempfindliche Matte nur die Präsenz eines Objektes anzeigen kann. Es ist keine Information über den Aufenthaltsort verfügbar. Damit ist die Planung von Ausweichbewegungen nicht möglich.³ Eine Geschwindigkeitsregelung ist jedoch realisierbar. Aus diesem Grund wird dieser Teil des Systems in die Klasse *ME/GR* eingeordnet.

2.2.3 Systeme mit lokaler Ausweichbewegung

In [Yu99] wird ein System vorgestellt, bei dem ein Laserscanner am Flansch des Roboters montiert ist. Das Versuchssystem ist in Abbildung 11 zu sehen. Dadurch, dass der Sensor an der Hand des Roboters montiert ist, kann er in verschiedene Richtungen sehen. Durch Aufnahmen aus verschiedenen Blickrichtungen kann der einzelne, drehbare Sensor im praktischen Betrieb mehrere Sensoren ersetzen. Dies gilt aber nur dann, wenn die Zeit, die ein Scan braucht, unwichtig ist. Dies trifft bei statischen Hindernissen zu, nicht aber bei dynamischen. Die Entfernungsdaten werden zum Aufbau eines Graphen der freien Konfigurationen (Roadmap) benutzt. Der Graph wird iterativ aufgebaut, bis Start und Ziel verbunden sind. Dies realisiert eine globale Ausweichbewegung. Allerdings gilt dies nur für den Fall von statischen Hindernissen. Bei dynamischen Hindernissen kann der Graph nicht aufgebaut werden, da die Hindernisse zwischen den einzelnen Scans ihre Position verändern können. Damit ist maximal noch eine lokale Ausweichbewegungsplanung *LA* möglich. Das System aus [Yu99] gehört daher in die Kategorie *EE/LA*.

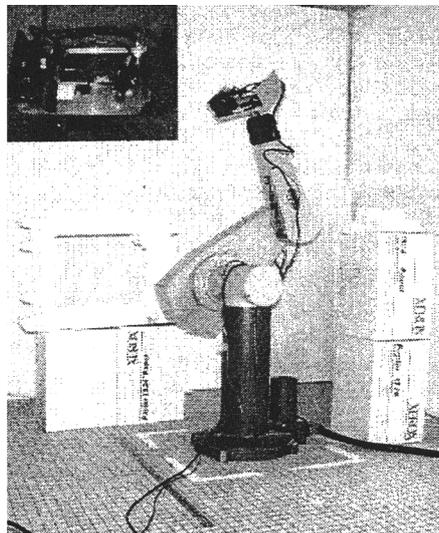


Abbildung 11: Versuchsmo­dell des Eye-in-Hand-Systems, der Sensor ist in der oberen linken Ecke vergrößert dargestellt (aus[Yu99])

³ Unter der Annahme von statischen Hindernissen ist die Planung von Ausweichbewegungen möglich. Beim Auslösen eines Sensors wird der Roboter solange in die Gegenrichtung verfahren, bis der Sensor wieder Freiraum anzeigt. Der Ort des Hindernisses wird in einer Karte eingezeichnet. Anschließend wird eine Bewegung zum Ziel unter der Berücksichtigung aller bekannten Hindernisse geplant. Dies wird solange fortgesetzt, bis das Ziel erreicht ist oder fest steht, dass kein freier Weg zum Ziel existiert. Dieser Ansatz funktioniert nicht mit beweglichen Hindernissen, da keine vertrauenswürdige Karte aufgebaut werden kann.

In [Stettmer94] und [Ameling96] wird ein Portalrobotersystem vorgestellt, dessen Bewegungen durch einen 2D-Laserscanner abgesichert sind. Der Laserscanner kann so gedreht werden, dass er immer in Fahrtrichtung des Arms zeigt. Den strukturellen Aufbau dieses Systems und ein gemessenes Höhenprofil der Arbeitszelle zeigt Abbildung 12. Mit Hilfe dieses Systems wird der Arbeitsraum während der Roboterbewegung erfasst. Falls ein Hindernis erkannt wird, wird der Roboter gestoppt. Anschließend wird der Bereich des Hindernisses bei stehendem Roboter erfasst und mit der Planung einer Ausweichbahn begonnen. Hauptsächlich wird die Kollision durch ein Verfahren in negativer Z-Richtung (Anheben des Armes) vermieden. Diese lokal geplante Ausweichbewegung setzt Freiraum über dem Roboter voraus. Ein anderes Verfahren zur Erzeugung lokaler Ausweichbewegungen umfährt die Hindernisse seitlich. Es wird davon ausgegangen, dass nur das Z-Glied des Portalroboters mit der Umwelt kollidieren kann. Durch die Bauform des Roboters und die Art der Scannermontage wird der in Fahrtrichtung zeigende und unterhalb des Scanners liegende Teil des Z-Gliedes abgesichert. Aus diesem Grund fällt dieses System in die Kategorie *ME/LA*.

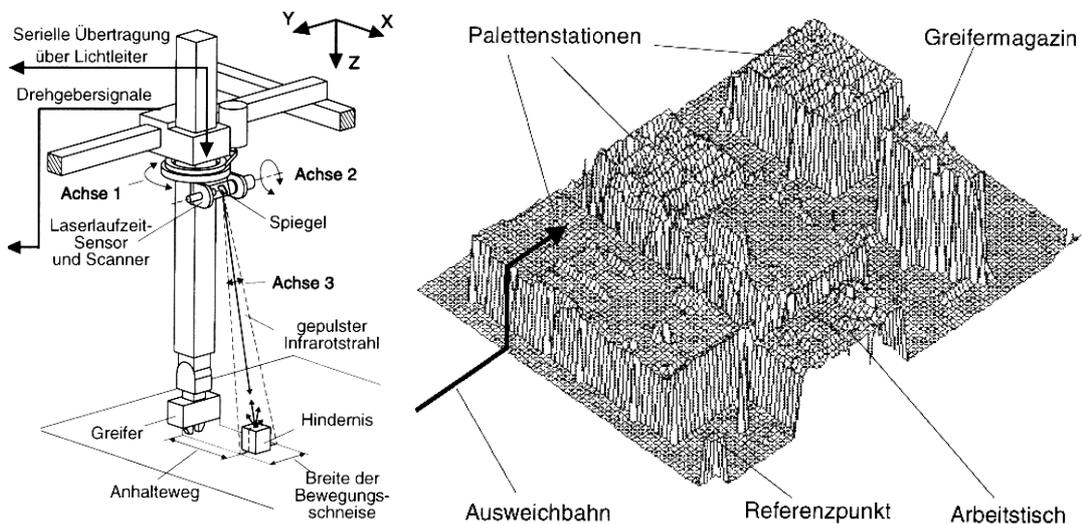


Abbildung 12: System zur Kollisionsvermeidung für einen Portalroboters (links) Scan mit Objekten und Ausweichbahn (rechts) (aus [Stettmer94])

In [Bischoff99] wird ein System vorgestellt, das bei einer linearen Bewegung des Roboters online Hindernissen ausweicht. Die Versuchszelle ist in Abbildung 13 zu sehen. Zwei Graubildkameras sind an einem Gestell über der Roboterzelle angeordnet und blicken nach unten. Sowohl der Boden der Zelle als auch der Roboter sind dunkel angestrichen. Die verwendeten Hindernisse sind weiß. Das dem Roboter in Bewegungsrichtung nächste Hindernis wird als kritisch bewertet und nur dieses wird betrachtet. Die Koordinaten von TCP und Hindernis werden in zylindrische Koordinaten umgerechnet und fuzzifiziert. Die Ausweichbewegung wird über linguistische Variablen für den Winkel und den Abstand von Hindernis und Roboter ermittelt. Wenn keine Bewegung möglich ist, bleibt der Roboter stehen, bis das Hindernis weg ist. In der Ausweichbewegung wird nur eine 2D-Regelung verwendet, die Ausweichhöhe ist fest vorgegeben. Es wird nur der TCP des Roboters betrachtet. Das System fällt in die Kategorie *EE/LA*.



Abbildung 13: Roboterzelle mit stationären Kameras zur online Kollisionsvermeidung (aus [Bischoff99])

Systeme mit künstlichen Häuten

Bei den Systemen mit lokaler Ausweichbewegung gibt es eine Untergruppe, die als auffällige Gemeinsamkeit die Verwendung von vielen, auf dem Roboterarm montierten Einzelsensoren aufweist. Diese Art der Montage wird als künstliche Haut bezeichnet. Hinter der Idee der künstlichen Haut steht der Versuch, den menschlichen Tastsinn nachzubilden.

Je nach verwendetem Sensortyp haben diese Ansätze unterschiedliche Einsatzgebiete. Allen gemeinsam ist, dass eine große Anzahl von relativ einfachen Sensoren an dem Roboterarm befestigt wird. Im Prinzip kann eine künstliche Haut mit einem beliebigen Sensortyp oder einer Mischung aus verschiedenen Sensortypen bestehen. In der Praxis werden vor allem solche Sensoren verwendet, die in einer kleinen Bauform und kostengünstig verfügbar sind.

Theoretisch gehören die Systeme mit künstlichen Häuten in die Klasse Mehrpunkterkennung, da jeder Sensor nur einen begrenzten Raumbereich überwacht. Allerdings ist es möglich, die Sensoren so anzuordnen, dass der gesamte Roboterarm überwacht wird. Diese Systeme werden daher in die Klasse der Ganzarmerkennung eingeordnet.

Die Verwendung von kapazitiven Sensoren zur Hinderniserkennung und Hindernisvermeidung bei Industrierobotern wird von [Novak92a], [Novak92b] und [Feddema94] beschrieben. Das vorgestellte System ist in Abbildung 14 zu sehen. Das Oberarmglied des verwendeten Roboters ist vollständig mit kapazitiven Sensoren überzogen. Beim Unterarmglied sind nur die Teile mit Sensoren ausgestattet, die bei der Bewegung des Roboters nicht mit dem Oberarmglied kollidieren können. Die Sensoren sind so angeordnet, dass die überwachten Bereiche überlappen. Bei einer Annäherung an Hindernisse wird die Geschwindigkeits-Komponente in Richtung des nächsten Hindernisses reduziert. Bei zu großer Annäherung bleibt der Roboter stehen. Da bei einer komponentenweisen Geschwindigkeitsregelung die vorbestimmte Bahn verlassen wird und die vorhandenen Sensoren nur einen Teil des Roboterarms abdecken, fällt dieses System in die Kategorie *ME/LA*.

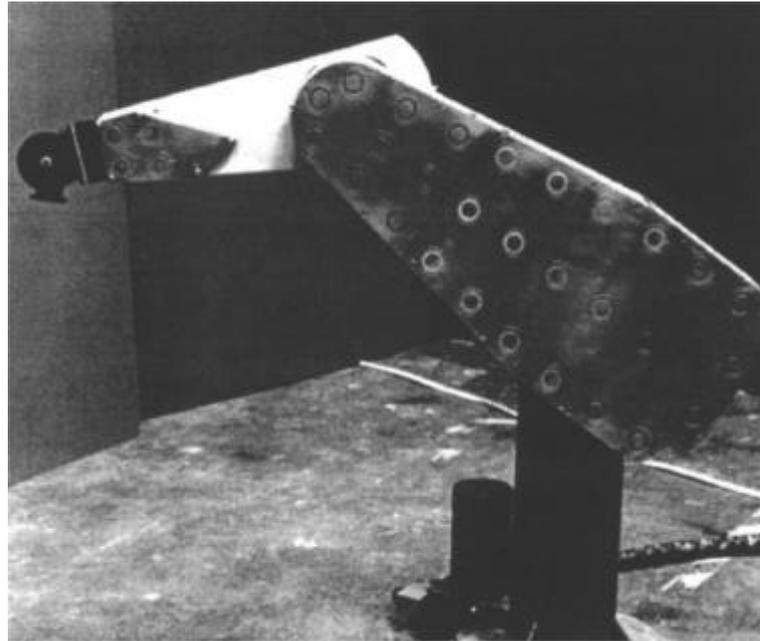


Abbildung 14: Roboter mit Sensorhaut aus kapazitiven Abstandssensoren (aus [Novak92a])

Die Firma Merritt Systems⁴ hat für die amerikanische Weltraumbehörde NASA ein modulares System entwickelt, das den kinematisch redundanten Roboterarm des Space-shuttles mit einer Haut aus vielen Einzelsensoren ausrüstet. Ein solcher Arm ist in Abbildung 15 zu sehen. Zur Verfügung stehen Ultraschallsensoren, Reflexlichttaster und kapazitive Sensoren. Das Versuchssystem an einem SCARA Roboter mit einer Sensorhaut aus Infrarotabstandssensoren ist in [Wegerif92] beschrieben. Das Steuerungssystem besitzt verschiedene Ausweichmodi. Solange wie möglich wird der Endeffektor unter Ausnutzung der kinematischen Redundanz auf der vorberechneten Bahn gehalten. Erst wenn dies nicht mehr möglich ist, wird aus den lokalen Sensorinformationen eine Ausweichbewegung geplant. Wenn der Roboterarm vollständig mit Sensoren überzogen ist, gehört dieses System in die Kategorie *GE/LA*.

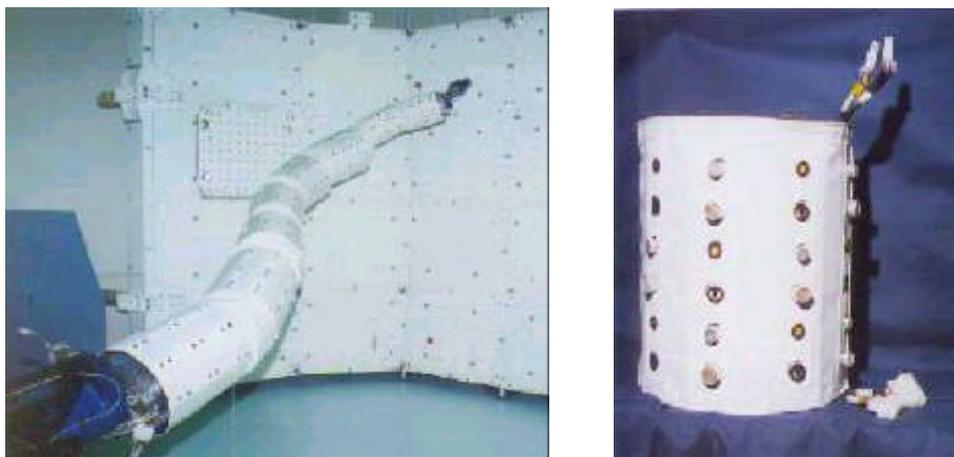


Abbildung 15: Redundanter Roboterarm mit Sensormodulen verkleidet (links)
Einzelnes Sensormodul (rechts) (aus [7])

⁴ Die Firma wurde im Jahr 2000 von der Rohwedder Gruppe übernommen und in Rohwedder Systems umbenannt. Ob die abgebildeten Module noch hergestellt werden, ist nicht bekannt.

Ein anderes System mit einer künstlichen Sensorhaut aus optischen Abstandssensoren ist in [Lumelsky93] beschrieben. Dieses System ist im Bereich der Teleoperation angesiedelt. Das System ist in Abbildung 16 zu sehen. Der Roboterarm wird von einem menschlichen Bediener gesteuert und ist mit einer Sensorhaut überzogen. Wenn der Roboterarm zu nahe an ein Hindernis herangeführt wird, wird die Bahn so verändert, dass der Roboter tangential zur Oberfläche des Hindernisses verfährt. Dies entspricht einer komponentenweisen Geschwindigkeitsregelung. Damit gehört dieses System in die Kategorie *GE/LA*.

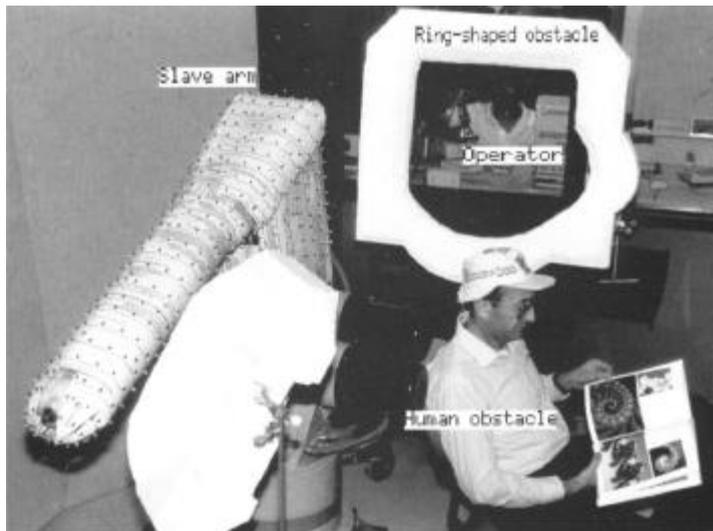


Abbildung 16: Master-Slave-System mit Sensorhaut aus optischen Abstandssensoren zur gesicherten Teleoperation (aus [Lumelsky93])

In [Martín99] wird ein System mit einer künstlichen Haut aus Ultraschallmodulen verwendet. Jedes Modul besteht aus Sender, Empfänger und einer lokalen Verarbeitungseinheit. Die insgesamt 13 Sensormodule sind auf den Ober- und Unterarmgliedern angebracht. Das System ist in Abbildung 17 zu sehen. Die Hindernisvermeidung wird über einen Reinforcement-Learning Ansatz realisiert. Das System soll selbständig freie Bewegungsbahnen zwischen vorgegebenen Zielpunkten finden. Da die Sensoren nicht den gesamten Roboter vor Kollision schützen und nur eine lokale Ausweichbewegungsplanung möglich ist, fällt dieses System in die Kategorie *ME/LA*.

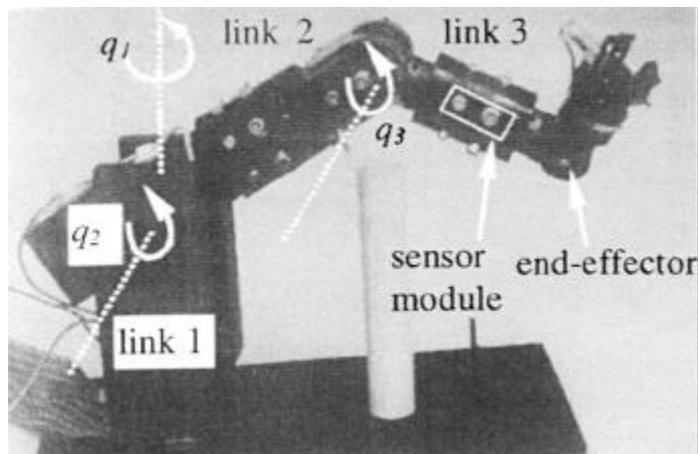


Abbildung 17: Roboter mit Sensorhaut aus Ultraschallmodulen (aus [Martín99])

2.3 Schlussfolgerungen

In Tabelle 2 sind die vorgestellten Systeme gruppiert nach Erkennungs- und Reaktionsstrategie. Auffallend ist, dass nur wenige Systeme die beste Erkennungsstrategie, die Ganzarmerkennung, verwenden. Keines der vorhandenen Systeme realisiert die beste Reaktionsstrategie, die globale Ausweichbewegung, wenn die Umgebung dynamische Hindernisse enthält.⁵

Reaktionsstrategie	Globale Ausweichbewegung				
	Lokale Ausweichbewegung		[Bischoff99] [Yu99] *)	[Feddema94] [Martín99] *) [Stettmer94]	[Lumelsky93] [Wegerif92]
	Geschwindigkeitsregelung	[Meisel94] *) [Morhard02] [Zettl02]		[Heiligensetzer02a] [Spingler02] [Zelinsky99]	
	Kollisionsentschärfung				[Morhard02] [Heiligensetzer02a] [Lim00] [Zelinsky99]
		Schutzraum	Einzelpunkt	Mehrpunkt	Ganzarm
		Erkennungsstrategie			

Tabelle 2: Einordnung der betrachteten Systeme. Systeme die mehrere Strategien verfolgen, sind in allen zutreffenden Klassen aufgeführt. Bei Systemen, zu denen mehrere Publikationen existieren, ist nur die jeweilige Hauptquelle aufgeführt.

**) Der Autor betrachtet eine andere Aufgabenstellung. Einordnung ist anhand der technischen Möglichkeiten erfolgt.*

Die Gründe für die Vermeidung der globalen Ausweichbewegung dürften vor allem in den hohen Anforderungen liegen, die diese Reaktionsstrategie an die Rechenleistung des Systems und die vorhandenen Sensoren stellt.

Ausgehend von der Erkennungs- und Reaktionsstrategie wird im folgenden Kapitel untersucht, aus welchen Komponenten ein System aufgebaut ist, das die Transferbewegung eines Roboters absichert. Die notwendigen Komponenten werden im Detail beschrieben.

⁵ Die Systeme aus [Yu99] und [Martín99] realisieren eine globale Ausweichbewegung durch sukzessive Exploration des Arbeitsraumes. Dies funktioniert nur bei statischer Hindernisumgebung.

Kapitel 3

System zur sicheren Transferbewegung

3.1 Konkretisierung der Aufgabenstellung

Damit die Anforderungen an das System zur sicheren Transferbewegung ermittelt werden können, muss die Aufgabenstellung aus Kapitel 1.3 konkretisiert werden.

Im Vergleich zu den im Kapitel 2.2 vorgestellten, bestehenden Systemen soll ein grundsätzlicher Vorteil existieren. Aus diesem Grund soll das zu realisierende System eine global geplante Ausweichbewegung auf der Basis einer Ganzarmerkennung verwenden. Dieses System fällt also in die Klasse *GE/GA*.

Um eine möglichst breite Anwendbarkeit zu bieten, sollten so wenig Einschränkungen wie möglich bezüglich der Hindernisse getroffen werden. Einfache Beispielhindernisse sind helle und dunkle Kisten. Reale Hindernisse sind Menschen in verschiedener Kleidung aber auch Getränkekästen und Getränkeflaschen. Eine Auswahl der Hindernisse ist in Abbildung 18 zu sehen.



Abbildung 18: Typische Hindernisse für das zu realisierende System.

Die Anzahl der Hindernisse und ihr Bewegungsverhalten war bislang nicht eingeschränkt. Es wird angenommen, dass im gemeinsamen Arbeitsraum typischerweise der

Roboter, der Mensch sowie ein weiteres Objekt vorhanden ist. Die Bewegung aller Hindernisse ist nicht vorhersehbar, erfolgt aber mit begrenzter Geschwindigkeit.

Typische Roboter für das zu realisierende System sind stationäre Roboter mit bekannter Geometrie. Es wird davon ausgegangen, dass die Position und Orientierung von mindestens einem Punkt des Roboters in bezug auf das Weltkoordinatensystem bekannt ist und dass die internen Sensoren es ermöglichen, die Konfiguration des Roboters zu ermitteln.⁶

3.2 Systemaufbau

Die Komponenten eines Systems, das die Transferbewegung eines Roboters absichert, können aus den zur Aufgabenlösung erforderlichen Informationen und Fähigkeiten abgeleitet werden. Ein System zur Sicherung der Transferbewegung braucht:

- Wissen über die Umwelt
- Wissen über die eigene Position und Form
- Fähigkeit, eine Kollision zu erkennen
- Fähigkeit, eine Kollision zu vermeiden

Die Fähigkeiten können direkt in entsprechenden Algorithmen zur Kollisionserkennung beziehungsweise Kollisionsvermeidung oder Bahnplanung realisiert werden. Das notwendige Wissen muss erworben werden durch Sensoren und die Fähigkeit die Sensordaten zu entsprechenden Modellen zu verarbeiten.

Damit besteht ein System zur Lösung der gestellten Aufgabe mindestens aus den folgenden Komponenten:

- Externe Sensoren
- Verarbeitung der externen Sensordaten
- Umweltmodell
- Interne Sensoren
- Verarbeitung der internen Sensordaten
- Robotermodell
- Kollisionserkennung
- Kollisionsvermeidung

3.2.1 Zusammenhang der Systemkomponenten

Die verschiedenen Komponenten eines Systems sind nicht unabhängig voneinander. Die Abhängigkeiten der verschiedenen Komponenten ist in Abbildung 19 dargestellt. Die Aufgabenstellung sowie die Zielvorgaben des Systems schlagen sich in der zu realisierenden Erkennungs- und Reaktionsstrategie, in den zu erkennenden Objekten sowie in den möglichen Robotertypen nieder.

⁶ Unter diesen Annahmen können für jeden Punkt des Roboters die Koordinaten im Weltkoordinatensystem ermittelt werden.

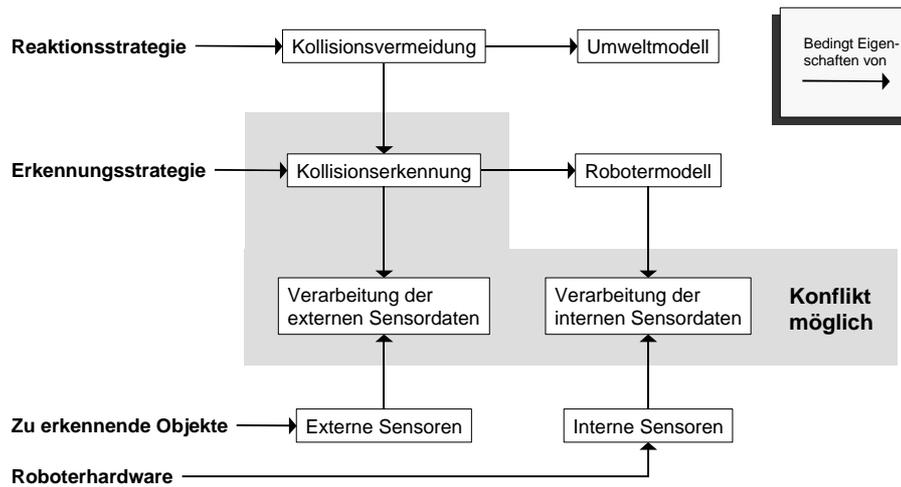


Abbildung 19: Abhängigkeiten der verschiedenen Komponenten eines Systems zur sicheren Transferbewegung. Konflikte sind bei denjenigen Komponenten möglich, die Bedingungen von zwei verschiedenen anderen Komponenten erfüllen müssen.

Die Reaktionsstrategie bedingt bestimmte Eigenschaften der Kollisionsvermeidung. Zum Beispiel ist eine globale Ausweichbewegung nur mit einem globalen Bahnplanungsverfahren zu realisieren. Die Kollisionsvermeidung bedingt ihrerseits das Umweltmodell sowie die Art der Kollisionserkennung. Beispielsweise benötigt ein globales Bahnplanungsverfahren ein Umweltmodell, das potentiell den gesamten Planungsraum darstellen kann sowie eine Kollisionserkennung, die einen beliebigen Punkt im Planungsraum bearbeiten kann.

Die Auswahl der Kollisionserkennung wird gleichzeitig durch die Erkennungsstrategie eingeschränkt. Dies kann unter Umständen zu Konflikten führen. Die Kollisionserkennung bestimmt ihrerseits die Eigenschaften des Robotermodells, welches wiederum bestimmte Anforderungen an die Verarbeitung der internen Sensordaten stellt.

Ebenso bedingen die verwendeten internen Sensoren die Art der Sensordatenverarbeitung. Wenn die internen Sensoren durch die Roboterhardware vorgegeben sind, kann dies zu Konflikten führen.

Die Kollisionserkennung bedingt zugleich Eigenschaften der Verarbeitung der externen Sensordaten. Denn die Sensordaten müssen so verarbeitet sein, dass die Kollisionserkennung daraus zusammen mit dem Robotermodell erkennen kann, wann eine Kollisionsgefahr besteht. Außerdem stellen die externen Sensoren bestimmte Anforderungen an die Sensordatenverarbeitung. Da die externen Sensoren durch die zu erkennenden Objekte eingeschränkt werden, kann dies zu Konflikten führen.

3.2.2 Zusammenfassung der Systemkomponenten

Die einzelnen Systemkomponenten müssen nicht notwendigerweise voneinander getrennt sein. Zum Beispiel ist es möglich, das Umweltmodell innerhalb der Kollisionsvermeidung aufzubauen.

Die internen Sensoren eines Industrieroboters sind normalerweise so ausgelegt, dass sie aus Anwendersicht keiner besonderen Verarbeitung bedürfen. Die Robotersteuerungen ermöglichen üblicherweise das Auslesen der Gelenkwinkel und damit die Bestimmung

der genauen Roboterkonfiguration. Daher werden im folgenden die internen Sensoren und die Verarbeitung der internen Sensordaten nicht betrachtet.

Die Komponenten „Externe Sensoren“ und „Verarbeitung der externen Sensordaten“ werden zu „externe Sensorik“ zusammengefasst. Die Motivation dahinter ist, dass die Sensordatenverarbeitung so stark von den gewählten Sensoren abhängt, dass eine unabhängige Betrachtung nicht sinnvoll ist.

Damit bleiben folgende zu untersuchende Komponenten, aus denen ein System zur sicheren Transferbewegung aufgebaut ist:

- Externe Sensorik
- Umweltmodell
- Robotermodell
- Kollisionserkennung
- Kollisionsvermeidung

3.3 Auswahl der Systemkomponenten

3.3.1 Externe Sensorik

Grundsätzlich ist es für das Arbeiten in unvollständig bekannten Umgebungen notwendig, aktuelle Informationen über den Zustand der Umgebung zu ermitteln. Daher lassen sich alle möglichen Ansätze nach der verwendeten Sensorik unterteilen. Außerdem wird durch die Wahl der Sensorik die möglichen Arten der Kollisionsvermeidung entscheidend eingeschränkt.

Aus den von den Sensoren gelieferten Informationen muss letztendlich auf die räumliche Präsenz von Objekten geschlossen werden können. Aus diesem Grund werden als Sensoren nur bildgebende Sensoren und Näherungs- bzw. Abstandssensoren betrachtet.

Auswahl der betrachteten Sensoriksysteme

Viele Sensoren lassen sich auf unterschiedliche Weise montieren, außerdem kann die Anzahl der verwendeten Sensoren verschieden sein. Die Montage der Sensoren legt fest, welchen Raumbereich der Sensor messen kann. Die verschiedenen Möglichkeiten lassen sich in folgende Klassen zusammenfassen:

- Stationäre Montage
- Montage auf aktiver Plattform
- Montage auf dem Roboter
- Montage als künstliche Haut

Die stationäre Montage ermöglicht bei geeigneter Positionierung die Erfassung des gesamten Arbeitsraumes. Außerdem kann durch die bekannte und ortsfeste Positionierung die Sensordatenverarbeitung einfacher werden. Der Nachteil ist, dass relevante Objekte durch andere Objekte im Arbeitsraum verdeckt sein können.

Die Montage auf einer aktiven Plattform ermöglicht, je nach Bewegungsfreiheit, sowohl den gesamten Arbeitsraum zu überwachen, als auch einen möglichst freien Blick. Allerdings stellt sich für die aktive Plattform im Prinzip die gleiche Kollisionsproblematik

wie für den Roboter dar. Aus diesem Grund wird die Montage auf einer aktiven Plattform im folgenden nicht weiter betrachtet. Der Aufwand der Sensordatenverarbeitung von Sensoren auf einer aktiven Plattform ist relativ hoch, da die Positionsveränderung der Plattform berücksichtigt werden muss.

Die Montage auf dem Roboter hat den Vorteil, dass praktisch keine Verdeckungen auftreten können. Der Nachteil jedoch ist, dass der Sensor häufig nicht in einer Position ist, um den gesamten Arbeitsraum einsehen zu können. Dies bedeutet, dass auf den Sensordaten keine globale Bahnplanung möglich ist. Außerdem kann der Roboter mit einem Teil außerhalb des Sensorbereiches kollidieren. Der Aufwand der Sensordatenverarbeitung ist relativ hoch, da die Position des Roboters berücksichtigt werden muss.

Die Montage als künstliche Haut ist eine Variante der Montage auf dem Roboter. Dabei werden mehrere Sensoren, verteilt über den kollisionsgefährdeten Teil des Roboters, montiert. Ein Problem ist, dass bei der Verwendung von vielen Sensoren immer mit dem Ausfall von einigen Sensoren gerechnet werden muss. Dies muss durch eine fehler-tolerante Sensordatenverarbeitung ausgeglichen werden. In [Um98] wird gezeigt, dass dies prinzipiell möglich ist, jedoch einen erheblichen Aufwand bedeutet. Zudem ist es normalerweise nicht möglich, das Werkzeug und sonstige Aufbauten auf dem System mit der Haut zu überziehen, so dass diese gesondert abgesichert werden müssen. Weiterhin kann mit den einfachen Sensoren nur die nächste Umgebung des Roboters erfasst werden. Deshalb ist eine global geplante Ausweichbewegung unmöglich.⁷

Aus diesem Grund werden nur stationäre Sensoren betrachtet. Von den bereits realisierten Systemen ausgehend wurden für die Sensorik folgende Alternativen ausgewählt:

- A) Stationäre Standard-Kameras
- B) Stationäre Hochfrequenz Kameras
- C) Stationäre 3D-Laserscanner
- D) Stationäre 2D-Laserscanner

Kriterien zur Bewertung der Sensorik

Die Auswahl einer geeigneten Sensorik⁸ zur Lösung der Aufgabe, erfolgt nach bestimmten Kriterien. Diese sind:

- Eignung
- Messbereich
- Messfrequenz
- Messpunktabstand
- Aufwand der Sensordatenverarbeitung
- Kosten

⁷ Bei der Verwendung von leistungsfähigeren Sensoren, wie Kameras oder Laserscanner, ist es möglich, dass der Arbeitsraum hinreichend erfasst wird, so dass der Unterschied zwischen lokaler und globaler Ausweichbewegung in der Praxis verschwindet. Zur Zeit ist dies jedoch, abgesehen von den hohen Kosten, wegen der erforderlichen flachen Bauweise der Sensoren kaum möglich.

⁸ Mit Sensorik ist die Kombination aus externem Sensor und Sensordatenverarbeitung gemeint.

Aus der Aufgabenstellung können die Anforderungen an die Sensoren abgeleitet werden. Ziel ist es, ein System zu realisieren, das die Transferbewegung eines Roboterarms absichert und dabei dynamischen Hindernissen durch eine globale Ausweichbewegung auf der Basis einer Ganzarmerkennung ausweicht.

Die Bewertung erfolgt dreistufig mit gut (+1), mittel (± 0), schlecht (-1). Die Summe der Einzelbewertungen ergibt die Gesamtbewertung.

Das Problem bei dieser Vorgehensweise ist der Übergang von der Ordinalskala (gut, mittel, schlecht) zur metrischen Skala. Da die Eignung und der Aufwand der Sensordatenverarbeitung nicht quantitativ erfasst werden können und gleichzeitig eine Verrechnung der Einzelbewertung zu einem Gesamtergebnis notwendig ist, bleibt keine andere Möglichkeit.⁹

Eignung

Die Eignung bedeutet, mit welcher Sicherheit man aus den Messwerten auf die reale Situation schließen kann. Angewendet auf die Hindernisvermeidung bedeutet das, wie sicher man sein kann, dass kein Hindernis im Weg ist, wenn die Sensorik eine freie Bahn anzeigt, beziehungsweise, wie sicher man sein kann, dass ein von der Sensorik gemeldetes Hindernis wirklich existiert. Dies bedeutet wie hoch die Wahrscheinlichkeit der Sensorik für Fehler 1. Art (Hindernis nicht erkannt) und Fehler 2. Art (Hindernis fälschlich gemeldet) ist.

Das Bewertungsergebnis aus der Menge { gut, mittel, schlecht } beschreibt, wie groß die Gefahr von Fehlern 1. und 2. Art bei der Erkennung von Hindernissen ist.

Messbereich

Der Messbereich besagt, wie weit die Sensoren von den zu messenden Objekten entfernt sein dürfen. Damit jederzeit alle relevanten Hindernisse erkannt werden können, muss die maximale Messentfernung mindestens so groß sein, wie die maximale Ausdehnung des überwachten Raumes. Bei Anwendungen in Innenräumen mit üblichen Industrierobotern ist eine maximale Messentfernung von 10m in den meisten Fällen ausreichend.

Da der zur Verfügung stehende Montageraum aber begrenzt ist, sollten die Sensoren in der unmittelbaren Umgebung des überwachten Bereiches montiert werden können. Dennoch müssen Objekte auch dann erkannt werden, wenn sie nah an den Sensor herankommen. Die minimale Messentfernung sollte also klein sein.

Die Bewertung des Messbereichs erfolgt nach dem Schema in Tabelle 3:

Maximale Messentfernung	$\geq 10\text{m}$	Gut (+1)	Mittel (± 0)
	$< 10\text{m}$	Mittel (± 0)	Schlecht (-1)
		$< 1\text{m}$	$> 1\text{m}$
Minimale Messentfernung			

Tabelle 3: Bewertungsschema für die Sensorikeigenschaft „Messbereich“

⁹ Selbst wenn die quantitative Erfassung aller Eigenschaften mögliche wäre, bliebe das Problem der Umrechnung der gemessenen Werte in die zu maximierenden Nutzeinheiten.

Messfrequenz

Die Messfrequenz beschreibt, wie häufig pro Zeiteinheit die Sensorik neue Daten liefern kann. Damit legt die Messfrequenz eine untere Schranke für die Aktualisierungsrate des Weltmodells fest.

Die Messfrequenz muss so groß sein, dass die Positionsveränderung der Hindernisse zwischen zwei Messzeitpunkten so klein ist, dass sie nicht mit dem Roboter kollidieren. In diese Positionsveränderung geht neben der Messfrequenz die Geschwindigkeit von Hindernissen und Roboter ein. Im Bereich des Visual Servoing wird eine Regelfrequenz von 1kHz angestrebt. Für die kollisionsfreie Transferbewegung ist eine Messfrequenz größer 10 Hz wünschenswert.

Die Bewertung erfolgt nach dem Schema in Tabelle 4.

Messfrequenz	Bewertung
> 50 Hz	Gut (+1)
10 Hz ... 50 Hz	Mittel (± 0)
< 10 Hz	Schlecht (-1)

Tabelle 4: Bewertungsschema für die Sensorikeigenschaft „Messfrequenz“

Messpunktabstand

Der Messpunktabstand besagt, wie die räumliche Beziehung der verschiedenen Messpunkte zueinander ist. Daraus kann abgeleitet werden, wie groß die zu erkennenden Objekte mindestens sein müssen. Objekte, die in einer Achse eine kleinere Ausdehnung haben als der Messpunktabstand werden unter Umständen nicht erkannt.

Die Bewertung erfolgt nach dem Schema in Tabelle 5.

Messpunktabstand	Bewertung
> 5 cm	Schlecht (-1)
1 cm ... 5 cm	Mittel (± 0)
< 1 cm	Gut (+1)

Tabelle 5: Bewertungsschema für die Sensorikeigenschaft „Messpunktabstand“

Aufwand der Sensordatenverarbeitung

Der Aufwand der Sensordatenverarbeitung beschreibt den Aufwand, der notwendig ist, um das Weltmodell aus den Sensordaten aufzubauen oder zu aktualisieren. Die Abschätzung kann nur qualitativ erfolgen. Ein hoher Aufwand wird als schlecht (-1) bewertet, entsprechend ein durchschnittlicher Aufwand als mittel (± 0) und ein niedriger Aufwand als gut (+1).

Wichtige Einflussgrößen für den Aufwand der Sensordatenverarbeitung sind die Anzahl der verwendeten Sensoren, die zu verarbeitende Datenmenge pro Sensor und Zeiteinheit. Daneben spielt noch der prinzipielle Aufwand eine Rolle, die Messdaten so aufzubereiten, dass sie in das Weltmodell integriert werden können.

Kosten

Das Kriterium „Kosten“ bereitet bei einer wissenschaftlichen Betrachtung Probleme. Einerseits kann es nicht ignoriert werden, da sowohl zum Zeitpunkt der Forschung und Entwicklung eines Systems als auch beim späteren Einsatz die zur Verfügung stehenden Ressourcen begrenzt sind. Andererseits sind die Kosten nicht fest, sondern verändern sich über die Zeit. Aus diesem Grund wird die Entscheidung der optimalen Sensorik, je nach Budget, Ort und Zeit auch bei sonst gleichen Voraussetzungen verschieden ausfallen können. Dadurch widersprechen Aussagen, die auf zeitabhängigen Kosten basieren, dem eigentlichen Ziel der wissenschaftlichen Arbeit, dauerhaft gültige Erkenntnisse zu finden. Dauerhafte Erkenntnisse können nur dann gewonnen werden, wenn die Kosten für die Zukunft prognostiziert werden.

Die Gesamtkosten setzen sich zusammen aus den Einzelkosten des Sensors und der notwendigen Anzahl. Allerdings sind auch die anderen Kriterien nicht unabhängig von den Kosten. Einige Sensoren, zum Beispiel Laserscanner, sind mit verschiedenen Messentfernungen und -frequenzen erhältlich. Unter sonst gleichen Bedingungen sind beide Faktoren mit dem Preis positiv korreliert. Im Prinzip sind auch alle weiteren Kosten zu berücksichtigen, die durch die Wahl des Sensors verursacht werden. Zum Beispiel kann eine aufwendigere Sensordatenverarbeitung höhere Kosten verursachen, wenn ein leistungsfähigerer Rechner notwendig ist. Ebenso müssten bei einer vollständigen Betrachtung die Entwicklungskosten berücksichtigt werden. Dies würde jedoch den Rahmen sprengen.

Im Folgenden wird angenommen, dass sich die Kosten von allen Sensorikvarianten relativ zueinander nicht verändern. Unter dieser Annahme ist die Rangordnung der Gesamtbewertung der Sensorik unabhängig von der Zeit.

Stationäre Kameras (Standard und Hochfrequenz)

Kameras gibt es in großer Bandbreite. Allen gemeinsam ist, dass sie die von Objekten ausgehenden oder reflektierten elektromagnetischen Wellen in einer Ebene abbilden. Üblicherweise ist die Bildebene als Matrix aus Bildpunkten (Pixel) realisiert.¹⁰ Die von der Kamera aufgenommenen Frequenzen können von thermischem Infrarot über den sichtbaren Bereich bis hin zu ultravioletter Strahlung reichen. Allerdings kann eine einzelne Kamera normalerweise nur einen kleinen Teil des Frequenzbandes darstellen.

Zur Hinderniserkennung genügen in der Regel Kameras, die das sichtbare Spektrum erfassen. Übliche CCD-Kameras erfassen meist noch Frequenzen, die jenseits des sichtbaren Bereichs liegen, vor allem im nahen Infrarotbereich. Hierbei gibt es zwei große Gruppen, die Grauwert- und die Farbkameras. Bei Grauwertkameras wird für jeden Pixel die über den ganzen Spektralbereich integrierte Intensität gemessen.

Farbkameras sind im Prinzip aus mehreren Grauwertkameras aufgebaut. Im Lichtweg sind Farbfilter eingebaut, die das einfallende Licht in je eine Rot- Grün- und Blau-Komponente zerlegen. Der Pixelwert ist das Tripel der über den jeweiligen Spektralbereich integrierten Intensitäten. In der technischen Realisierung unterscheidet man noch zwischen der 3-Chip Methode, die für jede Farbe eine eigene Filter/Sensoreinheit hat

¹⁰ Sogenannten Zeilenkameras sind der Extremfall, bei dem die Matrix zu einem eindimensionalen Vektor reduziert ist. Sie werden gerne zur Erkennung von Objekten, die auf Förderbändern liegen, verwendet. Zur Überwachung ganzer Arbeitsräume sind sie ungeeignet.

und der 1-Chip Lösung, bei der die Farbfilter direkt vor den lichtempfindlichen Sensoren liegen. Für die Hinderniserkennung ist der Unterschied jedoch ohne Bedeutung.

Im folgenden wird aus Gründen der Vereinfachung von Kameras ausgegangen, die das sichtbare Spektrum abbilden. Ein Beispiel ist in Abbildung 20 zu sehen.



Abbildung 20: Standardkamera, wie sie im realisierten System verwendet wird. ([8])

Eine entsprechend leistungsfähige Datenverarbeitung vorausgesetzt, kann aus den Kamerabildern ziemlich sicher auf die Präsenz oder Abwesenheit von Hindernissen geschlossen werden. Problematisch sind für Kameras Objekte, die transparent (z.B. Glas) oder reflektierend (Spiegel) sind. Die Eignung von Kameras zur Hinderniserkennung ist insgesamt „gut“ (+1). Gegen diese Bewertung könnte man einwenden, dass sie keinen Unterschied zwischen Grauwert- und Farbkameras macht. Dies ist jedoch insofern gerechtfertigt, als Farbkameras durch den Einsatz von mehreren mit Farbfiltern versehenen Grauwertkameras ersetzt werden können.

Die maximale Messentfernung von Kameras ist theoretisch unendlich, da sich elektromagnetische Wellen beliebig weit fortbewegen können. Praktisch gibt es Einschränkungen, aber der zur Hinderniserkennung relevante Bereich kann erfasst werden. Die minimale Messentfernung von Kameras ist größer Null, da Objekte direkt vor der Kamera den Lichtweg blockieren. Allerdings genügt ein kleiner Abstand ($<1\text{m}$) damit die Hindernisse bei üblichen Bedingungen erkannt werden. Damit ist die Bewertung des Messbereichs „gut“ (+1).

Die Messfrequenz beträgt bei Standard-Kameras üblicherweise 25-30 Hz. Hochfrequenz-Kameras erreichen teilweise Messfrequenzen über 100 Hz. Bei sogenannten CMOS-Kameras ist nicht die maximale Anzahl der gelieferten Bilder, sondern die maximale Anzahl der gelieferten Pixel fest vorgegeben. Die Messfrequenz bei einfachen Kameras ist „mittel“ (± 0), bei hochwertigen Kameras „gut“ (+1).

Kameras bilden die nebeneinanderliegenden Pixel sehr eng benachbarter Raumbereiche ab. Problematisch können Objekte werden, deren Abbildung kleiner als zwei Pixel ist. Die Abbildungsgröße ist stark abhängig von der Entfernung der Kameras zu den Objekten und den verwendeten Objektiven. Mit einer guten Optik werden aber Objekte von 1cm Größe bei einem Abstand von 10m erkannt. Der Messpunktabstand ist damit „gut“ (+1).

Der Aufwand der Sensordatenverarbeitung kann je nach verwendetem Verfahren schwanken. Nachteilig ist, dass Kameras nur 2D-Bilder liefern. Durch die Verwendung mehrerer Kameras kann dieser Nachteil jedoch umgangen werden (z.B. Stereosehen). Beim Stereosehen ist das Korrespondenzproblem zu lösen. Außerdem müssen aus den Pixelmengen die relevanten Objekte erkannt werden. Die Lösung dieser Probleme ist schwierig und erfordert einen hohen Aufwand bei der Sensordatenverarbeitung. Bei auf dem Manipulator angebrachten Kameras muss die durch die Bewegung des Roboters

verursachte Bildveränderung von den Veränderungen durch die Objektbewegungen unterschieden werden. Die hierzu erforderlichen Verfahren sind aufwändig. Die Bewertung des Aufwands bei der Sensordatenverarbeitung ist daher „schlecht“ (-1).

Die Kosten für ein Sensoriksystem aus Kameras können sehr verschieden sein. Für ein System aus einfachen Standardkameras mit einfachen Objektiven und Framegrabbern sind sie niedrig (+1). Bei der Verwendung von Kameras mit hohen Messfrequenzen und entsprechenden Framegrabbern mit eigenen Datenverarbeitungsprozessoren sind die Kosten hoch (-1).

2D und 3D Laserscanner

In der Automatisierungstechnik und in der Robotik werden 2D Laserscanner (Laserscanner) häufig eingesetzt. Sie erzeugen ein eindimensionales Tiefenbild. Im Gegensatz dazu liefern 3D Laserscanner ein zweidimensionales Tiefenbild. Bei Tiefenbildern kodieren die Grauwerte der einzelnen Pixel die Entfernung zum abgebildeten Objekt. Ein Tiefenbild ist in Abbildung 21 zu sehen.

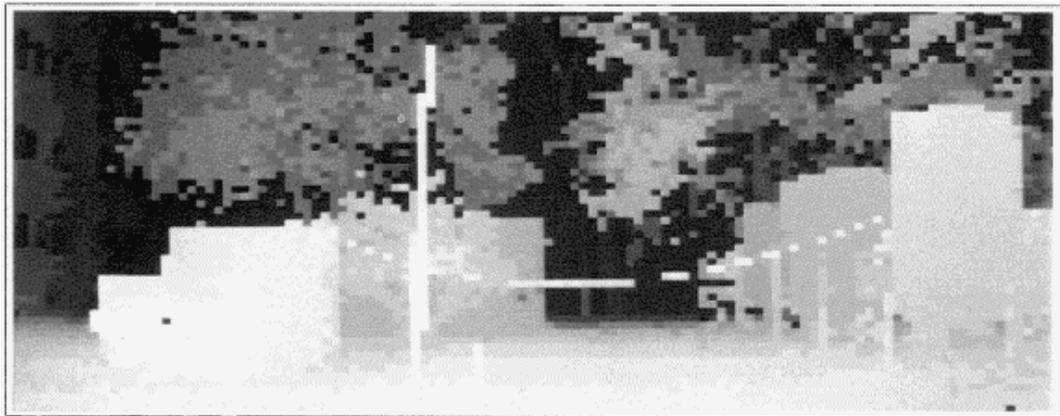


Abbildung 21: Beispielscan des Dornier EBK Laserscanners (aus [Dornier93])

Von einigen (wenigen) Herstellern gibt es 3D Laserscannersysteme, die Tiefenbilder erzeugen. Zwei Geräte sind in Abbildung 22 zu sehen. In [Fiebert96] wurde die Umweltmodellierung mittels eines 3D Laserscanners experimentell untersucht.



Abbildung 22: 2D Laserscanner (links, SICK ([9])) und 3D Laserscanner (Mitte: Minolta ([10]), rechts: Dornier ([Dornier93]))

Laserscanner messen direkt die Entfernung zu Objekten in der Umgebung. Damit eignen sie sich sehr gut zur Erkennung der Präsenz bzw. Absenz von Objekten. Lediglich

transparente oder reflektierende Objekte können Schwierigkeiten bereiten. Die Bewertung der Eignung ist damit „gut“ (+1).

Der Messbereich hängt davon ab, ob der Laserscanner das Messprinzip der Laufzeitmessung oder Triangulation verwendet. Allerdings ist bei den untersuchten 3D Laserscannern die minimal erlaubte Messentfernung größer als 1m. Außerdem ist bei einigen der Geräten mit Triangulation die maximale Messentfernung kleiner als 10m. Damit ist die Bewertung des Messbereiches „mittel“ (± 0), für die 3D-Laserscanner, die eine maximale Messentfernung von mehr als 10 m haben. Für 3D-Laserscanner mit einer kleineren maximalen Messentfernung ist sie „schlecht“ (-1).

2D Laserscanner decken den Messbereich von 1 m bis 10 m ab. Die Bewertung ist damit „gut“ (+1).

Die Messfrequenz beträgt bei den betrachteten Laserscannern unter 10 Hz. Damit ist die Bewertung „schlecht“ (-1).

Der zur Erkennung der Objekte verwendete Laserstrahl fächert sich weniger auf, als zu einer vollständigen Abdeckung notwendig wäre. Aus diesem Grund sind zwischen den einzelnen Messpunkten Lücken. Die Größe der Lücken hängt vom Abstand des Objektes ab. Typische Winkelauflösungen sind etwa $0,5^\circ$. Bei einer Messentfernung von 10m ist dann der Messpunktabstand etwa 8,7 cm und ist damit als „schlecht“ (-1) zu bewerten.

Bei 2D Laserscannern muss außerdem die Raumabdeckung durch die Verwendung von mehreren Laserscannern erfolgen. Bedingt durch die Bauhöhe wird dabei der Abstand normalerweise größer als 5 cm. Die Bewertung des Messpunktabstandes wäre damit auch bei kleinerer Winkelauflösung „schlecht“ (-1).

Bei Laserscannern können die Daten relativ einfach in das Weltmodell integriert werden, da bereits Abstände gemessen werden. Die Anzahl der Messdaten pro Sensor und Zeiteinheit ist mäßig groß. Bei der Verwendung von Laserscannern ist der Aufwand der Sensordatenverarbeitung daher „gut“ (+1).

Die Kosten für 3D Laserscanner sind sehr hoch, die Kosten für 2D Laserscanner sind hoch. Bedingt durch die notwendige hohe Anzahl bei 2D Laserscannern ist die Bewertung der Kosten in beiden Fällen „schlecht“ (-1).

Ergebnis

Das Ergebnis der obigen Bewertung ist in Tabelle 6 zusammengefasst.

Kriterium	Sensoriksystem			
	Standardkame- ras	Hochfre- quenzkame- ras	2D Laserscan- ner	3D Laserscan- ner
Eignung	+	+	+	+
Messbereich	+	+	+	o / -
Messfrequenz	o	-	-	-
Messpunktabstand	+	+	-	-
Aufwand der Sensordatenverarbeitung	-	-	+	+
Kosten	+	-	-	-
Gesamtbewertung	+3	+2	±0	-1 / -2

Tabelle 6: *Bewertung der betrachteten Sensorikvarianten*

Ausgehend von dieser Bewertung wird als Sensorik für das zu realisierende System die Lösung mit stationären Standardkame- ras gewählt.

3.3.2 Umweltmodell

Das Umweltmodell ist die rechnerinterne Repräsentation der Umgebung des Roboters. Bei der sicheren Mensch-Roboter-Koexistenz muss mindestens der gemeinsame Arbeitsraum von Mensch und Roboter modelliert werden.

Modellierungsraum

Bei der Modellierung der Umwelt ist zu entscheiden, ob die Objekte im Arbeitsraum oder im Konfigurationsraum repräsentiert werden sollen.

Die Modellierung im Arbeitsraum bildet direkt die räumliche Position und Orientierung der verschiedenen Objekte ab. Diese Modellierung ist für den Menschen intuitiv verständlich. Der Nachteil besteht darin, dass der Roboter mit modelliert werden muss.

Eine Konfiguration ist eine Menge von unabhängigen Parametern, welche die Position von jedem Punkt des Objektes vollständig beschreibt [Hwang92]. Der Konfigurationsraum ist die Menge aller zulässigen Konfigurationen. Da bereits eine Roboterkonfiguration die Position von allen Punkten des Roboters beschreibt, reduziert sich die Darstellung des Roboters in seinem Konfigurationsraum auf einen einzigen Punkt. Dies macht die Kollisionsvermeidung besonders einfach. Der Nachteil ist die notwendige Transformation der anderen Umweltobjekte in den Konfigurationsraum.

Bei Industrierobotern wird häufig das Tupel der Gelenkstellungen als Konfiguration bezeichnet. Genaugenommen ist dies nicht zutreffend, da zur vollständigen Beschreibung der Position noch die Position und Orientierung mindestens eines Punktes im kartesischen Raum notwendig ist. Dennoch wird im folgenden dieser eingeführte Begriff verwendet.

Repräsentation des Modellierungsraumes

Die Umwelt kann sowohl bei der Arbeitsraum- als auch bei der Konfigurationsraummodellierung auf verschiedene Weise dargestellt werden. In [Hwang92] werden einige Möglichkeiten der Umweltrepräsentation beschrieben. Diese sind in Abbildung 23 dargestellt und werden im Folgenden beschrieben.

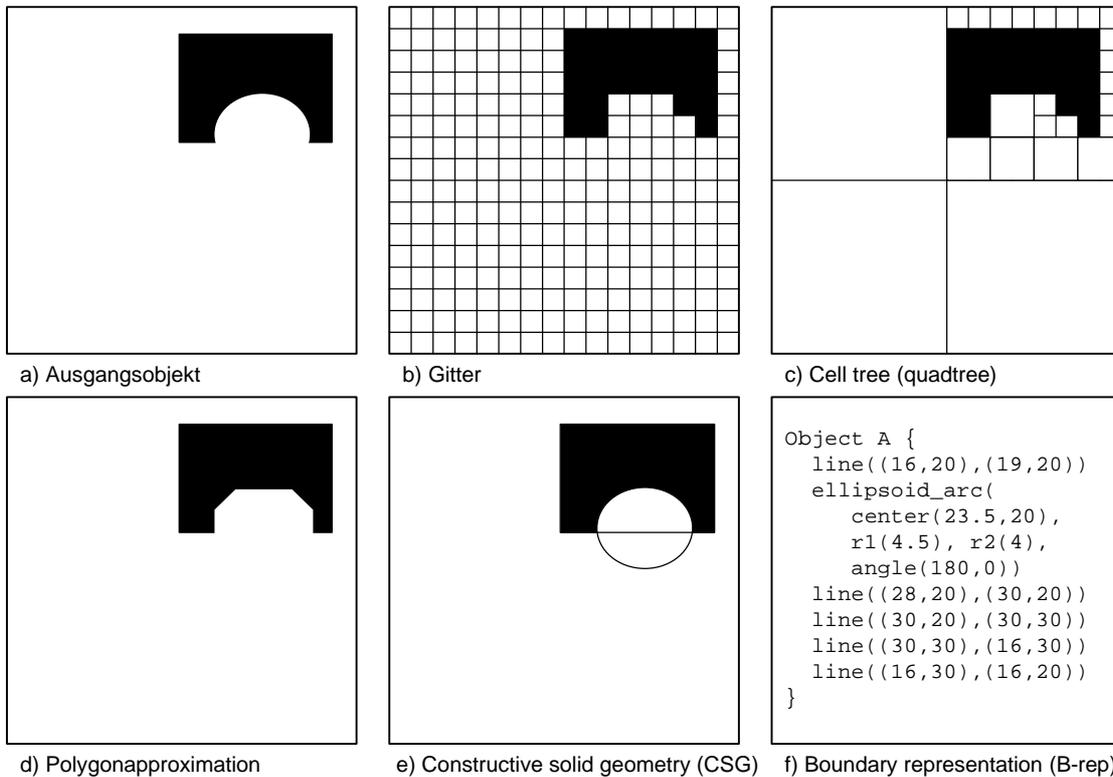


Abbildung 23: Methoden der Umweltmodellierung. Die Modellierung des Ausgangsobjekts a) durch die verschiedenen Methoden ist in den Teilabbildungen b) bis f) zu sehen. (ähnlich [Hwang92])

Bei der Gitterdarstellung besteht die Umwelt aus Zellen, die entweder belegt oder frei sein können. Diese Darstellung hat einen großen Speicherverbrauch. Allerdings ist die Bearbeitung von Objekten in Gitterdarstellung sehr einfach.

Die cell-tree Darstellung versucht den Nachteil des hohen Speicherverbrauchs der Gittermethode zu vermeiden. Dabei wird der zu modellierende Raum in Zellen unterteilt. Zellen die komplett innerhalb oder außerhalb eines Objektes liegen, werden als belegt bzw. frei markiert. Alle gemischten Zellen werden wiederum unterteilt. Dieser Prozess geht so lange bis alle Zellen belegt oder frei sind oder eine Mindestgröße erreicht ist. Bei einer häufig verwendeten Methode werden die Zellen bei der Unterteilung in allen Dimensionen halbiert. Diese Methode heißt Quadtree bei einer 2D Darstellung bzw. Octree in 3D. Der Hauptnachteil im Vergleich zur Gitterdarstellung ist der Aufwand, um Nachbarschaftsbeziehungen zwischen Zellen zu ermitteln.

Bei der Polygondarstellung wird die Umwelt durch Mengen von Polygonen (bzw. Polyedern bei 3D) dargestellt. Mit Hilfe dieser Darstellung lassen sich der Schnitt von Objekten und der Abstand zwischen Objekten schnell berechnen. Aus diesem Grund wird diese Darstellung in der Computergrafik häufig verwendet. Der Hauptnachteil bei der gegebenen Aufgabenstellung besteht im hohen Aufwand, aus den Sensordaten die Polygondarstellung zu erzeugen.

Bei der Constructive solid geometry (CSG) wird die Umwelt durch die Vereinigung, Schnitt und Differenz von Grundobjekten beschrieben. Bei der Boundary representation (B-rep) werden die Grenzen der Objekte durch Grundelemente wie Linien und Kreisbögen beschrieben. Der Vorteil beider Verfahren besteht darin, dass auch Objekte mit komplexen Formen durch wenige Grundelemente beschrieben werden können. Der Nachteil besteht in der aufwendigen Rechnung mit den Objekten und im hohen Aufwand die Darstellung aus den Sensordaten zu erzeugen.

Ergebnis

Aus den obigen Überlegungen heraus wird als Umweltdarstellung ein Gitter im Konfigurationsraum gewählt. Diese Darstellung hat den großen Vorteil, dass die Umwelt sehr einfach bearbeitet werden kann und der Roboter in der Umweltdarstellung nicht mehr auftaucht. Daher braucht die Robotergeometrie bei der Kollisionsvermeidung nicht mehr berücksichtigt zu werden. Für eine Transferbewegung ist keine besonders feine Diskretisierung des Raumes notwendig, daher ist der hohe Speicherverbrauch vertretbar.

Allerdings wird die Umwelt nur innerhalb der Kollisionsvermeidung modelliert. Dies hat den Vorteil, dass nur derjenige Teil der Umwelt modelliert werden muss, der für die Kollisionsvermeidung relevant ist.

3.3.3 Robotermodell

Die Aufgabe der Robotermodellierung besteht darin, dem System Wissen über die Form und räumliche Ausdehnung des Roboters zu geben. Hierzu werden die internen Sensoren und das Wissen über die räumliche Positionierung und die Geometrie des Roboters verwendet.

Bei der Modellierung der Umwelt im Arbeitsraum wird das Robotermodell verwendet, um die räumliche Ausdehnung und Position des Roboters zu berechnen. Bei der Modellierung der Umwelt im Konfigurationsraum wird das Robotermodell verwendet, um die Hindernisse in den Konfigurationsraum abzubilden.

Das Robotermodell besteht aus mehreren Teilen. Das *kinematische Modell* beschreibt den Zusammenhang zwischen den Gelenkstellungen des Roboters und der Position und Orientierung von Bezugspunkten im kartesischen Raum. Das *Geometriemodell* beschreibt die räumliche Ausdehnung des Roboters, ausgehend von den Bezugspunkten des kinematischen Modells.

Bei Industrierobotern wird die Kinematik mit Hilfe von homogenen Matrizen nach dem sogenannten Denavit-Hartenberg-Standard beschrieben. Dieser Standard legt die Lage der Bezugskoordinatensysteme ausgehend von den rotatorischen Achsen des Roboters fest. Die Lage jedes Punktes des Roboters, relativ zu einem bestimmten Bezugskoordinatensystem, lässt sich mit einer konstanten Transformationsmatrix beschreiben.

Bei der Wahl der Repräsentation der Geometrie des Roboters im Rechner gibt es die gleichen Möglichkeiten, wie sie im vorherigen Abschnitt bei der Umweltmodellierung dargestellt wurden. Allerdings ist die Bewertung der verschiedenen Möglichkeiten auf Grund der unterschiedlichen Voraussetzungen eine andere. Da die Robotergeometrie bekannt ist, bietet sich eine Polygondarstellung an. Sie ermöglicht eine schnelle Berechnung der räumlichen Ausdehnung des Roboters.

Die Ausgabe des Robotermodells muss letztendlich von der anfragenden Einheit (z.B. der Kollisionserkennung) bearbeitet werden können. Aus diesem Grund kann es notwendig sein, das Geometriemodell des Roboters weiteren Verarbeitungsschritten zu unterziehen. Bei der realisierten bildbasierten Kollisionserkennung besteht dieser zusätzliche Schritt in einer Projektion des Roboters in die Bildebenen der Kameras.

3.3.4 Kollisionserkennung

Die Aufgabe der Kollisionserkennung besteht darin, die Erkennungsstrategie zu realisieren. Im gegebenen Fall muss also eine Ganzarmerkennung realisiert werden. Da andererseits die Eigenschaften der Kollisionserkennung auch durch die Kollisionsvermeidung bestimmt werden, muss die Kollisionserkennung eine globale Ausweichbewegung ermöglichen.

Die verschiedenen Möglichkeiten, eine Kollisionserkennung zu realisieren sind in Abbildung 24 dargestellt und werden im folgenden diskutiert.

Eine globale Kollisionserkennung kann Eingaben aus dem gesamten Konfigurationsraum bearbeiten, unabhängig von der aktuellen Roboterkonfiguration. Im Gegensatz dazu kann bei einer lokalen Kollisionserkennung nur die Nachbarschaft der aktuellen Roboterkonfiguration bearbeitet werden. Zur Realisierung einer globalen Ausweichbewegung ist eine globale Kollisionserkennung erforderlich.

Es gibt zwei Standardverfahren zur Kollisionserkennung, die Abstandsberechnung und den Kollisionstest. Eine Abstandsberechnung berechnet den kleinsten Abstand zwischen Roboter und nächstem Hindernis. Als Zusatzinformation kann häufig auch die Richtung des minimalen Abstandes sowie das nächstgelegene Objekt ermittelt werden. Der Kollisionstest liefert lediglich die binäre Antwort, ob der Roboter mit dem Hindernis kollidiert oder nicht.

Die Kollisionserkennung kann in verschiedenen Räumen arbeiten. Eventuell ist es möglich, direkt aus den Sensordaten eine Antwort zu erzeugen. Es ist aber auch eine Transformation der Sensordaten in das Umweltmodell denkbar und dann das Treffen der Kollisionsentscheidung im Umweltmodell.

Die Kollisionserkennung wird von der Kollisionsvermeidung sehr häufig angefragt. Daher ist es notwendig, die Kollisionsentscheidung so schnell wie möglich zu treffen.

Normalerweise ist eine Abstandsberechnung aufwendiger als ein Kollisionstest. Dieser Mehraufwand ist nur dann sinnvoll, wenn die Kollisionsvermeidung die Zusatzinformation über den Abstand zum nächsten Hindernis nutzen kann. Dies ist bei den (im folgenden) betrachteten Bahnplanungsverfahren jedoch nicht der Fall.

Eine Transformation in das Umweltmodell ist normalerweise aufwendiger als das Arbeiten direkt auf den Sensordaten.

Im realisierten System wird daher ein globaler Kollisionstest verwendet, der, soweit wie möglich, auf den Bildern direkt arbeitet.

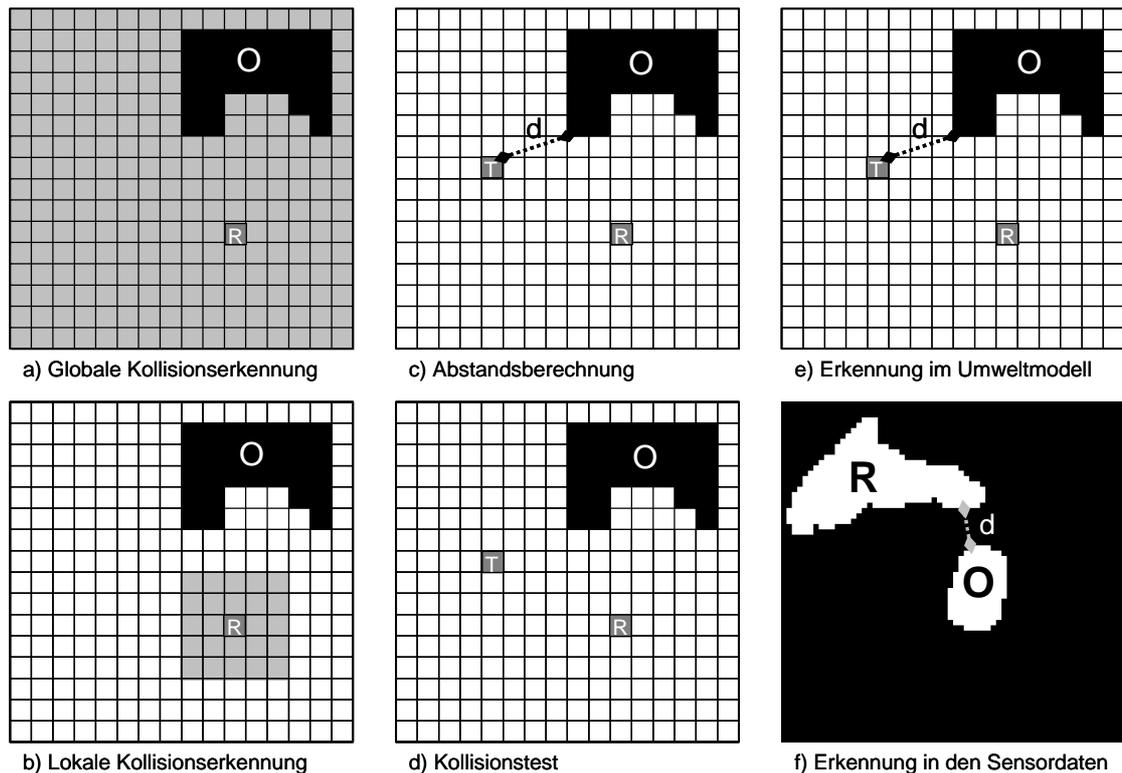


Abbildung 24: Verfahren der Kollisionserkennung. a) ,b): die aktuelle Roboterkonfiguration ist R, die grau hinterlegten Konfigurationen sind die zulässigen Testkonfigurationen. Bei der globalen Kollisionserkennung ist der gesamte Konfigurationsraum zulässig, bei der lokalen Kollisionserkennung nur die Nachbarschaft der aktuellen Konfiguration. c), d): die aktuelle Roboterkonfiguration ist R, angefragte Testkonfiguration ist T, während die Abstandsberechnung den kürzesten Abstand d zum Hindernis bestimmt, liefert der Kollisionstest nur die Information, dass T kollisionsfrei ist. e), f): Die Kollisionserkennung, am Beispiel der Abstandsberechnung, der Abstand kann im Umweltmodell (hier: Konfigurationsraum) oder in den Sensordaten (hier: Silhouetten) erfolgen.

3.3.5 Kollisionsvermeidung

Die Aufgabe der Kollisionsvermeidung ist die Umsetzung der Reaktionsstrategie. Eine global geplante Ausweichbewegung kann durch ein globales Bahnplanungsverfahren realisiert werden. Die Bahnplanung hat die Aufgabe, ausgehend von einer gegebenen Startkonfiguration eine kollisionsfreie Bahn zu einer gegebenen Zielkonfiguration zu finden.

Dieses Problem ist in der Robotik gut untersucht. Bislang wurden jedoch hauptsächlich Probleme betrachtet, in denen der betrachtete Raum entweder statisch und vollständig bekannt ist (bei stationären Robotern) oder der Raum dynamisch aber nie vollständig bekannt ist (bei mobilen Robotern).

Im realisierten System wird durch die stationären Kameras der gesamte gemeinsame Arbeitsraum von Mensch und Roboter erfasst. Die Kollisionserkennung ermöglicht den Test einer beliebigen Konfiguration. Wenn die Umwelt als quasistatisch betrachtet wird, ist eine echte Bahnplanung mit den üblichen Bahnplanungsverfahren möglich.

Allerdings verändert sich in der betrachteten Aufgabenstellung die Belegung des Raumes zwischen den einzelnen Sensoraufnahmen auf unvorhersehbare Weise. Daher können Verfahren, wie sie in [Fujimura91] beschrieben werden, nicht angewendet werden.

Stand der Forschung

Da die Forschung im Bereich der kollisionsvermeidenden Bahnplanung einen hohen Reifegrad erreicht hat, existieren zusammenfassende Werke, die den Stand der Forschung systematisch beschreiben. Methoden zur Bahnplanung in statischer, bekannter Umgebung werden zum Beispiel in [Hwang92] und [Latombe96] vorgestellt. Methoden für dynamische Umgebungen, die allerdings ebenfalls vollständig bekannt sind, werden in [Fujimura91] untersucht.

[Hwang92] unterteilt Bahnplanungsverfahren in die Kategorien Skelettverfahren, Zellzerlegungsverfahren, Potentialfeldverfahren und Verfahren der mathematischen Programmierung. In [Latombe96] werden die Verfahren zur Bahnplanung in die Kategorien Roadmap, exakte und approximative Zellzerlegung und Potentialfeldverfahren unterteilt. Die Roadmapverfahren entsprechen dabei den Skelettverfahren aus [Hwang92].

Der Nachteil ist, dass keines der vorgestellten Verfahren die Besonderheiten der sensorbasierten Planung berücksichtigt.

Innerhalb von [Gupta98] werden drei Verfahren zur sensorbasierten Planung von verschiedenen Autoren vorgestellt. Zum einen die Verwendung von harmonischen Funktionen zur Bahnplanung ([Souccar98]), Echtzeitkollisionsvermeidung durch eine künstliche Haut ([Lumelsky98]) und die sensorbasierte Navigation mit einem mobilen Roboter ([Noborio98]). [Lumelsky98] und [Noborio98] gehen von auf dem Roboter angebrachten Sensoren aus, die nur einen Teil der Umwelt wahrnehmen und zum Ziel navigieren. Im realisierten System sind aber die Sensoren stationär befestigt und es soll eine globale Kollisionsvermeidung realisiert werden. Wie die harmonischen Funktionen aus [Souccar98] mit Hilfe von Kamerabildern erzeugt werden können, ist unklar.

Anforderungen an sensorbasierte Bahnplaner

Im Anwendungsszenario steht ein globaler Kollisionstest zur Verfügung. Die Umwelt kann sich zwischen den einzelnen Planungsiterationen auf beliebige Weise verändern. Allerdings kann erwartet werden, dass sich relativ wenige Objekte im Konfigurationsraum befinden. Dies entspricht dem Szenario „SIMPLE“ aus [Hwang92].

In der klassischen Bahnplanung kann dem Ergebnis der Kollisionserkennung absolut vertraut werden. Bei einem System, das die Umwelt mit Sensoren erfasst, ist dies nicht so, da in der Sensorerfassung immer Fehler auftreten können.

Diese Eigenschaften stellen besondere Anforderungen an die Bahnplanung.

Je länger das Bahnplanungsverfahren braucht, desto älter sind die Sensordaten, auf denen geplant wird. Da sich die Umwelt ständig verändern kann, sollte der Bahnplaner so schnell wie möglich eine Bahn finden. Der Hauptanteil an Rechenzeit bei Bahnplanungsverfahren wird im Kollisionstest verbraucht. Damit die Bahn auf möglichst neuen Sensordaten geplant wird, sollte der Bahnplaner möglichst wenig Kollisionstests erfordern.

Wenn keine Bahn zum Ziel gefunden werden kann, sollte dies möglichst schnell erkannt werden, damit neue Sensordaten angefordert werden können. Dies hat den Grund, dass unter Umständen der Raum zum aktuellen Zeitpunkt blockiert, aber im nächsten Mo-

ment schon wieder frei ist. Dies kann im realisierten System zum Beispiel dadurch geschehen, dass ein Mensch direkt vor einer der Kameras vorbeiläuft. Es wäre nicht sinnvoll, mehrere Sekunden lang vergeblich nach einer Bahn zu suchen.

Wenn keine Bahn zum Ziel möglich ist, sollte möglichst nahe zum Ziel gefahren werden. Dies hat zum einen den Grund, dass unter Umständen das Ziel nur wegen einer fehlerhaften Kollisionserkennung nicht erreicht wird, falls die Qualität der Kollisionserkennung abnimmt mit der Entfernung der getesteten Konfiguration zur aktuellen Konfiguration. Zum anderen kann das Hindernis, das den Weg zum Ziel blockiert, ein kooperierender Mensch sein. Nur wenn der Mensch erkennt, dass er dem Roboter im Weg ist, wird er den Weg freimachen.

Damit Fehler in der Kollisionserkennung nicht sofort zu Kollisionen mit Hindernissen führen, sollte die gefundene Bahn nicht nahe an Hindernissen vorbeiführen. Am besten wäre eine Bahn in der Mitte des freien Raumes.

Bei der sicheren Mensch-Roboter-Koexistenz ist von Bedeutung, dass der Mensch das Verhalten des Roboters einschätzen kann. Unerwartetes Verhalten führt leichter zu Gefahrensituationen als erwartetes Verhalten. Angewendet auf die Bahnplanung bedeutet das, dass die vom Roboter ausgeführte Bahn für den Menschen intuitiv sein sollte.

In der klassischen Bahnplanung wird angestrebt, eine möglichst kurze Bahn zu finden. Wenn die Dynamik des Roboters berücksichtigt wird, kommt noch der Aspekt der „Glätte“ hinzu. Eine glatte Bahn erfordert weniger Beschleunigungskräfte und kann daher im allgemeinen schneller abgefahren werden.

Ausgehend von den obigen Überlegungen kann folgendes Anforderungsprofil an eine Bahnplanung für sensorbasierte Planung erstellt werden (Anordnung nach fallender Priorität):

1. Der Planer sollte möglichst wenig Kollisionstest erfordern.
2. Der Planer sollte schnell terminieren und eine Fehlerbedingung melden, wenn keine Bahn gefunden wird.
3. Die Bahn sollte möglichst kurz und glatt sein.
4. Die Bahn sollte für den Menschen vorhersehbar sein.
5. Die Bahn sollte nicht direkt an Hindernissen vorbei führen, sondern in der Mitte des freien Raumes verlaufen.
6. Der Planer sollte eine Teilbahn liefern, wenn der Weg zum Ziel blockiert ist.

Verwendete Bahnplaner

Damit die Vorteile der globalen Kollisionserkennung ausgenutzt werden können, ist ein globales Bahnplanungsverfahren notwendig. Bei lokalen Verfahren kann der Roboter in lokalen Minima festsitzen, obwohl es einen freien Weg zum Ziel gibt.

Es werden die Bahnplanungsalgorithmen Best-First und HD-I, ein Randomized-Roadmap-Planer sowie ein Wellenausbreitungsverfahren untersucht. Der Bahnplanungsalgorithmus HD-I und der verwendete Best-First-Algorithmus stammt aus [Noborio99]. Der Best-First realisiert eine Tiefensuche. Als Randomized-Roadmap-Planer wurde das Verfahren aus [Yu99] verwendet.

Bewertung der Bahnplaner

Eine Analyse der Anforderungen zeigt einige Probleme auf:

Die Anforderungen Nr. 1 und Nr. 5 sind nicht gleichzeitig erfüllbar. Die Forderung nach einer Bahn, möglichst in der Mitte des freien Raumes, verlangt, dass der Bahnplaner prinzipielles Wissen über alle Hindernisse im Raum hat. Dieses Wissen kann er aber nur über eine große Anzahl an Kollisionstests erlangen, was im Widerspruch zur Forderung nach einer kleinen Anzahl von Kollisionstests ist.¹¹

Außerdem stehen die Anforderung Nr.3 und Nr.5 im Widerspruch zueinander. Wenn ein Hindernis den direkten Weg blockiert, führt die kürzeste Bahn immer direkt am Hindernis vorbei. Eine Bahn, die der Forderung genügt, in der Mitte des freien Raumes zu verlaufen, kann daher nicht die kürzeste Bahn sein.

In der Praxis wird man sich deshalb mit einem Planer begnügen, der einen guten Kompromiss dieser Kriterien darstellt. Zum Beispiel mit einem Bahnplanungsverfahren, das mit wenigen Kollisionstests eine kurze Bahn erzeugt und das einen Mindestabstand zu Hindernissen einhält.

Die Anzahl der Kollisionstests wächst bei allen Bahnplanungsverfahren, außer dem Randomized-Roadmap, exponentiell mit der Anzahl der Dimensionen des Konfigurationsraumes (bei gleicher Diskretisierung aller Freiheitsgrade).

Bei den Randomized-Roadmap-Planern gibt es keine „natürliche“ Terminierung, wenn kein Pfad gefunden wird. Die anderen Planer können je nach Szenario ziemlich lang brauchen, bis sie erkennen, dass kein Weg möglich ist.

Wenn kein Weg zum Ziel existiert, kann aus den Datenstrukturen von Best-First, HD-I und Randomized-Roadmap zumindest ein Zwischenziel und eine kollisionsfreie Teilbahn zu diesem Zwischenziel abgeleitet werden. Beim Wellenausbreitungsverfahren kann erst dann eine sinnvolle Bahn generiert werden, wenn die Wellenfront Start und Ziel verbindet.

Wenn Hindernisse den direkten Weg versperren, generieren alle Bahnplaner, außer dem Randomized-Roadmap, Bahnen, die direkt an den Hindernissen vorbeiführen.

Die Bahn des Randomized-Roadmap ist zufällig und daher für einen Menschen nicht vorhersehbar. Die Bahn des Wellenausbreitungsverfahrens ist relativ gut vorhersehbar, da sie immer die kürzeste Verbindung von Start und Ziel darstellt.

Alle Planer arbeiten im Konfigurationsraum des Roboters mit diskreten Konfigurationen. Der Abstand zwischen Konfigurationen wird durch die Anzahl der dazwischenliegenden Konfigurationen bestimmt. Dadurch entspricht der Abstand im Konfigurationsraum im allgemeinen nicht dem Abstand im kartesischen Raum. Während der Mensch die Bahn im kartesischen Raum wahrnimmt, planen die Bahnplanungsverfahren im Konfigurationsraum. Dadurch kann, abhängig von der Diskretisierung des Konfigurationsraumes, eine wenig intuitive Bahn entstehen.

Die Bahn des Wellenausbreitungsverfahrens ist so kurz wie möglich und enthält keine überflüssigen Krümmungen. Die Bahnen von Best-First, HD-I sind unter Umständen etwas länger. Am schlechtesten ist die Bahn des Randomized-Roadmap. Sie kann deutlich länger sein und spitzere Winkel haben als notwendig.

¹¹ Bei der Verwendung einer Abstandsberechnung kann der Bahnplaner mit wenigen Anfragen die notwendigen Informationen zur Ermittlung einer Bahn in der Mitte des freien Raumes bekommen. Allerdings ist der Aufwand lediglich in die Abstandsberechnung verlagert und nicht wirklich eingespart.

Die obige Bewertung ist zusammengefasst in der Tabelle 7 zu sehen.

Kriterium	Bahnplaner			
	Wellenausbreitung	Best-First (Tiefensuche)	HD-I	Randomized-Roadmap
Anzahl der Kollisionstests	-	-	-	+
Schnelle Terminierung, wenn kein Weg vorhanden	-	o	o	-
Erzeugung von Teilbahn, wenn kein Weg vorhanden	-	o	o	o
Bahn in Mitte des Freiraumes	-	-	-	o
Vorsehbare und intuitive Bahn	+	o	o	-
Kurze und glatte Bahn	+	o	o	-
Gesamtbewertung	-2	-2	-2	-2

Tabelle 7: *Bewertung der betrachteten Bahnplanungsverfahren*

Ergebnis

Es zeigt sich, dass die Bewertung von allen Verfahren gleich ist. Aus diesem Grund werden alle Verfahren realisiert und im Experiment getestet. Außerdem ist keines der betrachteten Verfahren optimal für die sensorbasierte Planung geeignet. Daher wird versucht, die Eignung der Bahnplaner durch leichte Veränderungen zu verbessern.

3.4 Schlussfolgerungen

Aus den obigen Ausführungen wird die Wahl der Komponenten für ein System zur Sicherung der Transferbewegung eines Roboters abgeleitet. Die Systemkomponenten werden folgendermaßen realisiert:

- Die externe Sensorik besteht aus mehreren stationär befestigten, einfachen Kameras.
- Die Umweltmodellierung erfolgt durch ein Gitter im Konfigurationsraum. Allerdings erfolgt die Modellierung innerhalb der Bahnplanung und kann daher mit dem verwendeten Bahnplanungsverfahren variieren.
- Die Kollisionsvermeidung wird durch ein globales Bahnplanungsverfahren im Konfigurationsraum des Roboters realisiert. Hierzu wird auf Standardverfahren zurückgegriffen.
- Die Kollisionserkennung wird durch einen globalen Kollisionstest auf den Kamerabildern realisiert.
- Die Robotermodellierung wird durch eine Polygonapproximation realisiert, im Detail aber den Erfordernissen des Kollisionstests angepasst.

3.4.1 Systemkonzept

Die Bahnplanung hat die Aufgabe, einen kollisionsfreien Weg zum Ziel zu finden. Hierzu baut sie sich mit Hilfe des Kollisionstests eine Repräsentation des Belegungszustandes vom Konfigurationsraum des Roboters auf. Die Bahnplanung übergibt hierzu dem Kollisionstest eine Testkonfiguration. Der Kollisionstest meldet zurück, ob diese Testkonfiguration kollisionsfrei ist. Der Kollisionstest verwendet das Robotermodell, die von der Bildverarbeitung aufbereiteten Kamerabilder und die aktuellen Gelenkstellungen des Roboters, um den Belegungszustand der Testkonfiguration zu ermitteln. Diese Systemstruktur ist in Abbildung 25 dargestellt.

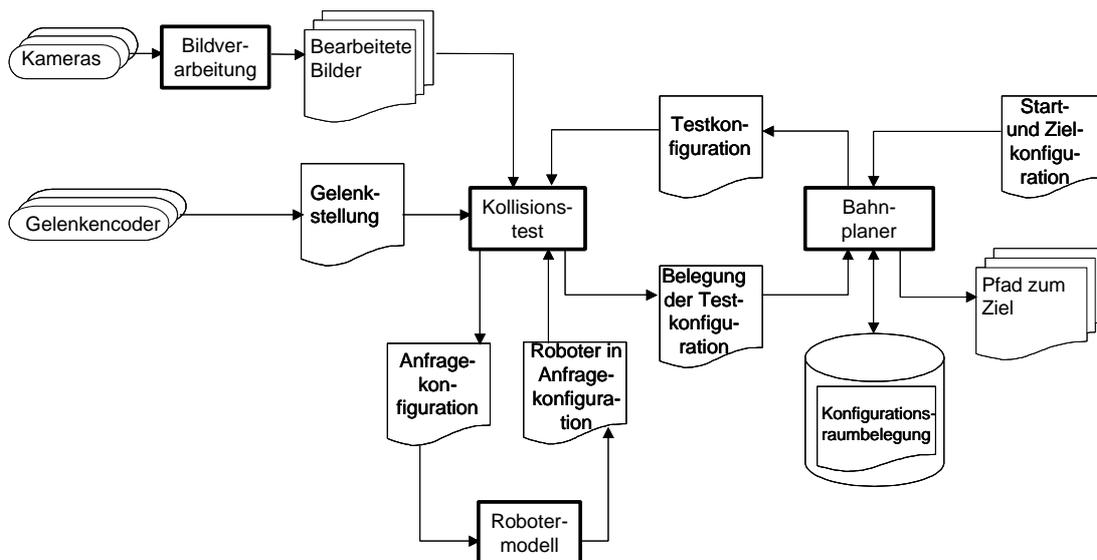


Abbildung 25: Skizze der Systemstruktur. Der Bahnplaner plant einen kollisionsfreien Pfad vom Start zum Ziel und verwendet hierzu eine Repräsentation des Belegungszustandes des Konfigurationsraumes des Roboters. Die Bahnplanung erhält die Information über den Belegungszustand einer Testkonfiguration vom Kollisionstest. Der Kollisionstest verwendet das Robotermodell, die von der Bildverarbeitung aufbereiteten Kamerabilder und die aktuelle Gelenkstellung des Roboters, um den Belegungszustand der Testkonfiguration zu ermitteln.

Ausgehend von den obigen Ausführungen werden folgende Annahmen über die Roboterzelle und deren Umwelt getroffen:

- Mehrere Kameras sind stationär an Halterungen oder Wänden befestigt.
- Die Kameras sind in Bezug auf das Weltkoordinatensystem kalibriert.
- Der überwachte Arbeitsraum ist vollständig im Sichtbereich aller Kameras.
- Die Beleuchtung ist diffus.
- Roboter und Vordergrundobjekte sind vom Hintergrund unterscheidbar.
- Die aktuelle Gelenkstellung des Roboters ist bekannt.
- Die Position mindestens eines Referenzpunktes des Roboters in Bezug auf das Weltkoordinatensystem ist bekannt.
- Die Robotergeometrie ist bekannt.

Eine solche Zelle ist schematisch in Abbildung 26 dargestellt. Die realisierte Versuchszelle ist in Abbildung 27 zu sehen.

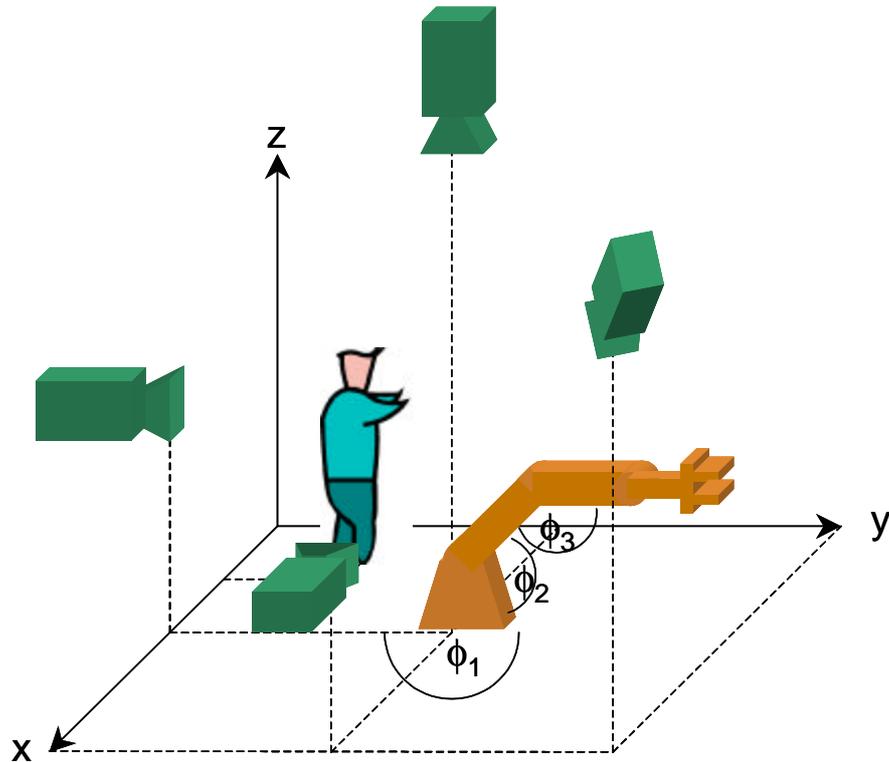


Abbildung 26: Skizze einer Roboterzelle zur sicheren Mensch-Roboter-Koexistenz. Der gemeinsame Arbeitsraum von Mensch und Roboter befindet sich vollständig im Sichtbereich von allen vier stationären Kameras. Der Roboter ist in drei rotatorischen Gelenken (f_1 , f_2 , f_3) beweglich.



Abbildung 27: Foto der realisierten Zelle zur sicheren Mensch-Roboter-Koexistenz. Vier Graubild-Kameras sind in der dargestellten Kamerakonfiguration in den oberen Ecken installiert und überwachen den gesamten, verwendeten Arbeitsraum des Roboters.

3.4.2 Bildbasierter Ganzarmkollisionstest

Zentrale Knotenstelle im System zur Sicherung der Transferbewegung ist der Kollisionstest. Einerseits muss der Kollisionstest den Anforderungen der Erkennungs- und Reaktionsstrategie genügen, im betrachteten Fall also eine globale Ausweichbewegung auf der Basis einer Ganzarmerkennung ermöglichen. Andererseits muss er in der Lage sein, die dazu notwendigen Informationen aus den Daten der externen und internen Sensoren zu gewinnen.

Im folgenden wird beschrieben, wie eine globale, bildbasierte Ganzarmkollisionserkennung realisiert werden kann.

Die Aufgabe des Kollisionstests ist es, zu entscheiden, ob es zu einer Kollision mit einem Objekt käme, wenn der Roboter in einer zu testenden Konfiguration wäre. Dabei wird lediglich diese angefragte Testkonfiguration betrachtet. Das Ergebnis ist unabhängig davon, ob es möglich ist, die Testkonfiguration aus der gegenwärtigen Konfiguration des Roboters kollisionsfrei zu erreichen.

Grundsätzlich kann der Kollisionstest auf verschiedene Arten durchgeführt werden. Aus den Kamerabildern wird eine Belegung des Planungsraumes der Kollisionsvermeidung abgeleitet. Falls der Roboter in den Kamerabildern auftaucht, muss sein Effekt auf die Belegung des Planungsraumes herausgerechnet werden. Dann wird geprüft, ob der Roboter in der zu testenden Konfiguration mit einem Objekt des Planungsraumes kollidiert.

Damit eine globale Planung der Ausweichbewegung mit konventionellen Bahnplanungsverfahren möglich ist, muss letztendlich der freie Konfigurationsraum des Roboters aus den Kamerabildern rekonstruiert werden können. Damit wird das Ausgangsproblem der Mensch-Roboter-Koexistenz unter den gegebenen Einschränkungen konkretisiert zu der Frage:

Wie kann aus den Bildern einer oder mehrerer Kameras geschlossen werden, ob irgendein Teil des Roboters in einer beliebigen, vorgegebenen Konfiguration mit einem Hindernis kollidiert?

3.4.3 Vorgehen

Damit das Problem des Kollisionstests leichter gelöst werden kann, wird es in zwei Teilprobleme zerlegt. Zuerst werden aus den Kamerabildern binäre Vordergrund-Hintergrund-Bilder erzeugt, im folgenden *Silhouetten* genannt. Anschließend wird aus diesen Silhouetten auf die Belegung des Konfigurationsraumes geschlossen. Dieser Ablauf ist in Abbildung 28 dargestellt.

Die Vorteile dieses Ansatzes sind, dass Bildverarbeitung und die eigentliche Kollisionserkennung voneinander unabhängig sind. Durch die Verwendung von Silhouetten müssen nur Bildbereiche auf Veränderungen untersucht werden. Es ist keine aufwendige Objekterkennung notwendig. Dies schränkt die zulässigen Roboter und Hindernisse nicht ein. Sie müssen jedoch vom Hintergrund unterscheidbar sein.

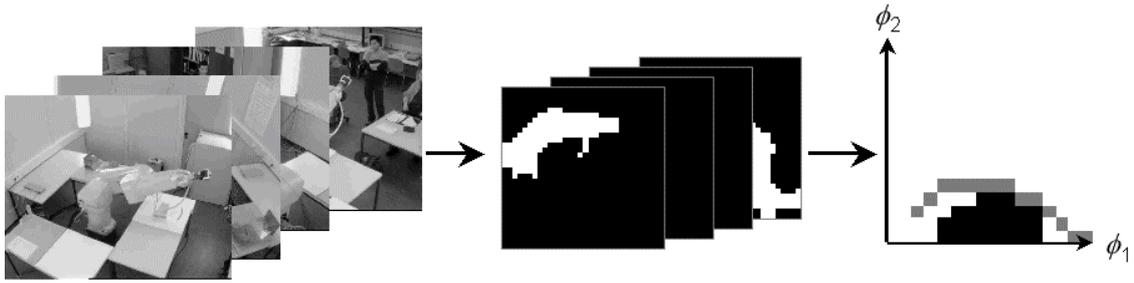


Abbildung 28: Aufspaltung des Kollisionstests. Zuerst werden aus den Kamerabildern (links) Silhouetten (Mitte) erzeugt. Anschließend erfolgt die Kollisionserkennung auf Silhouetten. Dadurch kann die Belegung des Konfigurationsraums (rechts, hier mit zwei Dimensionen ϕ_1 und ϕ_2 und eingezeichnete Roboterbahn) rekonstruiert werden.

Konsequenz dieses Ansatzes ist, dass nur zusätzliche Hindernisse im Vergleich zu einem als leer angenommenen Referenzzustand betrachtet werden können. Hindernisse, die im Referenzzustand vorhanden sind, können durch andere Maßnahmen berücksichtigt werden. Zum Beispiel wird im realisierten System durch eine Abstandsberechnung sichergestellt, dass kein Gelenk des Roboters mit Tischen im Arbeitsraum kollidieren kann.

Nachteilig ist, dass Roboter und Hindernisse in den Silhouetten nicht unterscheidbar sind. Zur Unterscheidung muss zusätzliches Wissen in der Form eines Robotermodells verfügbar sein.

Ein weiterer Nachteil des Ansatzes ist, dass der Roboter die Hindernisse leichter verdecken kann. Eine Verdeckung kann nicht nur auftreten, wenn das Hindernis aus Sicht der Kamera hinter dem Roboter ist, sondern auch dann, wenn es vor dem Roboter ist. In beiden Fällen ist das entsprechende Pixel in der Silhouette gesetzt.

3.4.4 Mathematische Formulierung des Problems

Gegeben sei ein Roboter R mit n Gelenken $1, \dots, n$. Die Stellung des i -ten Gelenkes wird mit f_i bezeichnet. Eine Konfiguration c des Roboters wird durch das n -Tupel (f_1, \dots, f_n) der zugehörigen Gelenkstellungen beschrieben.

Die drei Achsen x, y, z des Arbeitsraumes A bilden ein rechtshändiges, kartesisches Koordinatensystem. Die Achsen sind in einzelne Intervalle ${}_x I_i$ bzw. ${}_y I_i, {}_z I_i$ unterteilt. Intervalle ${}_x, {}_y, {}_z I_i$ sind vom Ursprung ausgehend, mit 0 beginnend, durchnummeriert.

Die Intervalle bilden ein 3-dimensionales Gitter aus Volumenelementen („Voxel“). Die Koordinate eines Voxels v wird durch das Tripel der zugehörigen Intervallnummern beschrieben.

Ein Punkt P des Arbeitsraumes A ist im Voxel $v(a, b, c)$ enthalten, wenn die Koordinaten von P innerhalb der Intervalle, die v begrenzen, liegen.

$$P \in A, v \subset A : P(x, y, z) \in v(a, b, c) \Leftrightarrow x \in {}_x I_a \wedge y \in {}_y I_b \wedge z \in {}_z I_c \quad (3)$$

Die Menge aller Voxel bildet den Voxelraum V :

$$V = \{ v \mid v \subset A \} \quad (4)$$

$Z_R(v)$ und $Z_O(v)$ sind binäre Funktionen, die jedem Voxel einen der Zustände aus $\{0,1\}$ zuordnen. Die Zustandsfunktion $Z_R(v)$ des Voxel $v(x,y,z)$ ist genau dann 1, wenn das Voxel v Teil des Roboters ist. Die Zustandsfunktion $Z_O(v)$ des Voxel $v(x,y,z)$ ist genau dann 1, wenn das Voxel v Teil eines Objektes ist.

Der vom Roboter eingenommene Raum V_R ist die Menge aller Voxel v , deren Zustandsfunktion $Z_R(v)=1$ ist:

$$\text{Roboterraum: } V_R = \{ v \in V \mid Z_R(v) = 1 \} \quad (5)$$

Analog lassen sich die weiteren Teilmengen des Voxelraumes V definieren

$$\text{(kartesischer) Objektraum: } V_O = \{ v \in V \mid Z_O(v) = 1 \} \quad (6)$$

$$\text{(kartesischer) Freiraum: } V_F = \{ v \in V \mid Z_R(v) = Z_O(v) = 0 \} \quad (7)$$

$$\text{(kartesischer) Belegtraum: } V_B = \{ v \in V \mid (Z_R(v) = 1) \vee (Z_O(v) = 1) \} \quad (8)$$

Der Roboterraum V_R kann als Funktion der Roboterkonfiguration c betrachtet werden. Diese Funktion heißt im folgenden *geometrisches Modell* M_G des Roboters. Über die Eigenschaften des geometrischen Modells werden vorerst keine Annahmen getroffen.

$$V_R(c) = M_G(R, c) \quad (9)$$

Der Konfigurationsraum C ist die Menge aller zulässigen Konfigurationen c des Roboters R .

Der freie Konfigurationsraum C_F ist die Menge aller Konfigurationen, die zu keiner Kollision des Roboters mit einem Objekt führen.

$$C_F = \{ c \in C \mid V_R(c) \cap V_O = \emptyset \} \quad (10)$$

Der durch Objekte belegte Konfigurationsraum C_O ist die Menge aller Konfigurationen c , die zu einer Kollision des Roboters mit mindestens einem Objekt führen.

$$C_O = \{ c \in C \mid V_R(c) \cap V_O \neq \emptyset \} \quad (11)$$

$$\text{Tertium non datur: } C = C_F \cup C_O \quad (12)$$

Aus den Kamerabildern K_k werden mit Hilfe der Referenzbilder K_k^R binäre Bilder B_k berechnet und diese werden zur Kollisionserkennung verwendet. Die Pixel p der Binärbilder können nur Werte aus $\{0, 1\}$ annehmen. Die Anzahl der Pixel der Binärbilder muss nicht gleich der Anzahl der Pixel der zugrunde liegenden Kamerabilder sein. Im Folgenden sind mit Bildern immer Binärbilder bezeichnet.

Gegeben seien l Kameras $0, \dots, (l-1)$. Die Bildebene der k -ten Kamera wird mit E_k bezeichnet. Die zwei Achsen u, v der Bildebene bilden ein kartesisches Koordinatensystem. Die Achsen sind in einzelne Intervalle ${}_u I_i$ bzw. ${}_v I_i$ unterteilt. Die Intervalle ${}_u, {}_v I_i$ sind vom Ursprung ausgehend, mit 0 beginnend, durchnummeriert.

Die Intervalle bilden ein 2-dimensionales Gitter aus Bildelementen („Pixel“). Die Koordinate eines Pixels p wird durch das Paar der zugehörigen Intervallnummern beschrie-

ben. Ein Punkt P der Bildebene E ist im Pixel p enthalten, wenn die Koordinaten von P innerhalb der Intervalle, die p begrenzen, liegen.

$$P \in E, p \subset E : P(u, v) \in p(d, e) \Leftrightarrow u \in {}_u I_d \wedge v \in {}_v I_e \quad (13)$$

Das Bild B_k der Kamera k ist die Menge aller Pixel von Kamera k :

$$B_k = \{ p \mid p \subset E_k \} \quad (14)$$

Die Aufgabe des Kollisionstests KT ist zu entscheiden, ob eine zu testende Konfiguration c_T Element von C_F ist, wenn der Roboter gerade in einer anderen Konfiguration c_R ist und von der Szene die Binärbilder $B_0 .. B_{l-1}$ vorliegen:

$$\text{KT}(c_T, c_R, B_0, \dots, B_{l-1}) = \begin{cases} 1; & c_T \in C_F \\ 0; & \text{sonst} \end{cases} \quad (15)$$

Die notwendigen Binärbilder werden durch die Bildverarbeitung BV aus den aktuellen Kamerabildern und Referenzbildern erzeugt:

$$B_k = \text{BV}(K_k, K_k^R) \quad (16)$$

Kapitel 4

Silhouettenerzeugung von Szene und

Roboter

Der silhouettenbasierte Kollisionstest erwartet als Eingaben Silhouetten der aktuellen Szene sowie des Roboters in der aktuellen und der zu testenden Konfiguration. Eine Silhouette ist ein binäres Bild, in dem alle Vordergrundpixel gesetzt und alle Hintergrundpixel gelöscht sind. Im folgenden wird beschrieben, wie diese Silhouetten von Szene und Roboter erzeugt werden.

4.1 Erzeugung von Szenensilhouetten aus Kamerabildern

Die Aufgabe der Bildverarbeitung ist, aus den Kamerabildern mit Hilfe von Referenzbildern ein Silhouettenbild der aktuellen Szene zu erstellen. Beispiele für die Eingaben und die erwartete Ausgabe der Bildverarbeitung sind in Abbildung 29 zu sehen.



Abbildung 29: *Problemstellung der Silhouettenerzeugung. Aus Referenzbild (links) und aktuellem Szenenbild (Mitte) ist die Silhouette (rechts) zu berechnen. Die Beispielsilhouette basiert auf einer manuellen Klassifikation) (nach [Heinzen03]).*

Die robuste und schnelle Erstellung eines Silhouettenbildes aus Grauwertbildern ist ein relativ schwieriges Problem. Der Grund hierfür ist der eindimensionale Farbraum der Kameras. Das zu messende Signal (echter Grauwert eines Szenenpunktes) ist direkt mit der Störgröße (z.B. Schatten) überlagert. Im Gegensatz dazu kann bei RGB-Farbkameras eine Transformation in einen Farbraum vorgenommen werden, bei der Signal und Störung sich unterschiedlich auf die verschiedenen Dimensionen verteilt. Ein Beispiel hierfür ist [Horprasert00]. Dort wird der Unterschied eines Pixels im Ver-

gleich zum Referenzpixel durch den Abstand im RGB-Raum zwischen dem gemessenen Pixel und der Gerade bestimmt, die durch den Ursprung und das Referenzpixel aufgespannt wird. Es wird angenommen, dass sich Unterschiede in der Beleuchtung hauptsächlich in einer Verschiebung des Farbwertes entlang dieser Geraden äußern. Anhand des Abstandes wird ein Pixel als unveränderter Hintergrund, abgeschatteter Hintergrund, beleuchteter Hintergrund oder als Vordergrund klassifiziert. Auf Grund des eindimensionalen Farbraums von Grauwertkameras kann diese Methode nicht auf Grauwertpixel angewendet werden.

Der Nachteil von Farbkameras liegt darin, dass das Datenvolumen dreimal so groß ist wie bei Graubildkameras gleicher Auflösung.¹² Da zur Sicherung der Transferbewegung mehrere Kameras benötigt werden, wird bei der Verwendung von Farbkameras deutlich aufwendigere Bildverarbeitungshardware notwendig. Aus diesem Grund wird im folgenden von Grauwertbildern ausgegangen. Prinzipiell sind aber alle aufgezeigten Methoden auch auf Farbbilder anwendbar.

4.1.1 Stand der Forschung

In [Rosin98] wird ein Verfahren zur Ermittlung eines Schwellwertes zur Binarisierung von Grauwertbildern vorgestellt. Es basiert auf der Veränderung der Anzahl der erkannten Regionen in Abhängigkeit des eingestellten Schwellwertes. Anstatt die Anzahl der Regionen direkt zu zählen, wird die Eulerzahl des Bildes verwendet. Die Eulerzahl eines Binärbildes ist definiert als die Anzahl der Objekte in diesem Bild minus der Anzahl der Löcher in den Objekten. Solange der Schwellwert so klein ist, dass auch verrauschte Hintergrundpixel als Vordergrund klassifiziert werden, ändert sich die Eulerzahl stark, wenn der Schwellwert verändert wird. Wird der Schwellwert so groß, dass eine gute Trennung zwischen Vordergrund und Hintergrund vorhanden ist, ändert sich der Schwellwert nicht mehr so stark. Das vorgestellte Verfahren eignet sich zur Ermittlung eines guten Schwellwertes, solange das Hauptproblem das Pixelrauschen ist. Im realisierten System spielt allerdings das Pixelrauschen keine große Rolle, das Problem ist viel mehr der Schattenwurf von Roboter und Hindernissen. Zur Unterscheidung von realem Objekt und Schatten ist das Verfahren aber nicht in der Lage.

Ein Verfahren zur Kompensation von Schatten bei Grauwertkameras wird in [Rosin95] vorgestellt. Nach der Segmentierung werden die im Bild gefundenen Regionen untersucht, ob sie echter Vordergrund oder Schatten sind. Schatten wird an Hand von Regeln erkannt. Schatten sollte dunkler als das Referenzbild sein und der Grad der Verdunkelung sollte für den gesamten Schatten ähnlich sein. Dieses Verfahren kann bei Außenszenen relativ gut funktionieren, da die Umgebung und die Objekte stark texturiert sind. In industriellen Umgebungen sind Umwelt und Objekte häufig einheitlich gefärbt. Dies kann dazu führen, dass Objekte als Schatten erkannt werden.

In [Ivanov00] wird ein Verfahren vorgestellt, das die Grauwertbilder von mehreren Kameras berücksichtigt. Zu Beginn wird in der leeren Szene eine „Disparity-Map“ erstellt, die jedem Pixel einer Kamera das korrespondierende Pixel der anderen Kameras zuordnet. In der leeren Szene sollten die Werte aller Pixel den gleichen Betrag haben.

¹² Aus Anwendersicht besteht jedes Pixel aus einer Rot-, Grün- und Blau-Komponente. Die Hardwarerealisierung kann allerdings anders aussehen. Bei Ein-Chip-CCD-Kameras gibt es zum Beispiel meistens deutlich mehr Grün-Pixel als Blau-Pixel.

Im Betrieb werden dann die Unterschiede der Pixel ausgewertet. Stehen mehr als zwei Kameras zur Verfügung, können die Probleme durch Verdeckungen reduziert werden.

Die Annahme, dass in der leeren Szene alle Kameras den gleichen Grauwert an der selben Stelle sehen ist nur unter einschränkenden Annahmen zutreffend. Je nach Kamerateyp, Einstellung von Blende, Shutter, Gain und Zoom können verschiedene Kameras für die gleiche Stelle der Szene unterschiedliche Grauwerte liefern. Dies ließe sich durch eine aufwendige Kalibrierung umgehen. Allerdings bleibt das Problem, dass Kameras an unterschiedlichen Positionen zum Beispiel durch Reflektionen andere Grauwerte sehen. Ein Problem, das nicht betrachtet wurde, ist das Vorhandensein von Objekten im Zustand der leeren Szene, die deutlich vor dem Hintergrund stehen. Dies tritt besonders dann auf, wenn die Kameras einen großen Basisabstand haben. Sind solche Objekte vorhanden, gibt es für einige Pixel keine Korrespondenzen in den anderen Kameras. Ein weiteres Problem ist die Erstellung der Disparity-Map. Wenn ein komplettes CAD-Modell der Szene und der Kameras vorhanden ist, kann sie berechnet werden. Andernfalls muss sie aus den Bildern der leeren Szene erstellt werden. Dabei stellen große, uniforme Flächen ein Problem dar, da die Korrespondenz nicht sicher gelöst werden kann.

In der Anwendung zur Sicherung der Transferbewegung muss mit Objekten in der leeren Szene gerechnet werden und der Basisabstand der Kameras ist relativ groß. Dies könnten Tische, Maschinen oder ähnliches sein. Außerdem sind industrielle Umgebungen üblicherweise einheitlich gefärbt und nur schwach texturiert. Aus allen diesen Gründen ist die Methode von [Ivanov00] in der gegebenen Aufgabenstellung nicht anwendbar.

4.1.2 Bewertung der Silhouettenerzeugung

Zur Bewertung der Qualität der erzeugten Silhouette wird eine typische Szene von Hand in Vordergrund und Hintergrund segmentiert. Der Roboter ist Vordergrundobjekt, braucht aber nicht erkannt zu werden, da er im Kollisionstest über das Robotermodell herausgerechnet wird. Damit der Roboter die Erkennung der Hindernisse nicht beeinflusst, wird er ausmaskiert. Abbildung 30 zeigt die Kamerabilder des Referenzzustandes und der Szene, sowie die von Hand erstellte Maske des Roboters und die handsegmentierte Silhouette.



Abbildung 30: *Obere Reihe: Kamerabilder von leerer Zelle (links) und Zelle mit Hindernissen (rechts)*
Untere Reihe: Maske zur Ausmaskierung des Roboters (links) von Hand segmentiertes Bild (rechts) (nach [Heinzen03])

Um die Güte der Silhouettenerzeugung zu messen, werden die berechneten Silhouetten mit den von Hand klassifizierten Silhouetten verglichen. Damit auch unterschiedliche Kachelgrößen beurteilt werden können und Kacheln, die nur zum Teil aus Vordergrund bestehen, mit in die Bewertung einfließen können, wird die Qualität an Hand der Anzahl der korrekt klassifizierten Pixel des Grauwertbildes bestimmt ([Heinzen03]).

Dazu wird nach der Silhouettenerzeugung die Anzahl der korrekt und falsch klassifizierten Pixel in Abhängigkeit von der tatsächlichen Klassenzugehörigkeit bestimmt. Diese vier Anzahlen sind in Tabelle 8 definiert.

	Als Vordergrund klassifiziert	Als Hintergrund klassifiziert
Tatsächlicher Vordergrund	KP (korrekt positiv)	FN (falsch negativ)
Tatsächlicher Hintergrund	FP (falsch positiv)	KN (korrekt negativ)

Tabelle 8: Bewertung der Klassifikationsgüte

Gleichung 17 gibt den Anteil korrekt klassifizierter Pixel a_{kor} an:

$$a'_{kor} = \frac{KP + KN}{KP + FN + KN + FP} \quad (17)$$

Neben dem Anteil der korrekt klassifizierten Pixel sind auch andere Bewertungskriterien denkbar. In [Heinzen03] wird zum Beispiel auch die Fläche unter der ROC-Kurve (Receiver Operating Characteristics) als Kriterium untersucht.

4.1.3 Ansatz

Der silhouettenbasierte Kollisionstest benötigt keine hochauflösenden Silhouetten. Aus Rechenzeitgründen ist für den Kollisionstest eine Silhouette mit relativ grober Auflösung vorteilhaft, solange darunter die Güte der Silhouettenqualität nicht leidet. Daher wird ein Ansatz verwendet, bei dem jedes Pixel der Silhouette aus mehreren Grauwertpixeln berechnet wird ([Ebert01]). Hierzu wird das Grauwertbild in Kacheln unterteilt. Für jede Kachel werden Merkmale berechnet. Dieses Prinzip ist in Abbildung 31 beispielhaft dargestellt.



Abbildung 31: Kachelung des Grauwertbildes zur Merkmalsberechnung. Grauwertbild mit eingezeichneten Kachelgrenzen (links), Merkmalsbild aus zwei Merkmalen, die durch zwei Grauwertbilder dargestellt werden (Mitte und rechts). ([Ebert01])

Die Merkmale einer Kachel der aktuellen Szene werden mit den Merkmalen der gleichen Kachel in einem Referenzbild verglichen. Aus diesem Vergleich wird geschlossen,

ob sich die entsprechende Kachel verändert hat oder nicht. Eine veränderte Kachel wird als Vordergrund, eine unveränderte als Hintergrund interpretiert. Bei der Klassifikation können auch die Merkmalsdifferenzen der benachbarten Kacheln mit einbezogen werden. Ebenso ist eine Nachbearbeitung der Differenzbilder durch morphologische Operatoren möglich. Der Datenfluss der Bildverarbeitung ist in Abbildung 32 zu sehen.

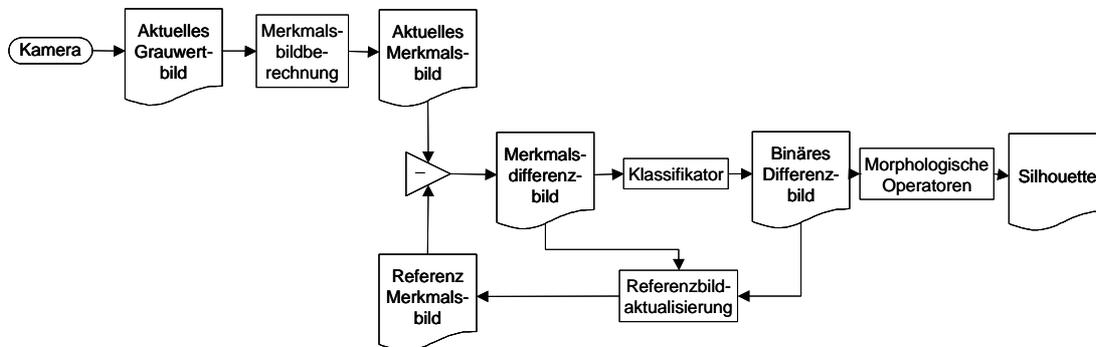


Abbildung 32: Datenfluss in der Bildverarbeitung bis zur Silhouettenerzeugung. Die aktuellen Grauwertbilder der Kameras werden in Kacheln unterteilt. In der Merkmalsbildberechnung wird für jede Kachel ein Satz von Merkmalen berechnet, die zusammen das Merkmalsbild ergeben. Die Differenz zwischen dem aktuellen Merkmalsbild und dem Referenzmerkmalsbild wird durch den Klassifikator zu einem binären Differenzbild umgewandelt. Mit Hilfe des Differenzbildes kann das Referenzbild aktualisiert werden. Das binäre Differenzbild kann durch morphologische Operatoren nachbearbeitet werden. Das Ergebnis ist die gesuchte Silhouette der aktuellen Szene

Die Bildverarbeitung gliedert sich in die Teile Merkmalsbilderstellung und Differenzbilderstellung sowie die Hilfstteile Referenzbilderstellung, Referenzbildaktualisierung und Kamerakalibrierung. Diese Teile werden im Folgenden vorgestellt.

4.1.4 Merkmalsbilderstellung

Nach der Aufnahme der Grauwertbilder werden die Merkmalsbilder erstellt. Zur Berechnung der Merkmale werden jeweils die Grauwertpixel einer Kachel verwendet. Die Merkmale können beliebig definiert werden, Voraussetzung ist lediglich, dass es eine Metrik gibt, die den Unterschied zwischen zwei Merkmalen als skalare Größe angibt. Im realisierten System sind aus Gründen der Einfachheit bereits alle Merkmale selbst skalare Größen und als Metrik wird die absolute Differenz der Merkmale verwendet.

Betrachtete Merkmale

Insgesamt wurden 14 verschiedene Merkmale für eine Kachel untersucht. Diese sind Mittlerer Grauwert, Varianz, Schiefe des Histogramms, Exzess des Histogramms, Häufigster Grauwert, Quantität des häufigsten Grauwertes, Median, Interquartilsabstand, Grauwertschwerpunkt in X-Richtung, Grauwertschwerpunkt in Y-Richtung, Kontrast, Bandbreite, Kantenerkennung und Skalarprodukt ([Diehl01], [Ebert01], [Heinzen03]).

Auswahl der geeigneten Merkmale

Die Verwendung von zu vielen Merkmalen hat eine Verschlechterung der Klassifikation zur Folge ([Duda73], [MarquesDeSá01]). Daher muss die Anzahl der verwendeten

Merkmale reduziert werden. Dies ist auch aus dem Grund sinnvoll, dass die zur Verfügung stehende Rechenzeit begrenzt ist.

Es existieren Verfahren, die auf der Bewertung einzelner Merkmale basieren ([MarquesDeSá01], [Abmayr94]). Diese Verfahren setzen allerdings eine statistische Unabhängigkeit der betrachteten Merkmale voraus. Da im betrachteten Fall alle Merkmale auf den gleichen Grauwertpixeln beruhen, ist diese Voraussetzung nicht erfüllt.

Die Merkmalsauswahl lässt sich auch durch die Qualität der aus den ausgewählten Merkmalen erzeugten Silhouetten bewerten. Die Qualität der tatsächlichen Klassifikation im Vergleich zu einer als ideal angenommenen Klassifikation ist ein Bewertungsmaß, das auch bei abhängigen Merkmalen und einer unbekanntenen Verteilung eine zuverlässige Beurteilung beliebiger Merkmalskombinationen zulässt.

Der Nachteil dabei ist, dass die Bewertung nur für jede Merkmalskombination und jeden Klassifikator einzeln berechnet werden muss. Bei der Verwendung von vielen Merkmalen und Klassifikatoren ist die Anzahl der zu untersuchenden Kombinationen sehr groß. Der Vorteil dabei ist, dass für jeden Klassifikator die optimale Merkmalsauswahl gefunden wird.

In einem Versuch wurden für alle möglichen Merkmalskombinationen ($2^{14} - 1 = 16383$) die Güte der erzeugten Silhouetten bei der Verwendung des linearen Klassifikators untersucht. Innerhalb der nach Qualität sortierten Liste der Merkmalskombinationen wurde für jedes Merkmal ermittelt, wie häufig es unter den besten n Merkmalskombinationen auftritt. Das Ergebnis ist in Abbildung 33 zu sehen.

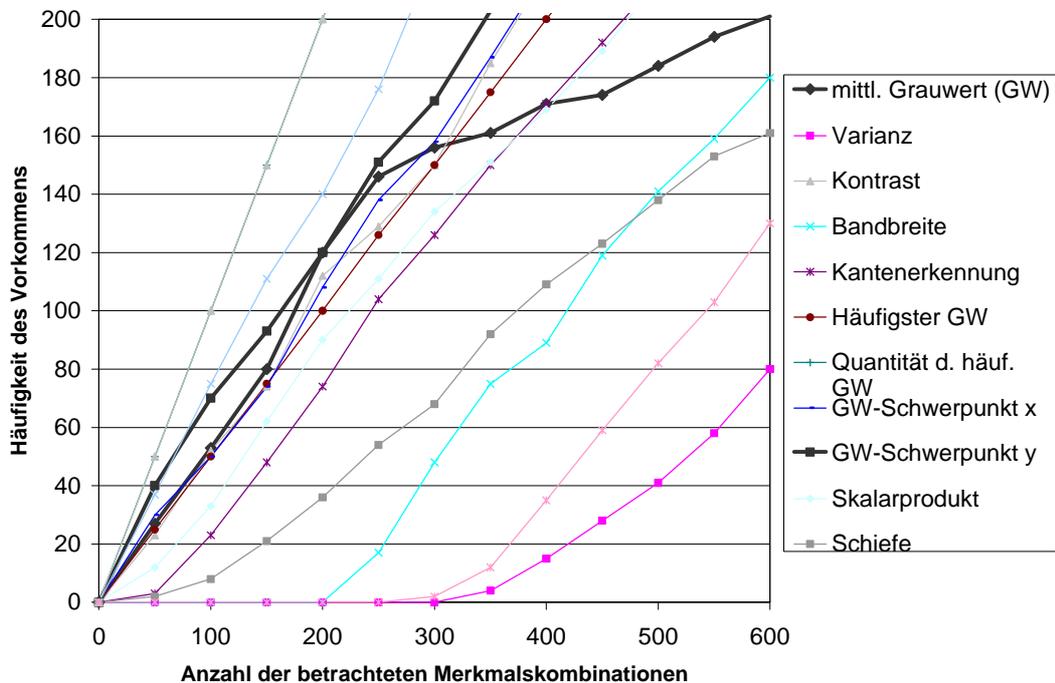


Abbildung 33: Diagramm zur Merkmalsauswahl. Dargestellt ist die Häufigkeit des Auftretens eines Merkmals unter den besten Merkmalskombinationen. Die beiden fett hervorgehobenen Merkmale werden im Versuchssystem verwendet. ([Heinzen03])

Rechenaufwand

Neben dem Beitrag des Merkmals zur Klassifikation ist die zur Berechnung des Merkmals notwendige Rechenzeit ein entscheidender Faktor. Die Zeit, die zur Berechnung eines Merkmalsbildes auf dem Versuchsrechner (siehe Kapitel 7.1) benötigt wurde, ist in Abbildung 34 zu sehen.

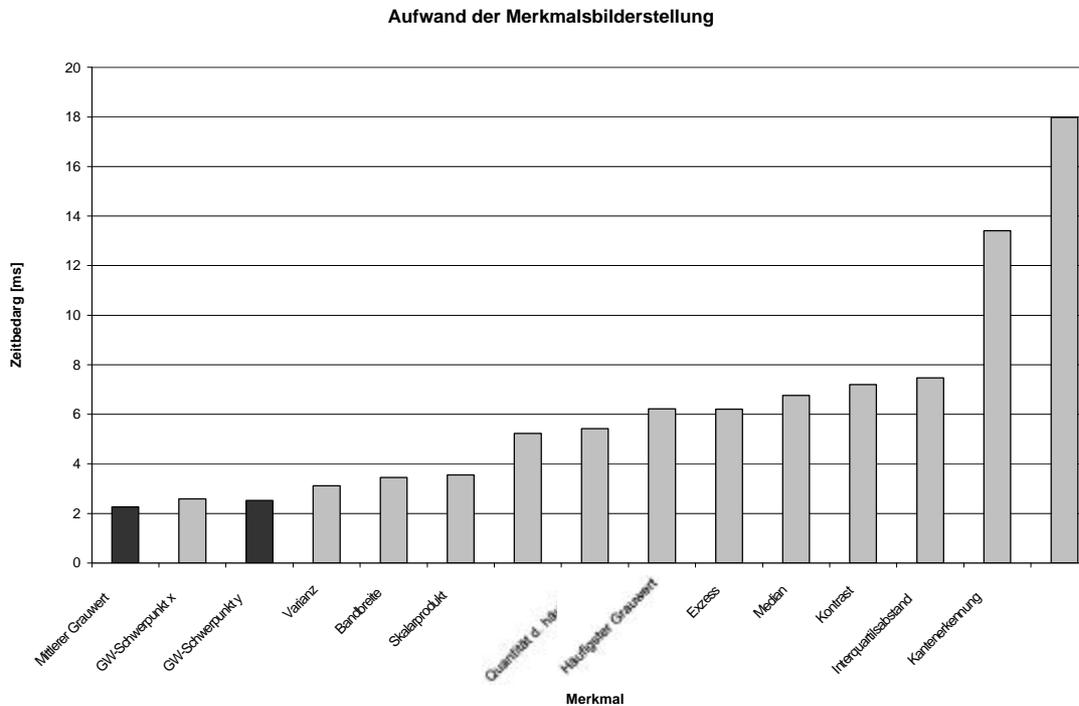


Abbildung 34: Rechenaufwand zur Erstellung eines Merkmalsbildes in Abhängigkeit des verwendeten Merkmals. Die beiden dunkel hervorgehobenen Merkmale werden im Versuchssystem verwendet. ([Heinzen03])

Zu erkennen ist, dass sich die Merkmale bezüglich der Rechenzeit in drei Gruppen zusammenfassen lassen. Die erste Gruppe besteht aus Merkmalen, die in weniger als 4ms berechnet werden können. Die zweite Gruppe braucht zwischen 4ms und 8ms, die dritte Gruppe mehr als 8ms.

Bei der Verwendung von mehreren Merkmalen ist die Gesamtrechenzeit kleiner als die Summe der einzelnen Zeiten. Die Ursache dafür ist, dass der Speicherzugriff für jedes Pixel nur einmal notwendig ist, unabhängig von der Anzahl der berechneten Merkmale. Außerdem brauchen eventuell vorhandene, gemeinsame Teilterme der Merkmale nur einmal berechnet werden.

Einfluss der Kachelgröße

Kleine Kacheln lassen eine genauere Darstellung der Objekte im Raum zu. Aus Sicherheitsgründen ist eine konservative Abschätzung der Hindernisse notwendig. Daher bewirken kleinere Kacheln, dass ein kleinerer Anteil des Differenzbildes Vordergrund ist. Dies erhöht die Bewegungsfreiheit des Roboters.

Der Rechenaufwand für ein Merkmalsbild ist weitgehend unabhängig von der Segmentgröße. Jedes Grauwertpixel wird einmal gelesen und zur Berechnung des Merkmals in

einer Kachel verwendet. Da jedoch der Kollisionstest pixelweise auf den Silhouetten arbeitet, hängt die Gesamtsystemleistung stark von der Kachelgröße ab.

Aus diesem Grund wird eine möglichst große Kachelung gesucht, die noch eine ausreichende Silhouettenqualität liefert.

Der Anteil der korrekt klassifizierten Pixel in Abhängigkeit der verwendeten Kachelgröße ist in Abbildung 35 dargestellt. Zu erkennen ist, dass die Qualität bis zu einer Kachelgröße von 16x16 Pixeln ungefähr gleich bleibt. Bei größeren Kacheln wird die Qualität deutlich schlechter.

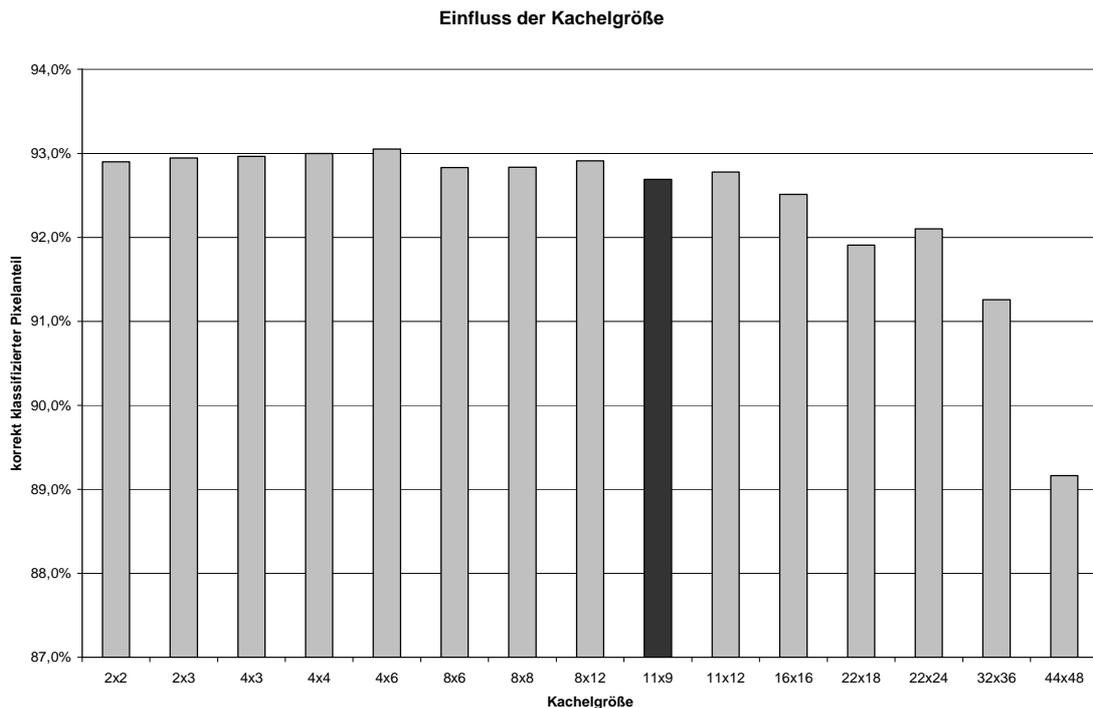


Abbildung 35: Einfluss der Kachelgröße auf die Klassifizierungsqualität. Die dunkel hervorgehobene Kachelgröße wird im Versuchssystem verwendet. ([Heinzen03])

Ergebnis

Als Merkmale werden mittlerer Grauwert und Grauwertschwerpunkt in Y-Richtung verwendet, da diese bereits eine gute Erkennung bei sehr kleinem Rechenaufwand ermöglichen.

Als Kachelgröße werden 11x9 Pixel verwendet. Dieser Wert wurde gewählt, weil bei der verwendeten Grauwertbildauflösung von 704x576 Pixeln die resultierenden Merkmals- und Differenzbilder genau 64x64 Pixel groß sind und der Unterschied in der Klassifikationsqualität zu den Alternativgröße 11x12 und 16x16 gering ist. Die Eigenschaft, dass die Silhouetten in beiden Dimensionen gleich groß sind und die Größe eine 2er-Potenz ist, wurde im bilderzeugenden Robotermodell zur Kompression der Bilder verwendet.

4.1.5 Differenzbilderstellung

Die Aufgabe der Differenzbilderstellung ist die eigentliche Berechnung des Silhouettenbildes aus dem Szenen- und Referenzbild. Hierzu werden lediglich die Merkmalsbilder verwendet, nicht die ursprünglichen Grauwertbilder. Es handelt sich hierbei um ein zweistufiges Verfahren.

Im ersten Schritt wird die auf den Merkmalen definierte Metrik verwendet, um den Abstand zwischen einer Szenenkachel und der zugehörigen Referenzkachel zu bestimmen. Dieser Abstand ist ein Vektor mit der Anzahl der Merkmale als Dimension. Im zweiten Schritt entscheidet ein Klassifikator anhand des Abstandes, ob die Kachel zum Vordergrund oder Hintergrund gehört.

Die im Klassifikator entstandenen Silhouettenbilder können unter Umständen noch nachbearbeitet werden. Dies kann zum Beispiel geschehen, um die Hindernisse zu vergrößern und so einen Mindestabstand zwischen Roboter und Hindernis zu erzielen oder um Einzelpixelfehler auszugleichen.

Im realisierten System sind aus Gründen der Einfachheit alle Merkmale skalare Größen. Als Metrik wird die absolute Differenz der Merkmale verwendet. Diese Differenz wird für einige Versuche nichtlinear verzerrt, zum Beispiel über die Gamma-Funktion. Außerdem werden in einigen Versuchen die Merkmale der Nachbarkacheln als zusätzliches Merkmal verwendet.

Klassifikatoren

Die Aufgabe der Klassifikatoren besteht darin, die Pixel der Merkmalsbilder in Vordergrund oder Hintergrund zu unterteilen. Da nur diese zwei Klassen auftreten, kann nach [Duda73] ein Klassifikator mit nur einer Entscheidungsfunktion verwendet werden. Die allgemeine Form eines solchen Klassifikators ist in Abbildung 36 dargestellt.

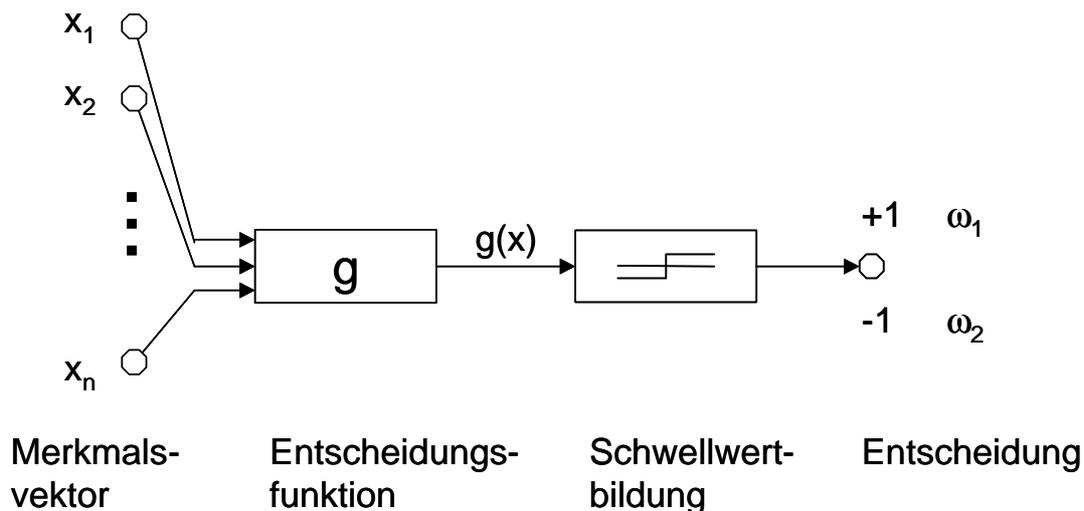


Abbildung 36: Allgemeine Form eines Klassifikators für 2 Klassen. ([Duda73])

Von einem Objekt O werden die Merkmale x_1, \dots, x_n berechnet. Ausgehend von diesem Merkmalsvektor \mathbf{x} wird eine Entscheidungsfunktion $g(\mathbf{x})$ berechnet. Ist diese Entscheidungsfunktion größer als 0, ordnet der Klassifikator das Objekt O der Klasse ω_1 zu, ansonsten der Klasse ω_2 .

In der Silhouettenerzeugung sind die zu klassifizierenden Objekte die Pixel der Merkmalsbilder. Der Merkmalsvektor des Klassifikators ist die Differenz der Merkmale zwischen aktuellem Merkmalspixel und Referenzmerkmalspixel. Die Klassen, denen die Pixel zuzuordnen sind, sind Vordergrund und Hintergrund.

Statistischer Klassifikator

Der optimale Klassifikator ist der Bayes-Klassifikator [Duda73]. Allerdings hat er den Nachteil, dass zum Training des Klassifikators bei einer unbekanntenen Merkmalsverteilung sehr große Trainingsmengen notwendig sind.

Bei der Verwendung von 4 Merkmalen mit je 256 (2^8) Werten gibt es bereits über 4 Milliarden Kombinationen (2^{32}). Je nach Kachelgröße besteht ein Merkmalsbild aus etwa 1000 – 65 000 Pixeln. Bei einer angestrebten Anzahl von etwa 10 Einträgen pro Kombination bräuchte man im besten Fall zum Training des Klassifikators 650 000 von Hand klassifizierte Trainingsbilder.

Da dies nicht sinnvoll machbar ist, wird im folgenden ausschließlich der lineare Klassifikator verwendet. Die Silhouettenerzeugung mit Hilfe des Bayes-Klassifikator bei der Beschränkung auf zwei Merkmale wurde in [Heinzen03] untersucht.

Linearer Klassifikator

Bei einem linearen Klassifikator ist die Entscheidungsfunktion:

$$\mathbf{w} \cdot \mathbf{x} - w_0 \stackrel{?}{=} 0 \quad (18)$$

Dabei ist \mathbf{w} der Gewichtsvektor für die Merkmale und w_0 der Schwellwert.

Bestimmung von Gewichtsvektor und Schwellwert

Der Gewichtsvektor kann mit Fisher's linearer Diskriminante ermittelt werden [Duda73]:

Wenn \mathbf{m}_i der Vektor der Mittelwerte der i -ten Klasse ist mit

$$\mathbf{m}_i = \frac{1}{n} \sum_{\mathbf{x} \in X_i} \mathbf{x} \quad (19)$$

dann wird die Streuungsmatrix (Scatter Matrix) \mathbf{S}_i der i -ten Klasse zu

$$\mathbf{S}_i = \sum_{\mathbf{x} \in X_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \quad (20)$$

Mit Hilfe der Gesamtstreuung \mathbf{S}_w

$$\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2 \quad (21)$$

lässt sich der Gewichtsvektor \mathbf{w} bestimmen als

$$\mathbf{w} = \mathbf{S}_w^{-1} (\mathbf{m}_1 - \mathbf{m}_2) \quad (22)$$

Im Prinzip kann der optimale Schwellwert mit Hilfe der ROC-Kurve ermittelt werden. Allerdings gibt es das Problem, dass die ROC-Kurve in der Realität nicht glatt genug ist, um robuste Ergebnisse zu erzielen ([Heinzen03]). Daher wird der Schwellwert w_0 durch systematisches Durchprobieren aller sinnvollen Werte ermittelt.

Morphologische Operatoren und Nachbarschaft

Bei der Klassifikation können die Merkmale der Nachbarpixel berücksichtigt werden. Damit die Anzahl der Merkmale nicht zu sehr steigt, werden die Merkmale der Nachbarpixel zu einem einzigen Merkmal verrechnet. Die 4er Nachbarschaft umfasst die direkt benachbarten Pixel, die 8er Nachbarschaft berücksichtigt zusätzlich die diagonal benachbarten Pixel.

Die vom Klassifikator gelieferten Silhouetten können zusätzlich durch die morphologischen Operatoren Erosion, Dilatation, Öffnen, Schließen und Median nachbearbeitet werden. Als charakteristisches Element wird dabei die 8er-Nachbarschaft verwendet.

Der Einfluss der Morphologischen Operatoren und der Nachbarschaft auf die Silhouettenqualität ist in Abbildung 37 dargestellt.

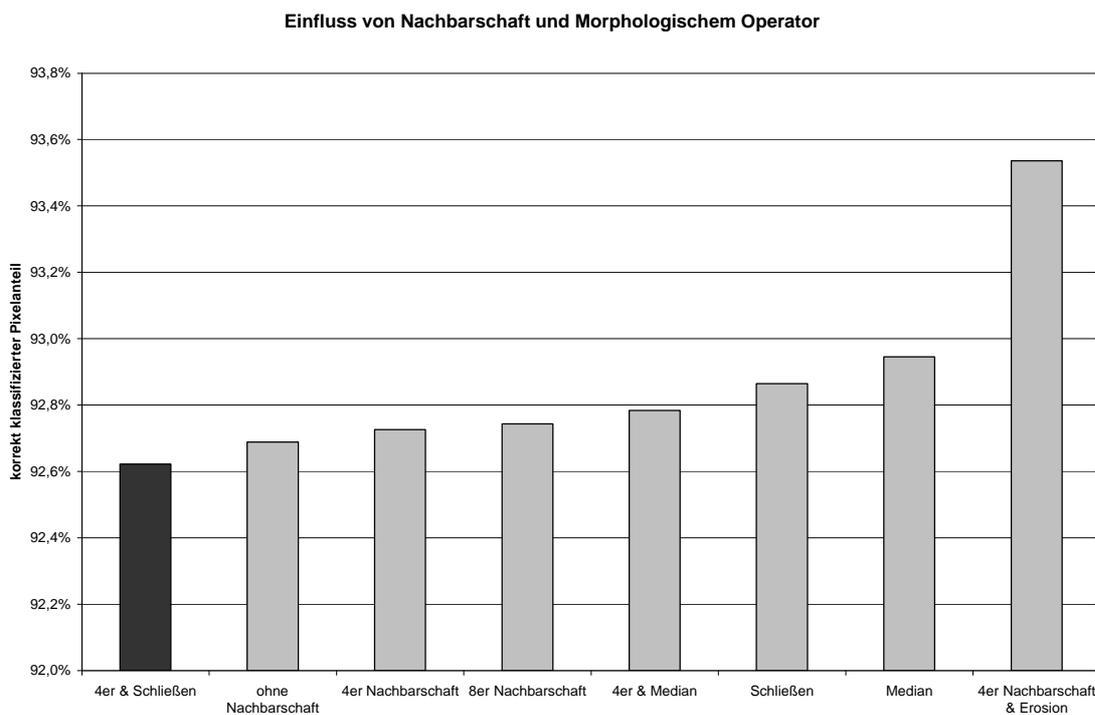


Abbildung 37: Einfluss von Nachbarschaft und morphologischem Operator ([Heinzen03])

Im Versuchssystem wird die 4er-Nachbarschaft zusammen mit einem CLOSE-Operator verwendet, obwohl der Anteil der klassifizierten Pixel schlechter ist als bei anderen Kombinationen. Durch das Vergrößern der Hindernisse wird allerdings verhindert, dass der Roboter mit einem Objekt kollidiert.

Die Ursache dafür ist, dass beide Arten des falsch klassifizierten Pixels bislang als gleich schwerwiegend angenommen wurden. Tatsächlich ist der Schaden der beiden Fehlermöglichkeiten FN bzw. FP aber unterschiedlich. Ein nicht erkanntes Vordergrundpixel kann im schlimmsten Fall zu einer Kollision führen, während ein falsch er-

kanntes Hintergrundpixel höchstens zum Stehen bleiben des Roboters führt. Um diesen Unterschied zu berücksichtigen, werden die Fehler unterschiedlich gewichtet.

Mit Hilfe eines Parameters λ kann bestimmt werden, um welchen Faktor es wichtiger ist, den Vordergrund als solchen zu erkennen. Bei $\lambda > 1$ wird der Vordergrund stärker gewichtet, bei $\lambda < 1$ der Hintergrund ([Heinzen03]). Dadurch wird der gewichtete Anteil der korrekt klassifizierten Pixel a_{kor} zu:

$$a_{kor} = \frac{I \cdot KP + KN}{I \cdot KP + I \cdot FN + KN + FP} \quad (23)$$

4.1.6 Referenzbilderstellung

Als Referenzbild wird das Merkmalsbild einer leeren Szene verwendet. Da Industrieroboter normalerweise nicht einfach aus der Zelle entfernt werden können, muss das Referenzbild künstlich erzeugt werden.

Aus den Grauwertreferenzbildern für jede Kamera werden Merkmalsbilder berechnet. Nur diese Merkmalsreferenzbilder werden für die weiteren Schritte der Bildverarbeitung verwendet. Ausnahme ist das Merkmal „Skalarprodukt“, bei dem direkt die Grauwertpixel verwendet werden.

Die Merkmalsberechnung erfolgt genauso wie bei den Grauwertbildern der aktuellen Szene, die im Abschnitt 4.1.4 beschrieben wurden. Die Erzeugung der zugrundeliegenden Grauwertreferenzbilder wird im folgenden beschrieben.

Ansatz

Die Kameras sind so ausgerichtet, dass der gemeinsame Arbeitsraum von Mensch und Roboter überwacht wird. In den Kamerabildern ist jeweils nur ein relativ kleiner Bildanteil durch den Roboter verdeckt.

Hierzu wird der Roboter in verschiedene Konfigurationen gefahren und die zugehörigen Grauwertbilder aufgenommen. Für jede Koordinate der Grauwertpixel wird der Grauwert gesucht, der dem Hintergrund entspricht. Unter der Annahme, dass jedes Grauwertpixel bei geschickter Wahl der Konfigurationen häufiger den Hintergrund zeigt als den Vordergrund, kann aus den verschiedenen Pixelwerten der Wert für den Hintergrund ermittelt werden.

Bestimmung des Hintergrundpixelwertes

Unter der Annahme, dass jedes Pixel häufiger den Hintergrund zeigt als den Vordergrund, entspricht der Median der sortierten Liste der verschiedenen Grauwerte, die für ein Pixel gemessen wurden, dem Hintergrundgrauwert. Ein weiterer Ansatz wurde in [Diehl01], untersucht führte aber zu keinen besseren Ergebnissen.

Auf Grund von Schattenwurf durch den Roboter könnte allerdings bei einigen Pixeln der abgeschattete Hintergrund in der Mitte der sortierten Liste liegen.

Bestimmung der Roboterkonfigurationen

Die Roboterkonfigurationen, die zur Referenzbilderstellung verwendet werden, müssen so gewählt sein, dass in der sortierten Liste der Grauwerte von möglichst vielen Pixeln möglichst häufig der Hintergrund zu sehen ist.

Um diese Bedingung zu erfüllen, ist Wissen über die Positionierung der Kameras notwendig. Im Prinzip ist eine Ableitung dieser Konfigurationen aus den Kameraparametern, die während der Kalibrierung gewonnen werden, möglich. Dies wurde aber aus Zeitgründen nicht realisiert.

Eine offene Frage ist, wie viele Konfigurationen zur Referenzbilderstellung verwendet werden sollen. Für die Verwendung möglichst vieler Konfigurationen spricht, dass die Fehler Pixelrauschen und Schattenwurf eine geringere Rolle spielen. Für möglichst wenige Konfigurationen spricht, dass der Roboter an weniger Pixeln mehr als einmal zu sehen ist und die kürzere Zeit zum Erstellen der Referenzbilder.

Im realisierten System werden zwei verschiedene Verfahren zur Referenzbilderstellung eingesetzt, die sich darin unterscheiden, ob Vorwissen über die Positionierung der Kameras ausgenutzt wird. Die Variante, die das Vorwissen ausnutzt, positioniert den Roboter so, dass er in allen Bildern möglichst verschieden aussieht. Dann reichen pro Kamera drei Bilder aus. Dies ist in Abbildung 38 zu sehen. Wenn die Position der Kameras nicht bekannt ist, wird ein generischer Satz aus neun Positionen verwendet [Ebert01]. Die Qualität des Referenzbildes ist dann allerdings schlechter.

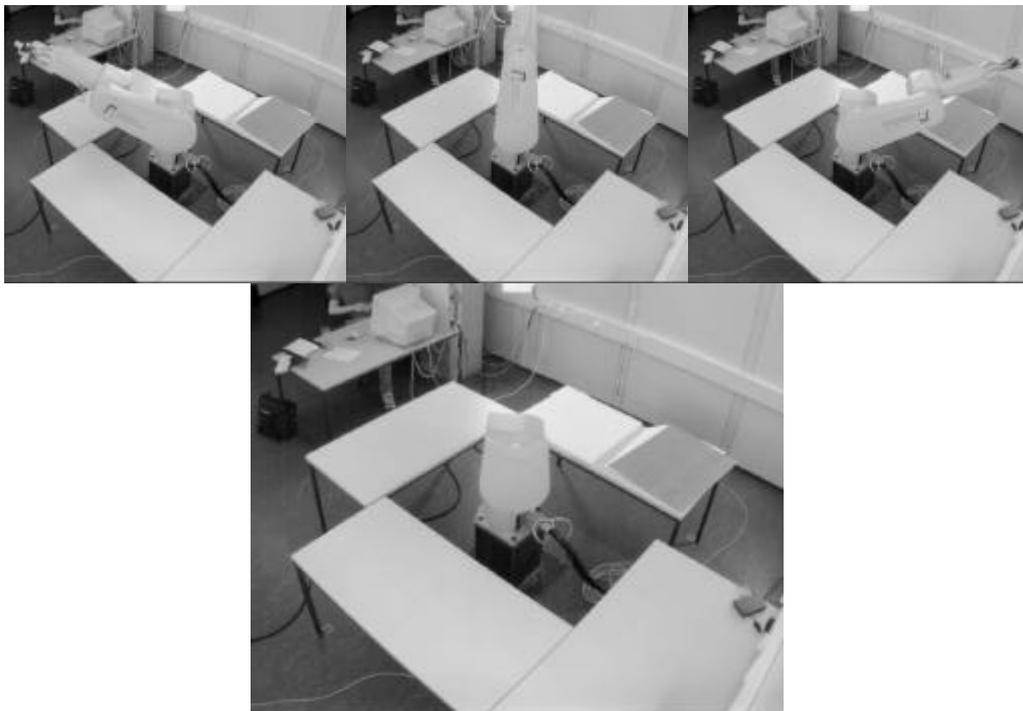


Abbildung 38: Referenzbilderstellung. Der Roboter wird in drei verschiedene Konfigurationen gefahren. Zu diesen Konfigurationen werden die Kamerabilder aufgenommen (Bilder oben). Mittels des Median der sortierten Liste der Grauwerte an einer bestimmten Pixelkoordinate wird der Hintergrund ermittelt (Bild unten).

Referenzbildaktualisierung

Im längeren Betrieb kann das Referenzbild veralten. Dies geschieht zum Beispiel durch langsame Änderungen der Helligkeit im Tagesablauf. Der Effekt ist, dass übermäßig viele und große Vordergrundobjekte erkannt werden. Durch eine Aktualisierung des Referenzbildes wird diesem Veralten entgegengewirkt.

Problematisch dabei ist das Finden einer geeigneten Aktualisierungsrate. Wird das gesamte Szenenbild gleich stark zur Aktualisierung verwendet, tritt der Effekt auf, dass Hindernisse, die längere Zeit an einem Ort bleiben, nicht mehr wahrgenommen werden. Aus diesem Grund ist es vorteilhaft, diejenigen Bereiche, die aktuell als Vordergrund wahrgenommen werden nicht oder nur schwach zur Aktualisierung des Referenzbildes zu verwenden.

In einem Versuch wurde die Veränderung der Pixelwerte durch die Sonneneinstrahlung untersucht. Die Beleuchtung wurde 30 Minuten vor Versuchsbeginn angeschaltet, damit die Leuchtstoffröhren bereits ihre maximale Helligkeit erreicht haben. Insgesamt 12 Stunden lang wurden alle 15 Minuten von allen Kameras jeweils 3 Bilder aufgenommen. Die drei Bilder dienten zur Eliminierung des Kamerarauschens. Es zeigte sich, dass sich die Referenzbilder durch das Tageslicht nicht wesentlich ändern.

In Abbildung 39 ist die Differenz der mittleren Grauwerte von jeweils einer Stunde vor Sonnenaufgang (6:30-7:30 Uhr) und einer Stunde während der höchsten Helligkeit (11:00 Uhr-12:00 Uhr) aufgetragen. Es ist zu erkennen, dass die große Mehrzahl der Pixel sich nur wenig verändert hat. Das Kamerarauschen beträgt bei den verwendeten Kameras etwa 2 Bit. Daher ist der größte Teil der Veränderung über das Kamerarauschen erklärbar.

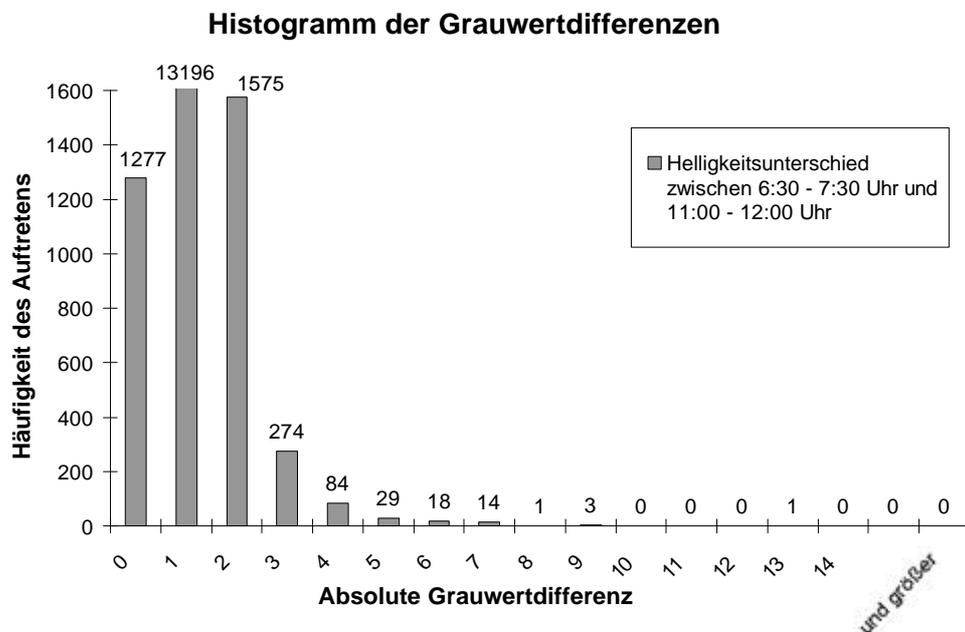


Abbildung 39: Histogramm der Grauwertdifferenzen. Dargestellt ist die Verteilung der Differenz der mittleren Grauwerte zwischen 6:30-7:30 Uhr (vor Sonnenaufgang) und 11:00-12:00 Uhr von einer Stichprobe aus 16472 Pixeln.

Wesentliche Veränderungen der gemessenen Grauwerte kommen in den ersten 15 Minuten nach Einschalten der Beleuchtung dadurch zustande, dass sich die Helligkeit der Beleuchtung langsam dem Maximum annähert. Bei durchgeführten Experimenten tritt die Alterung der Referenzbilder vor allem durch kleine Verschiebungen der Umweltgegenstände auf, die in der leeren Zelle bereits vorhanden sind.

Sicherheitsaspekte

Aus Sicherheitsüberlegungen heraus ist die Referenzbilderstellung ein kritischer Vorgang. Der Roboter wird über große Entfernungen schnell bewegt, ohne dass diese Bewegung abgesichert ist. Eventuell vorhandene Hindernisse im Arbeitsraum werden nicht wahrgenommen.

Außerdem werden für die folgenden, abgesicherten Transferbewegungen nur Objekte berücksichtigt, die im Vergleich zum Referenzzustand hinzukommen. Ein Hindernis, das im Normalbetrieb im Weg des Roboters steht, wird dann nicht wahrgenommen, wenn es bereits beim Aufbau der Referenzbilder im Arbeitsraum war.

Die Referenzbildaktualisierung überträgt die Gefahr des Übersehens von Hindernissen in den Normalbetrieb. Diese Sicherheitsüberlegungen haben, zusammen mit der geringen Alterung der Referenzbilder im praktischen Betrieb, dazu geführt, dass die Referenzbildaktualisierung im realisierten System nicht weiter verwendet wird.

4.1.7 Kamerakalibrierung

Einige Module, wie das bilderzeugende Robotermodell (siehe Abschnitt 4.2.2) und die Kollisionstests mit Epipolarlinien (siehe Abschnitt 5.2.5) benötigen die genauen Abbildungsparameter aller Kameras des Systems. Diese Abbildungsparameter werden durch die Kamerakalibrierung ermittelt. Zur Kalibrierung von Kameras gibt es eine große Zahl an Verfahren. Im realisierten System wird das Verfahren aus [Bischoff99] verwendet.

Ermittlung der Kalibrierungstupel

Die Grundlage für alle Kalibrierungsmethoden ist die Anpassung eines Kameramodells. Hierbei werden Tupel von bekannten 3D-Raumkoordinaten und deren 2D-Bildkoordinaten verwendet. Diese Kalibrierungstupel werden üblicherweise dadurch ermittelt, dass ein Kalibrierungskörper an einer bekannten Raumposition aufgestellt wird.

Bei der Wahl des Koordinatensystems bezüglich dessen die Kameras kalibriert werden sollen, bietet sich das Weltkoordinatensystem des Roboters an. Dies hat den Vorteil, dass keine weitere Transformation zwischen den Koordinaten, die mit Hilfe der Kameras ermittelt werden und den Koordinaten, die zur Ansteuerung des Roboters verwendet werden, notwendig ist.

Für eine gute Kalibrierung ist es günstig, wenn die Bildkoordinaten der Kalibrierungstupel gleichmäßig über den ganzen Bildbereich verteilt sind. Normalerweise erfordert dies einen großen Aufwand, da die Kalibrierungskörper meist nicht das ganze Bild überdecken und daher mehrmals genau positioniert werden müssen. Diese Positionierung kann jedoch auch dadurch geschehen, dass ein Kalibrierungskörper durch den Roboter bewegt wird.

Dieser Ansatz wurde beispielsweise in [Bischoff99] und [Maeda02] verfolgt, bei dem der Roboter eine punktförmige Marke, z.B. LED, im Greifer hatte.

Bei der gewählten Methode der Ermittlung der Kalibrierungstupel hält der Roboter einen Kalibrierungskörper mit Schachbrettmuster an verschiedenen Positionen ins Blickfeld der Kameras. Die Positionen sind so gewählt, dass die Kalibrierungstupel im Rahmen der Einschränkungen durch die Reichweite des Roboters möglichst über das gesamte Bild verteilt sind.

Der Kalibrierungskörper wird durch eine spezielle Befestigung formschlüssig und exakt im Greifer positioniert und ist dennoch schnell einsetz- und entfernbar. Der Kalibrierungskörper und die Befestigung sind in Abbildung 40 zu sehen.

Mittels des Algorithmus von [Hamfeld02] werden alle inneren Berührungspunkte der Schachbrettfelder gesucht. Für jeden dieser Berührungspunkte wird mit Hilfe der Robotersteuerung oder eines internen Robotermodells die Weltkoordinaten berechnet. Zusammen mit den gefundenen Bildkoordinaten ergeben sich so die Kalibrierungstupel.

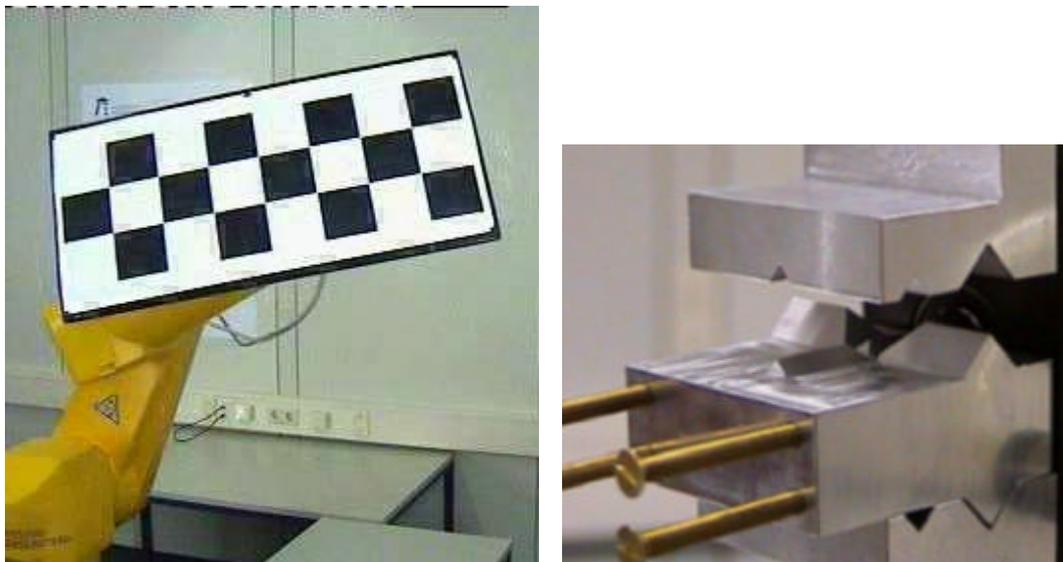


Abbildung 40: Kalibrierungskörper durch den Roboter gehalten (links). Formschlüssige Befestigung des Kalibrierungskörpers im Greifer (rechts).

Berechnung der Epipolarlinien

Im Prinzip könnten die Epipolarlinien direkt aus dem parametrisierten Kameramodell abgeleitet werden. Es wurde jedoch ein anderes Verfahren gewählt, bei dem die Epipolarlinien über eine schwache Kalibrierung ermittelt werden.

Die dazu notwendige Fundamentalmatrix für jedes Kamerapaar wird durch virtuelle Kalibrierungstupel bestimmt. Die virtuellen Kalibrierungstupel werden erzeugt, in dem zu beliebigen Raumkoordinaten mittels der in den vorherigen Abschnitten beschriebenen Kalibrierung die zugehörigen Bildkoordinaten berechnet werden.

Dieses Vorgehen hat den Vorteil, dass die Berechnung der Epipolarlinien unabhängig vom gewählten Kameramodell ist, sowie dass ein beliebiges Standardverfahren zur Ermittlung der Fundamentalmatrix verwendet werden kann. Der Nachteil ist, dass eventuell die berechneten Epipolarlinien ungenauer werden. Die im realisierten System verwendete Methode der Berechnung der Fundamentalmatrix ist in [Körte03] beschrieben.

Eine Besonderheit im realisierten System ist die flächige Darstellung der Epipolarlinien. Technisch wird dies dadurch erreicht, dass die Epipolarlinien als eine Pixelmenge betrachtet werden, die nicht notwendigerweise eine Gerade sind.

4.2 Silhouettenerzeugende Robotermodellierung

Die Aufgabe der Robotermodellierung besteht darin, die Kollisionserkennung mit Wissen über die Geometrie des Roboters zu versorgen. Im realisierten System wird ein silhouettenbasierter Kollisionstest verwendet. Daher liefert die Robotermodellierung zu einer angefragten Konfiguration ein Silhouettenbild des Roboters.

Diese Aufgabe kann durch der Berechnung und Rendern eines geometrischen Modells des Roboters gelöst werden.

Neben der Berechnung und Abbildung eines geometrischen Modells kann die gestellte Aufgabe auch auf andere Art gelöst werden.

In einer Bilddatenbank könnten Silhouetten für alle möglichen Anfragekonfigurationen gespeichert sein. Die Erzeugung der Silhouetten wäre dann durch einen Zugriff auf diese Datenbank realisierbar. In einer Trainingsphase müsste die Datenbank mit Silhouetten gefüllt werden. Auch eine sensorgestützte Anpassung eines Modells ist möglich. Dabei wird ein allgemeines Modell des Roboters mit Hilfe der Sensordaten parametrisiert.

Alle Methoden und ihre Vor- und Nachteile werden im folgenden beschrieben.

4.2.1 Bilddatenbank

Bei der Robotermodellierung mittels Bilddatenbank wird der Roboter nacheinander in alle möglichen Abfragekonfigurationen bewegt. In jeder dieser Konfigurationen werden mit Hilfe der Bildverarbeitung Silhouettenbilder des Roboters erzeugt und in einer Datenbank abgelegt. Bei der Abfrage wird das entsprechende Bild aus der Datenbank abgerufen.

Der Vorteil ist, dass keinerlei Annahmen über die Kinematik und Geometrie des Roboters gemacht werden. Außerdem kann bei geeigneter Realisierung der Zugriff auf die Robotersilhouetten sehr schnell gehen.

Der Nachteil ist, dass das Modell nur zu den gelernten Konfigurationen die Roboterbilder erzeugen kann. Dies lässt sich auch nicht durch Verfahren des Image-based Renderings umgehen, wie sie zum Beispiel in [Matusik00] beschrieben sind. Diese Verfahren erzeugen Bilder, wie sie aus einer bestimmten Perspektive zu sehen sind und verwenden als Eingabe Bilder aus anderen Perspektiven. Allerdings wird dabei angenommen, dass das dargestellte Objekt seine Form nicht ändert. In der Robotermodellierung müssten Bilder erzeugt werden, die aus der gleichen Perspektive wie die Eingabebilder aufgenommen sind, jedoch den Roboter in einer anderen Form zeigen.

Ein weiterer Nachteil ist, dass der Zeitaufwand zum Lernen bei einer großen Anzahl an Konfigurationen sehr hoch ist. Außerdem muss bei jeder Veränderung der Kamerakonfiguration, Robotergeometrie oder Konfigurationsraumparametern die Bilddatenbank neu aufgebaut werden.

Außerdem besteht das Problem, dass die Bildverarbeitung nicht auf das Erkennen des Roboters spezialisiert ist. Beim Aufbau der Bilddatenbank ist es erforderlich, dass der

Roboter gut erkannt wird. Da der Grauwert des eigentlich orange-gelben Roboters sehr ähnlich dem Grauwert der Umgebung ist, hat das resultierende Robotermodell Lücken. Dies kann zum Beispiel durch den Einsatz von Farbfiltern bei der Kamera umgangen werden. Der Effekt von Farbfiltern ist in Abbildung 41 zu sehen.



Abbildung 41: Effekt eines Farbfilters auf den Grauwert des Roboters. Vor der Kamera wird ein Blaufilter angebracht (links). Ohne Farbfilter ist der Grauwert des orange-gelben Roboters sehr ähnlich dem Grauwert des Hintergrundes (Mitte). Mit Farbfilter ist der Roboter deutlich dunkler (rechts).

4.2.2 Modellberechnung

Bei der reinen Modellberechnung ohne Unterstützung von externen Sensoren wird zuerst aus den Gelenkwinkeln des Roboters mit Hilfe eines CAD-Modells die räumliche Ausdehnung des Roboters bestimmt. Anschließend werden die räumlichen Objekte des Robotermodells mit Hilfe von Kameramodellen in Bildobjekte der Kameras abgebildet.

Der Vorteil der Methode ist, dass das Roboterbild für jede beliebige Konfiguration ermittelt werden kann. Der Nachteil ist, dass das Modell nur für genau einen Roboter gilt und alle Modifikationen eine manuelle Anpassung des Modells erfordern.

Aufbauend auf einem kinematischen Modell wird das Volumen des Roboters durch Pyramidenstümpfe approximiert. Anschließend wird das Geometriemodell des Roboters in die Kamera projiziert. Das im realisierten System verwendete Verfahren zur Modellberechnung ist in [Gecks03] beschrieben. Ein Beispiel ist in Abbildung 42 zu sehen.

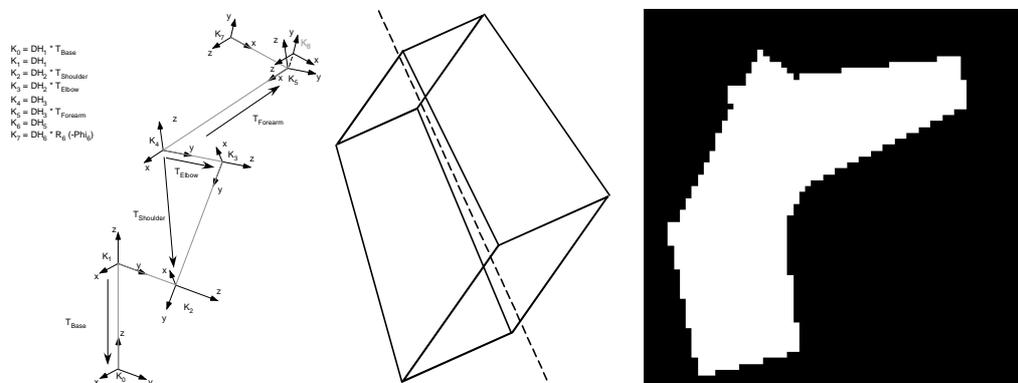


Abbildung 42: Kinematisches Modell des Roboters (links), Pyramidenstumpf zur Approximation der Gelenke (Mitte), berechnete Robotersilhouette (rechts) ([Gecks03]).

4.2.3 Anpassung eines Modells

Eine Art Mischform aus den beiden bisher vorgestellten Methoden ist die sensorgestützte Anpassung eines Modells. Anstatt alle Modelldaten vorzugeben, wird nur ein Teil, zum Beispiel das Roboterskelett, vorgegeben, während die anderen Parameter, zum Beispiel die Segmentdurchmesser, in einer Lernphase adaptiert werden.

Der Vorteil dieses Verfahrens ist, dass der Lernaufwand wesentlich reduziert wird, dass bei leichten Veränderungen am Roboter lediglich eine neue Adaption nötig ist und dass der Roboter zu jeder beliebigen Konfiguration herausgerechnet werden kann, sofern das adaptierte Modell korrekt ist.

Der Nachteil ist, dass die Korrektheit des Modells nur für die gelernten Konfigurationen sicher ist und dass eine Änderung des Roboters immer noch eine manuelle Modelländerung erfordert. Das robuste Lernen ist sehr aufwendig in der Realisierung. Diese Art des Modells hat, ebenso wie die Bilddatenbank, das Problem, dass die Bildverarbeitung eigentlich nicht auf das Erkennen des Roboters spezialisiert ist.

4.2.4 Modellierung von Konfigurationsfolgen

Die bisher vorgestellten Verfahren erzeugen die Robotersilhouette für eine einzige Anfragekonfiguration. Wenn der Kollisionstest eine ganze Bahn auf Kollision testen möchte, muss er die Bahn zuerst in einzelne Konfigurationen aufspalten. Dann kann er die notwendigen Robotersilhouetten durch eines der beschriebenen Robotermodelle bekommen.

Effizienter wäre es, könnte der Kollisionstest die ganze Bahn auf einen Schlag auf Kollision testen. Ein entsprechendes Verfahren wird in Kapitel 5.5.1 beschrieben. Dieses Verfahren benötigt eine Silhouette, welche die gesamte Bewegung des Roboters abdeckt.

Wenn das überstrichene Volumen der Roboterbewegung (Swept Volume) bekannt ist, können die zugehörigen Swept-Volume-Silhouetten durch die Projektion des Volumens in die Kameraebenen erzeugt werden.

Im Folgenden werden Ansätze beschrieben, wie das überstrichene Volumen einer Roboterbewegung ermittelt werden kann.

Stand der Forschung

Eine Möglichkeit das überstrichene Volumen einer Roboterbewegung zu bestimmen, besteht in der Aufspaltung der Bahn in einzelne Konfigurationen. Für jede einzelne Konfiguration wird das vom Roboter belegte Volumen bestimmt. Die Vereinigung der Volumina für alle Konfigurationen der Bahn ergibt das überstrichene Volumen.

Der Aufwand dieses Ansatzes hängt davon ab, in wie viele Einzelkonfigurationen eine Bahn aufgespaltet werden muss. In [Baginski97] wird ein Verfahren vorgestellt, das auf einer Vergrößerung der Abmessungen des Roboters beruht. Wenn der Roboter in jeder Richtung um d Einheiten vergrößert wird, kann keine Kollision mit einem Hindernis auftreten, wenn sich der Roboter zwischen zwei Konfigurationen in jedem Punkt um weniger als $2d$ bewegt. Dies ist in Abbildung 43 dargestellt.

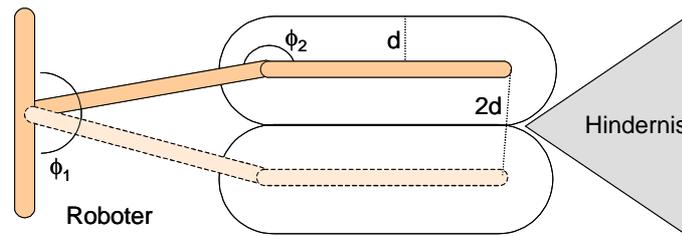


Abbildung 43: Reduktion der Anzahl der zu testenden Konfigurationen durch Vergrößern des Roboters. Wenn der Roboter in allen Richtungen um d Einheiten vergrößert wird, müssen nur Konfigurationen getestet werden, die sich in einem Punkt um mehr als $2d$ unterscheiden. Im Beispiel ist die Vergrößerung nur für das vorderste Gelenk eines ebenen Roboters mit zwei Freiheitsgraden f_1 und f_2 dargestellt. (ähnlich [Baginski97])

Das Problem dabei ist die Ermittlung der maximalen Gelenkwinkel ϕ_1, \dots, ϕ_n , die zwischen den zu testenden Konfigurationen liegen dürfen, ohne dass ein Punkt des Roboters sich mehr als $2d$ bewegt. Dies lässt sich durch eine Worst-Case Abschätzung ähnlich der in [Henrich98] vorgestellten MaxMove-Methode erreichen. Die Anzahl der insgesamt zu testenden Robotersegmente wird in [Baginski97] dadurch verringert, dass die Worst-Case Abschätzung für jedes Segment einzeln gemacht wird. Dies hat zur Folge, dass die vordersten Glieder des Roboters häufiger getestet werden als die inneren Glieder. Der Nachteil dabei ist, dass das Robotermodell die getrennte Berechnung der einzelnen Glieder unterstützen muss.

In [Baginski99] wird die Anzahl der zu testenden Konfigurationen dadurch reduziert, dass die PTP-Bewegung im Konfigurationsraum des Roboters durch mehrere lineare Bewegungen im Arbeitsraum abgebildet wird. Wird der Roboter geeignet verlängert, kann die konvexe Hülle des Roboters zum Test verwendet werden. Dies ist in Abbildung 44 zu sehen. Zur Berechnung der konvexen Hülle von Punktmenge existieren effiziente Algorithmen.

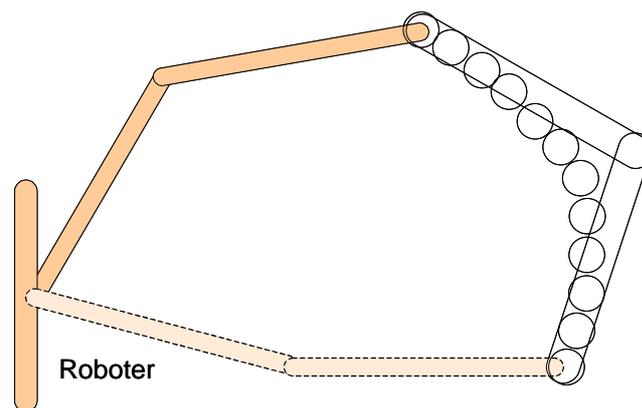


Abbildung 44: Kollisionstest durch konvexe Hülle des ausgedehnten Roboters. Wenn die Bahn von der oberen zur unteren Roboterposition getestet werden soll, untersucht das Verfahren aus [Baginski97] Einzelkonfigurationen entlang der Bahn. In der Abbildung ist dies durch die Kreise symbolisiert. Das Verfahren aus [Baginski99] berechnet einen vergrößerten Roboter in einer Position innerhalb der Bahn so, dass die konvexe Hülle dieses vergrößerten Roboters zusammen mit dem Roboter in Start- und Zielposition, die gesamte Bahn umschließt (ähnlich [Baginski99]).

Ansatz

Die Idee des verfolgten Ansatzes besteht darin, das vom Roboter während der Bewegung überstrichene Volumen durch eine Achsparallele Bounding-Box zu approximieren. Mit Hilfe des kinematischen Modells kann die maximale und minimale Position des Roboters in den Achsen des Weltkoordinatensystems ermittelt werden. Dabei muss die Dicke des Gelenkes an jedem Punkt des kinematischen Modells berücksichtigt werden. Dann kann garantiert werden, dass der Roboter innerhalb einer achsparallelen Bounding-Box liegt, die durch die Extremwerte bestimmt ist. Dies ist in Abbildung 45 dargestellt.

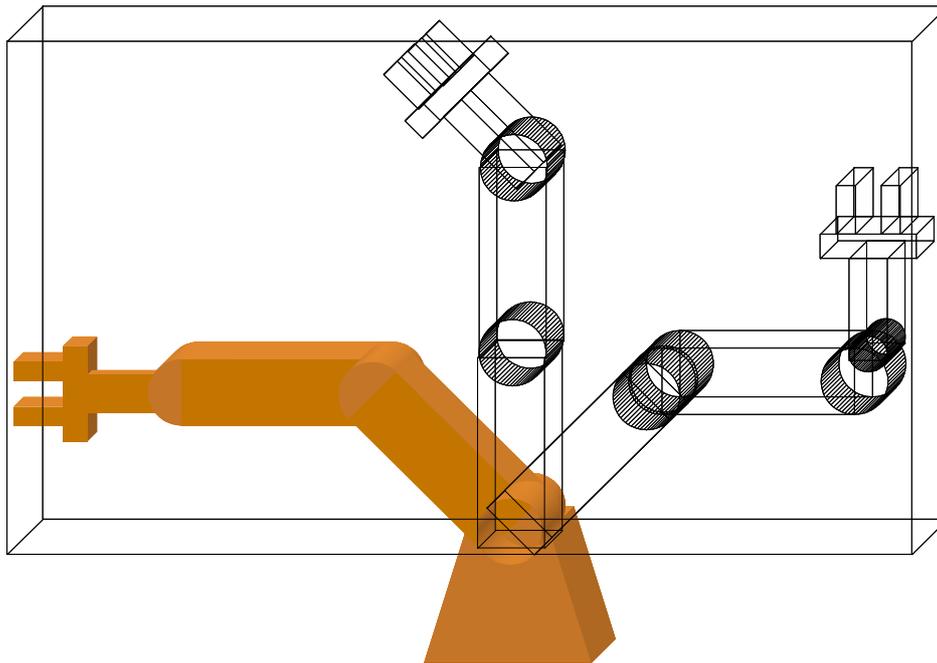


Abbildung 45: Approximation durch Bounding-Box. Das vom Roboter überstrichene Volumen wird durch eine achsparallele Bounding-Box approximiert. Hierzu wird die minimale und maximale Position des Roboters in x, y und z -Richtung während der Bewegung berechnet.

Ziel ist es durch eine schnelle Berechnung der Bounding-Box den Nachteil der qualitativ schlechteren Approximation des überstrichenen Volumens mehr als auszugleichen.

Obwohl der Roboter bei einer Punkt-zu-Punkt-Bewegung im allgemeinen alle Gelenke bewegt, ist die Bewegung eigentlich nur von einem einzigen Parameter l abhängig. Dieser Parameter gibt an, welchen Anteil des Weges die Gelenke bereits zurückgelegt haben.

Für die Gelenkwinkel \mathbf{f}_i gilt.

$$\mathbf{f}_i = \mathbf{f}_i^{Start} + l \cdot (\mathbf{f}_i^{Ende} - \mathbf{f}_i^{Start}), \quad l \in [0, 1] \quad (24)$$

Die Idee besteht darin, die Ableitung des kinematischen Modells des Roboters durch ein Polynom 3. Grades zu approximieren. Die Nullstellen eines Polynoms dritten Grades können durch eine geschlossene Formel berechnet werden. Damit eine Approximation möglich ist, wird der Wertebereich der Gelenkwinkel auf $[-\pi, +\pi)$ beschränkt. Die auf-

wändige symbolische Ableitung des kinematischen Modells und die Polynomapproximation kann durch ein mathematisches Softwarepaket erfolgen.

Für jeden Punkt des kinematischen Modells werden die Extremwerte berechnet. Die Bounding-Box ergibt sich dann aus den maximalen und minimalen Werten für jede Achse.

In [Urbanczik03] wurde der oben beschriebene Ansatz für das Versuchssystem realisiert. Die Swept-Volume-Silhouette wird durch die Projektion der Bounding-Box in die Kameraebenen erzeugt. Ein Beispiel ist in Abbildung 46 zu sehen.

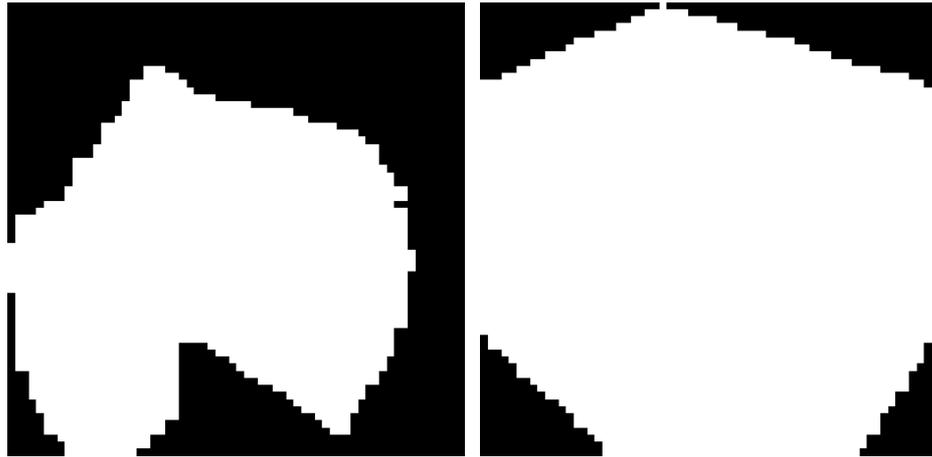


Abbildung 46: Swept-Volume-Silhouetten. Swept-Volume-Silhouette durch Überlagerung der Silhouetten der einzelnen Konfigurationen (links). Swept-Volume-Silhouette auf Basis der Bounding-Box (rechts). ([Urbanczik03])

4.2.5 Ergebnis

Im Versuchssystem wird normalerweise eine Bilddatenbank verwendet. Auf Grund des Kontrollflusses (siehe Kapitel 6.2) müssen nur die Bilder an bestimmten Abfragekonfigurationen gespeichert werden. Solange der Konfigurationsraum relativ klein ist, wiegt der Vorteil des schnellen Zugriffes schwerer als der notwendige Speicherbedarf.

Um die Probleme der schlechten Robotererkennung und des aufwändigen Aufbaus der Bilddatenbank zu umgehen, wird die Bilddatenbank mit Hilfe eines berechneten Geometriemodells gefüllt.

Die Swept-Volume-Silhouette wird bei der Kollisionserkennung für PTP-Bahnen verwendet, die im Kapitel 5.5.1 beschrieben ist.

Kapitel 5

Silhouettenbasierter, globaler

Ganzarmkollisionstest

Die Aufgabe des Kollisionstests ist die Entscheidung, ob eine angefragte Testkonfiguration zu einer Kollision mit Objekten der Umwelt führt. Der im Folgenden beschriebene Kollisionstest verwendet zur Entscheidung die Silhouetten von Roboter und Szene. Die Testkonfiguration darf aus dem gesamten Konfigurationsraum stammen, unabhängig von der aktuellen Roboterkonfiguration. Der Kollisionstest berücksichtigt bei der Kollisionsentscheidung den gesamten Arm des Roboters. Auf Grund dieser Eigenschaften handelt es sich um einen silhouettenbasierten, globalen Ganzarmkollisionstest.

5.1 Stand der Forschung

Eine Möglichkeit, das Problem des Kollisionstests auf Silhouetten zu lösen, besteht in einer Rekonstruktion des Arbeitsraumes. Aus den Silhouettenbildern wird mittels einer Rückprojektion der Silhouetten die Belegung des Arbeitsraumes rekonstruiert. Dieses Verfahren ist auch unter den Namen „Shape from Silhouette“, „Volume Intersection“ oder „Volumenschnitt“ bekannt. Es wird in der Computergrafik verwendet, um räumliche Objekte in den Rechner einzuscannen.

Die Anwendung des Volumenschnittes zur Objektrekonstruktion wird zum Beispiel in [Eckert00], [Eisert00] und [Hamfeld02] vorgestellt. Allerdings liegt bei diesen Anwendungen der Hauptaugenmerk auf der möglichst exakten Rekonstruktion eines einzelnen Objektes. In der Anwendung sind jedoch mindestens zwei Objekte, Mensch und Roboter, im Arbeitsraum vorhanden. Die bei der Rekonstruktion eines einzelnen Objektes auftretenden Fehler werden in [Niem97] untersucht.

Die meisten Autoren aus dem Bereich der Computergrafik betrachten nur die Situation mit einem einzigen Objekt im Sichtbereich der Kameras. Im Gegensatz dazu wird in [Noborio92] ein Volumenschnittverfahren zur Rekonstruktion mehrerer Objekte verwendet. Allerdings wird die Annahme getroffen, dass die verschiedenen Sichtkegel auf Grund von Farbinformation den einzelnen Objekten zugeordnet werden können. Diese Annahme setzt voraus, dass die Objekte aus Sicht der verschiedenen Kameras die gleiche Farbe tragen. Bei kleinem Abstand der Kameras zueinander ist diese Annahme erfüllt, allerdings sind dann die Rekonstruktionsfehler der Objekte relativ groß, da die Sichtkegel der verschiedenen Kameras schleifend geschnitten werden. Bei großem Ba-

sisabstand wird der Rekonstruktionsfehler kleiner, aber dann kann bei natürlichen Objekten nicht mehr erwartet werden, dass alle Kameras die gleiche Farbe sehen.

Ein Verfahren zur Kollisionserkennung bei einem Portalroboter wird in [Meisel94] vorgestellt. Dabei wird ein Gitter aus Raumpunkten definiert, die auf Veränderungen überwacht werden sollen. Diese Überwachungsraumpunkte werden in die Bildebenen der Kameras transformiert. Die resultierenden Überwachungsbildpunkte werden auf Veränderung überprüft. Aus der Veränderung der Überwachungsbildpunkte wird auf die Veränderung der Überwachungsraumpunkte geschlossen.

Ein anderer Ansatz wird in [Leou91] beschrieben. Anstatt den Arbeitsraum zu rekonstruieren, wird die Kollisionserkennung ausschließlich im Bildraum der Kameras durchgeführt. Dort wird allerdings das Problem der Kollisionsvermeidung bei mehreren Robotern in einem gemeinsamen Arbeitsraum betrachtet. Der gemeinsame Arbeitsraum wird durch mehrere stationäre Kameras überwacht. Die Kanten der Roboter sind so markiert, dass sie leicht in den Kameras zu finden sind. In den Kamerabildern werden Linienmerkmale gesucht, die den Robotersegmenten zugeordnet werden. Die erwartete zukünftige Position der Segmente wird in den Bildern aus der aktuellen und aus vorhergehenden Positionen berechnet. Die Liniensegmente der Roboterkonturen werden auf Schnitt untersucht. Werden in allen Kameras Kollisionen erkannt, hält das System die Roboter an. Abbildung 47 zeigt das System von [Leou91].

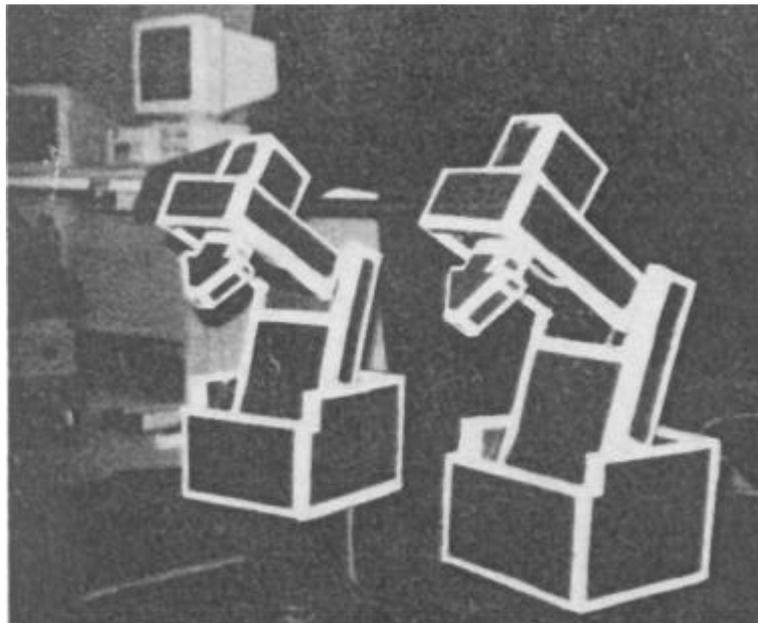


Abbildung 47: System zur bildbasierten Kollisionserkennung von [Leou91].

Das in [Leou91] vorgestellte Verfahren hat mehrere Schwächen. Die Zuordnung der Segmente zu den verschiedenen Robotern und die Identifizierung der einzelnen Segmente ist sehr aufwändig. Zudem muss die Bewegung der einzelnen Segmente in den Bildern erfasst werden. Dazu müssen die Bilder der Segmente über mehrere Aufnahmen hinweg korrekt zugeordnet werden. Der Aufwand der Kollisionserkennung wächst mit der Anzahl der Liniensegmente. Jede Kombination von zwei Segmenten der beiden Roboter muss untersucht werden. In einer natürlichen Szene würden sehr viele Kanten gefunden werden. Vor allem ein Mensch hat sehr viele Kanten durch Muster und Faltenwurf der Kleidung. Eine hinreichend korrekte Zuordnung wäre kaum möglich und der Kollisionstest wäre zu aufwändig. Vor allem aber kann das System nur zur Geschwindigkeitsregelung verwendet werden, es ist nicht möglich, Ausweichbahnen zu generie-

ren, da kein Wissen über die Robotergeometrie vorliegt. Auf das Problem der Mensch-Roboter-Koexistenz angewendet, würde das System in die Kategorie *GE/GR* fallen.

In [Quick96] wird die Methode der bildbasierten Kollisionsvermeidung diskutiert. Es wird von stationären Kameras ausgegangen, die den Arbeitsraum überwachen. Die auf Grund des Roboters belegten Pixel müssen herausgerechnet werden, es wird aber nicht gesagt wie. Der Tool-Center-Point des Roboters wird in die Bilder transformiert. Die Pixel in einem bestimmten Abstand zum projizierten TCP werden auf den Belegungszustand untersucht. Aus dem Belegungszustand wird eine abstoßende, virtuelle Kraft berechnet, die den TCP verschiebt. Bei mehreren Kameras wird vorgeschlagen, die einzeln berechneten Kräfte zu überlagern. Es ist unklar, in wie weit die theoretischen Ideen in die Praxis umgesetzt worden sind. Ebenso ist unklar, ob das Problem der vom Roboter verdeckten Hindernisse erkannt und gelöst wurde. Der gewählte Ansatz der Planung einer Ausweichbewegung für den TCP mittels eines Potentialfeldes fällt in die Kategorie *EE/LA*.

5.2 Kollisionserkennung im Bildraum der Kameras

Die Kernidee ist, eine Kollisionserkennung im Bildraum der Kameras durchzuführen. Diese Idee nutzt aus, dass zwei Objekte im Raum nicht miteinander kollidieren können, wenn es eine Ansicht gibt, aus der die Bilder der Objekte nicht kollidieren.¹³ Anders als in [Leou91] und [Quick96] soll eine globale Ganzarmkollisionserkennung realisiert werden.

Wenn das Bild des Roboters in einer beliebigen Konfiguration berechnet werden kann, ist es möglich, eine beliebige Konfiguration dadurch auf Kollision zu testen, dass das berechnete Bild des Roboters auf Kollisionen mit dem aktuellen Bild der Hindernisse untersucht wird. Die Rekonstruktion des gesamten Konfigurationsraums kann, sofern notwendig, durch das Testen von allen möglichen Konfigurationen erfolgen.

5.2.1 Formale Beschreibung der Kollisionserkennung

Die Kameras bilden den Arbeitraum A auf die Kameraebene E_k ab. Bei einer linearen Kameraabbildung gilt, dass ein Punkt P auf das Pixel p des Binärbildes B_k abgebildet wird, welches von der Geraden $g(P, z_k)$ geschnitten wird, die durch P und das optische Zentrum z_k der Kamera k gebildet wird. Diese Abbildung wird im Folgenden Kameraabbildung KA genannt:

$$KA(P, k): P \in A \rightarrow p \in B_k : g(P, z_k) \cap p \neq \emptyset \quad (25)$$

Jedes Pixel p des Binärbildes B_k der Kamera k beschreibt zusammen mit dem optischen Zentrum z_k der Kamera einen Raumbereich, der ein verallgemeinerter Kegel ist. Dieser Sichtkegel, im Folgenden auch mit dem englischen Begriff *vision cone* (*VC*) bezeichnet, wird definiert als:

$$VC(p, k) = \{ v \in V \mid \exists P \in v : KA(P, k) = p \} \quad (26)$$

¹³ Die Umkehrung gilt im allgemeinen nicht. Wenn die Bilder zweier (nichtkonvexer) Objekte aus jeder möglichen Ansicht kollidieren, ist nicht garantiert, dass die Objekte im Raum kollidieren. Beispiel: Eine Kugel, die in einem Becher schwebt.

Für die Sichtkegel einer Kamera k gelten folgende Beziehungen:

$$\text{Vollständigkeit} \quad \bigcup_{p \in B_k} VC(p, k) = V \quad (27)$$

Die Zustandsfunktion $Z(p)$ eines Pixels p des Binärbildes B_k ist nur dann 0, wenn alle Voxel des zugehörigen Sichtkegels zum Freiraum gehören:

$$Z(p) = \begin{cases} 0; & \forall v \in VC(p, k) : v \in V_F \\ 1; & \text{sonst} \end{cases} \quad (28)$$

Eine Silhouette ist die Teilmenge der Pixel eines Binärbildes mit Zustand 1.

$$S_k = \{ p \in B_k \mid Z(p) = 1 \} \quad (29)$$

Als abkürzende Schreibweise wird $S_k(V)$ für die Silhouette der Kamera k des Voxerraumes V geschrieben. Für die Silhouette des zur Konfiguration c gehörenden Roboter- raumes V_R wird abkürzend $S_k(c) = S_k(V_R(c))$ geschrieben.

5.2.2 Grundalgorithmus

Der Grundalgorithmus verwendet als Eingabe die Silhouette des Roboters in der zu testenden Konfiguration c_T im Folgenden *Testsilhouette* $S(c_T)$ genannt und die Silhouette der aktuellen Hindernisse, die *Hindernissilhouette* $S(V_O)$. Der Roboter ist in der Hindernissilhouette nicht enthalten.

Wenn es eine Kamera k gibt, in der die Test- und die Hindernissilhouette kein gemeinsames Pixel haben, dann ist die Konfiguration c_T Teil des freien Konfigurationsraums C_F .

$$\exists k \in \{0, \dots, (l-1)\} : S_k(V_O) \cap S_k(c_T) = \emptyset \Rightarrow c_T \in C_F \quad (30)$$

Beweis durch Widerspruch:

Eine Konfiguration c_T führt zur Kollision mit einem Objekt:

$$c_T \in C_O \quad (31)$$

Trotzdem gelte für eine beliebig gewählte Kamera $k \in \{0, \dots, (l-1)\}$:

$$S_k(V_O) \cap S_k(c_T) = \emptyset \quad (32)$$

Aus (31) folgt nach (11)

$$V_R(c_T) \cap V_O = V_{RO} \neq \emptyset \quad (33)$$

Sei $v \in V_{RO}$ aus der Schnittmenge, dann gilt nach (6), dass

$$Z_O(v) = Z_R(v) = 1 \quad (34)$$

Weiterhin gilt nach (27), dass mindestens ein Pixel $p \in B_k$ existiert mit $v \in VC(p,k)$

Aus (34) folgt durch (28)

$$p \in S_k(V_O) \quad (35)$$

und

$$p \in S_k(V_R(c_T)) \quad (36)$$

Daraus folgt

$$p \in S_k(V_O) \cap S_k(c_T) \quad (37)$$

Damit ist die Schnittmenge $S_k(V_O) \cap S_k(c_T)$ nicht leer. Dies steht im Widerspruch zur Annahme.

Damit ist bewiesen, dass die Konfiguration c_T kollisionsfrei ist, wenn die Test- und die Hindernissilhouette in mindestens einer Kamera kein gemeinsames Pixel haben.

Algorithmische Umsetzung

Aus technischer Sicht ist es meist einfacher mit Feldern fester Länge zu operieren, anstatt mit Mengen, die eine variable Anzahl von Elementen beinhalten. Aus diesem Grund wird in der Pseudocodedarstellung, die in Algorithmus 1 zu sehen ist, jedes Pixel überprüft, ob es in beiden Silhouetten gesetzt („true“) ist.

```

bool collisionTest(configuration ctest, silhouette obstacleSilhouette)
{
    silhouette testSilhouette = GetSilhouetteFromConfiguration(ctest)

    FOR camera = 0 TO maxcamera-1
        bool collision = false
        FOR row = 0 TO maxrow
            FOR column = 0 TO maxcolumn
                IF ( obstacleSilhouette[kamera][column][row] AND
                    testSilhouette[kamera][column][row] )
                    collision = true
                END IF
            END FOR column
        END FOR row
        IF NOT(collision) THEN RETURN true
    END FOR camera
    RETURN false
END collisionTest

```

Algorithmus 1: Grundalgorithmus zum Testen einer Konfiguration

Beispiel

Gegeben seien die Bilder, wie sie in Abbildung 48 zu sehen sind. Zur einfacheren Visualisierung sind die Hintergrundpixel (B : Background) und die Vordergrundpixel (O : Objekt bzw. R : Roboter) farblich und textuell unterschieden. Im Beispiel haben Hin-

dernis- und Testsilhouette ein gemeinsames Vordergrundpixel. Wenn keine weiteren Bilder von anderen Kameras vorliegen, kommt der Algorithmus zum Ergebnis, dass die Testkonfiguration zu einer Kollision mit einem Hindernis führen würde.

Abbildung 48: Beispiel zur Anwendung des Grundalgorithmus. Hindernissilhouette (links), aus der Testkonfiguration ermittelte Testsilhouette (Mitte) und Schnittergebnis (rechts). Zur einfacheren Visualisierung sind die Hintergrundpixel (B : Background) und die Vordergrundpixel (O : Objekt bzw. R : Roboter) farblich und textuell unterschieden. Wenn keine weiteren Kameras existieren, erkennt der Algorithmus im Beispiel eine Kollision.

5.2.3 Verfahren für Szenen mit Roboter

Normalerweise enthält die von den Kameras aufgenommene Szene den Roboter. Damit nun die Kollisionserkennung mit dem vorgestellten Algorithmus erfolgen kann, muss zuerst aus der *Szenensilhouette* die Hindernissilhouette ermittelt werden. Da in den Silhouetten keine Unterscheidung zwischen Roboter und Objekt möglich ist, muss auf Zusatzwissen über die Roboterform zurückgegriffen werden.

Mit Hilfe des Robotermodells kann eine Silhouette erzeugt werden, die den Roboter in der aktuellen Konfiguration zeigt. Diese *Robotersilhouette* wird zur Unterscheidung zwischen Roboter und Objekt verwendet.

Der erste Schritt in der Erzeugung der Hindernissilhouette ist der Aufbau eines Zwischenbildes mit Hilfe der in Tabelle 9 aufgezeigten Abbildungsvorschrift. Da die Pixel von Szenen- und Robotersilhouette jeweils zwei Zustände besitzen, können insgesamt vier Zustandskombinationen auftreten.

Szenensilhouette	Robotersilhouette	Zwischenbild
B	B	B
B	R	E
F	B	O
F	R	U

Tabelle 9: Abbildungsvorschrift zur Erzeugung des Zwischenbildes aus Szenen- und Robotersilhouette. B – Background (Hintergrund), F – Foreground (Vordergrund), R – Roboter, O – Objekt, E – Error (Fehler), U – Unbekannt.

Es werden jeweils die Zustände der Pixel an der gleichen Bildposition betrachtet. Wenn sowohl in der Szenensilhouette als auch in der Robotersilhouette das Pixel gelöscht ist, wird angenommen, dass an dieser Stelle der Bildhintergrund gesehen wird und der Zustand des entsprechenden Pixels im Zwischenbild auf „B“ für Background gesetzt wird.

Eigentlich dürfte die Zustandskombination, dass das Pixel in der Szenensilhouette gelöscht und in der Robotersilhouette gesetzt ist, nicht vorkommen. Bei einer idealen Robotermodellierung würde überall dort, wo der Roboter sein müsste, auch der Roboter gesehen und daher das entsprechende Szenenpixel gesetzt sein. Falls diese Zustandskombination dennoch auftritt, wird der Zustand des Ergebnispixels auf „E“ für Error gesetzt.

Ist ein Pixel in der Szenensilhouette gesetzt, in der Robotersilhouette aber nicht, sieht man an dieser Stelle ein Objekt. Daher wird der Zustand des Ergebnispixels auf „O“ für Objekt gesetzt.

Ist ein Pixel sowohl in der Szenensilhouette als auch in der Robotersilhouette gesetzt, sieht man an dieser Stelle des Bildes immer den Roboter. Allerdings könnte aus Blickrichtung vor oder hinter dem Roboter noch ein weiteres Objekt stehen. Aus diesem Grund wird der Zustand des Ergebnispixels auf „U“ für unbekannt gesetzt.

Das Beispiel zur Erzeugung der Zwischenbilder ist in Abbildung 49 zu sehen.

B	B	B	B	B	B	F	F	B	B	B	B	B	B	B	B	B	B	B	B	B	B	O	O
B	B	B	B	B	F	F	F	B	B	B	B	B	B	B	B	B	B	B	B	O	O	O	
B	B	B	F	F	F	F	B	B	B	R	R	R	R	B	B	B	B	U	U	U	U	B	
B	B	F	F	F	B	B	B	B	B	R	R	R	B	B	B	B	B	B	U	U	U	B	
B	F	B	F	B	B	F	F	B	R	R	R	B	B	B	B	U	E	U	B	B	O	O	
F	F	F	B	B	B	F	F	R	R	R	B	B	B	B	U	U	U	B	B	B	O	O	
F	F	F	B	B	B	B	B	R	R	R	B	B	B	B	U	U	U	B	B	B	B	B	
F	F	F	B	B	B	B	B	R	R	R	B	B	B	B	U	U	U	B	B	B	B	B	

Abbildung 49: Beispiel zur Erzeugung des Zwischenbildes. Aus der Szenensilhouette (links) wird mit Hilfe der Robotersilhouette (Mitte) das Zwischenbild (rechts) berechnet.

Damit der Grundalgorithmus zur Kollisionserkennung verwendet werden kann, muss das Zwischenbild in eine Hindernissilhouette umgewandelt werden. Während die Pixel, welche die Zustände „B“ oder „O“ haben, direkt übernommen werden können, müssen die Pixel mit den Zuständen „E“ und „U“ in „B“ oder „O“ umgewandelt werden.

Die Pixel mit dem Zustand „E“ können dadurch auftreten, dass das Robotermodell den Roboter konservativ approximiert. In diesem Fall wäre eine Interpretation als Hintergrund „B“ korrekt. Andererseits besteht die Möglichkeit, dass durch einen Fehler in der Bildverarbeitung, der Roboter nicht korrekt erkannt wurde. Trotzdem kann ein Hindernis durch den Roboter verdeckt werden. In diesem Fall wäre die Interpretation als Unbekanntes Pixel „U“ korrekt.

Im realisierten System werden Pixel mit dem Zustand „E“ in Hintergrund „B“ umgewandelt. Dies ist insofern vertretbar, als dass einzelne Hintergrundpixel in der Mitte des Roboters das Ergebnis des Kollisionstests nicht beeinflussen. Error-Pixel an den Rändern des Roboters kommen durch die konservative Approximation des Roboters durch das Robotermodell zustande.

5.2.4 Unkalibrierte Kameras

Wenn das System nur eine einzige Kamera hat oder die Kameras nicht kalibriert sind, bleibt als einzige Möglichkeit, Pixel mit den Zuständen „E“ und „U“ zu ignorieren (auf Background „B“ zu setzen). Dies führt allerdings dazu, dass Hindernisse, die vom Roboter verdeckt sind, nicht wahrgenommen werden können. Dies kann zu Kollisionen führen.

Stehen mehrere Kameras zur Verfügung, kann der Grundalgorithmus zur Kollisionserkennung dahingehend modifiziert werden, dass nicht nur die Hindernis- und Testsilhouette einer Kamera, sondern mehrerer Kameras überschneidungsfrei sein müssen.

Die Anzahl der Kameras, deren Hindernis- und Testsilhouette überschneidungsfrei sein müssen, bestimmt sich nach der Anzahl der Kameras, aus deren Sicht der Roboter ein bestimmtes Voxel verdecken kann. Nach Gleichung (27) wird jedes Voxel v in jeder Kamera auf mindestens ein Pixel p_k abgebildet. Ist $Z_O(v) = 1$ folgt daraus $Z(p_k)=1$ für die Szenensilhouetten aller l Kameras. Wenn der Roboter in der aktuellen Konfiguration aus Sicht von $d < l$ Kameras ebenfalls auf das Pixel p abgebildet wird, bleiben noch $l-d$ Kameras, die zur Kollisionserkennung beitragen können. Der Nachteil dieser Vorgehensweise ist, dass der Fehler 2. Art (fälschlich als kollidierend erkannte Konfigurationen) stark ansteigt.

Um unterschiedliche Kameraanzahlen zu berücksichtigen, wird der Schwellwert q_{Cam} im Folgenden definiert als die Anzahl der Kameras, die einen leeren Schnitt erkennen müssen, damit eine Konfiguration als frei klassifiziert wird.

Im realisierten System wird beim Algorithmus für Szenen mit Roboter ein Schwellwert q_{Cam} von 2 verwendet. Das heißt, dass mindestens zwei Kameras keinen Schnitt erkennen dürfen, damit eine Konfiguration als frei klassifiziert wird. Dies geht von der Hypothese aus, dass der Roboter kein Voxel des Raumes aus der Sicht von mehr als einer Kamera verdecken kann.

5.2.5 Kalibrierte Kameras

Besitzt das System mehrere Kameras und sind diese kalibriert, können die Zwischenbilder der anderen Kameras verwendet werden, um die Unbekannten und Fehlerpixel eines Zwischenbildes aufzulösen. Die resultierende Hindernissilhouette kann durch den Grundalgorithmus (Algorithmus 1) mit dem Schwellwert $q_{\text{Cam}} = 1$ bearbeitet werden.

Dadurch lässt sich die Anzahl der im Mittel zu testenden Silhouetten im Vergleich zum vorher beschriebenen Verfahren für unkalibrierte Kameras verkleinern. Dies ist am Beispiel für drei Kameras in Tabelle 10 dargestellt. Unter der Voraussetzung, dass jede Kamerabelegung gleich wahrscheinlich ist und dass jeder Schnitttest gleich lang dauert, ist der Aufwand für die Kollisionserkennung mit zwei freien Kameras im Beispiel um fast 43% größer als die Kollisionserkennung, wenn nur eine Kamera frei sein muss.

Beitrag von Kamera Nr.			Anzahl notwendiger Schnittteste	
0	1	2	$q_{\text{Cam}} = 1$	$q_{\text{Cam}} = 2$
0	0	0	1	2
0	0	1	1	2
0	1	0	1	3
0	1	1	1	3
1	0	0	2	3
1	0	1	2	3
1	1	0	3	2
1	1	1	3	2
Mittel			1,75	2,5

Tabelle 10: Notwendige Anzahl an Schnitttests. Jede Kamera kann als Beitrag 0 (leerer Schnitt) oder 1 (nicht leerer Schnitt) liefern. Für alle 8 Möglichkeiten wird bestimmt, nach dem wievielten Schnitttest (mit Kamera 0 beginnend) feststeht, ob die getestete Konfiguration in C_F oder C_O liegt.

Für jedes Pixel eines Bildes wird die Epipolarlinie¹⁴ in den anderen Bildern berechnet. Die Epipolarlinie EP in der Kamera l zu einem Pixel p der Kamera k ist die Menge aller Pixel aus l , deren Sichtkegel mindestens ein gemeinsames Voxel mit dem Sichtkegel des Pixel p hat.

$$EP(p,k,l) = \{ q \in B_l \mid VC(q,l) \cap VC(p,k) \neq \emptyset \} \quad (38)$$

Für jedes Pixel p des Zwischenbildes B_k der Kamera k mit dem Zustand „U“ werden die Epipolarlinien in allen anderen Kameras untersucht. Wenn in mindestens n Kameras Pixel mit dem Zustand „O“ zu finden sind, wird das entsprechende Pixel p der erzeugten Hindernissilhouette hinzugefügt. Zusätzlich sind alle Pixel des Zwischenbildes mit dem Zustand „O“ in der Hindernissilhouette S enthalten.

$$S = \{ p \in B_k \mid Z(p) = O \} \cup \{ p \in B_l \mid Z(p) = U \wedge |\{ m \in \{0, \dots, (l-1) : \exists q \in EP(p,k,m) : Z(q) = O \}| \geq n \} \} \quad (39)$$

Im realisierten System wird $n = l-1$ gewählt. Dahinter steht die Annahme, dass der Roboter jedes Voxel maximal aus Sicht einer Kamera verdecken kann. Trifft diese Annahme zu und wird ein Voxel des Objekts vom Roboter verdeckt, dann ist mit Sicherheit in allen anderen Kameras dieses Voxel des Objektes auf der zugehörigen Epipolarlinie zu sehen.

¹⁴ Der Begriff Epipolarlinie bzw. epipolar line wird hier verwendet, auch wenn die beschriebene Struktur keinen Linien- sondern Flächencharakter hat.

5.3 Vergleich mit der Rekonstruktion des Arbeitsraums

Es stellt sich die Frage, ob durch den Verzicht auf die Rekonstruktion des Arbeitsraumes die Qualität des Ergebnisses schlechter wird. Um diese Frage zu beantworten, wird untersucht, in wie weit die Rekonstruktion beziehungsweise ein Verzicht darauf Einfluss auf das Ergebnis hat.

Ein generischer Kollisionstest mit Arbeitsraumrekonstruktion in ein Voxelmmodell besteht aus den folgenden Schritten:

1. Rückprojektion der Silhouetten in den Voxerraum
2. Klassifizierung der einzelnen Voxel mit Hilfe des 3D-Robotermodells in Hintergrund, Hindernis und Roboter
3. Schnitt des Roboters in der Testkonfiguration mit allen Hindernis-Voxeln. Ist die Schnittmenge nicht leer, gibt es eine Kollision.

Da als Eingabe nur binäre Silhouetten verwendet werden, kann keine Korrespondenz zwischen den Bildpunkten der verschiedenen Kameras gefunden werden. Dies verhindert auch die Anwendung von Verfahren zur Rekonstruktion von Konkavitäten, wie sie in [Eckert00] beschrieben wurden. Auch kann der Roboter nicht anhand seines Aussehens direkt erkannt werden, sondern ein Robotermodell ist zwingend erforderlich.

Durch die Rückprojektion können die Hindernisse je nach verwendeter Anzahl und Position der Kameras größer erscheinen, als sie in Wirklichkeit sind. Außerdem werden die Hindernisse als konvexe Polyeder approximiert und es können zusätzliche Hindernisse auftreten. Diese Probleme sind in Abbildung 50 dargestellt.

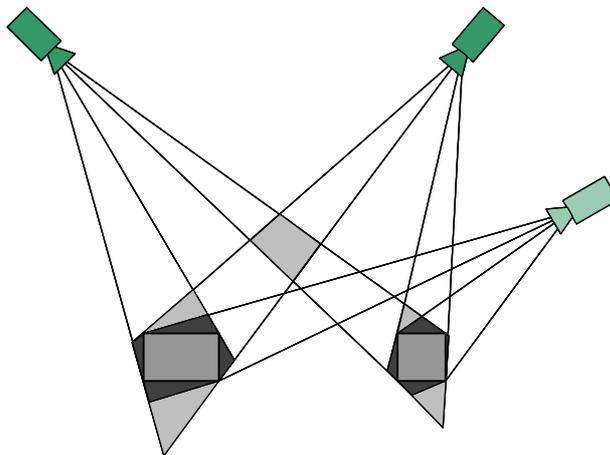


Abbildung 50: Darstellung der Probleme der Objektvergrößerung und Phantomobjekte im 2-dimensionalen Raum. (ähnlich [Meisel94])

Durch das bilderzeugende Robotermodell werden die Rekonstruktionsfehler automatisch berücksichtigt. Wenn die Vergrößerung des Roboters bei einer Kollisionserkennung mit Arbeitsraumrekonstruktion nicht berücksichtigt würde, wäre nach dem Herausrechnen des Roboters der Raum um den Roboter vollständig belegt. Diese Problematik ist in Abbildung 51 dargestellt.

Ein Verfahren mit Rekonstruktion des Arbeitsraumes hat den Nachteil, dass die Objektvergrößerung des Roboters durch den Rekonstruktionsfehler explizit modelliert werden muss.

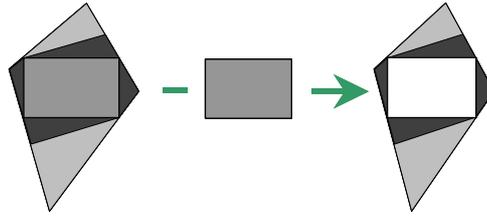


Abbildung 51: Notwendigkeit der Berücksichtigung des Rekonstruktionsfehlers in der Robotermodellierung. Der Roboter sei das dunkle Rechteck. Die schraffierte Fläche entsteht durch die Rekonstruktionsfehler. Werden diese nicht berücksichtigt, scheint der Roboter vollständig von Hindernissen umgeben und kann nicht bewegt werden.

Der Vorteil eines Verfahrens mit Rekonstruktion des Arbeitsraumes scheint darin zu bestehen, dass beim Schnitt des Roboters in der Testkonfiguration keine Objektvergrößerung des Roboters auftritt. Dadurch könnte der Roboter enger an den Hindernissen vorbeifahren und sich in dichter besetzten Räumen bewegen, als dies bei der bildbasierten Methode der Fall ist.

Allerdings kann der Test mit einem Roboter, ohne Berücksichtigung der Objektvergrößerung, unter Umständen zu Kollisionen führen. Angenommen, der Roboter soll in eine Position fahren, die zu einer Kollision führen würde. Da der Roboter beim eigentlichen Kollisionstest sehr exakt modelliert wird, kann er sehr nah an das Hindernis heranzufahren. Bei der Klassifizierung des Roboters muss, wie oben beschrieben, der Rekonstruktionsfehler berücksichtigt werden. Dabei kann es vorkommen, dass der rekonstruierte Roboter in das tatsächliche Hindernis hineinragt. In der Abbildung 52 ist ein solcher Fall dargestellt. Bei hinreichend feiner Diskretisierung des Raumes wird mit jedem Schritt ein zunehmender Teil des Hindernisses dem Roboter zugerechnet, bis es zur Kollision kommt.

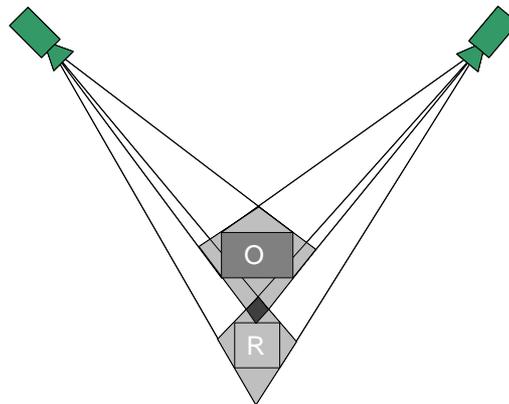


Abbildung 52: Zeichnung zum Problem des Kollisionstests mit Roboter ohne Rekonstruktionsfehler. Das reale Objekt sei das obere, dunkelgraue Rechteck. Das rekonstruierte Objekt ist durch den schraffierten Bereich dargestellt. Der Roboter ist das untere, hellgraue Rechteck. Auf Grund des Tests ohne Berücksichtigung des Rekonstruktionsfehlers kann der Roboter direkt an das rekonstruierte Hindernis heranzufahren. Dabei überlappen die rekonstruierten Volumen von Objekt und Roboter (dunkelkarierter Bereich). In der folgenden Iteration wird der überlappende Bereich dem Roboter zugerechnet und der Roboter kann dichter an das Objekt heranzufahren. Dies kann schließlich zur Kollision mit dem realen Objekt führen.

Folglich muss auch bei Verfahren mit Rekonstruktion des Arbeitsraumes der Schnitttest mit einem Robotermodell durchgeführt werden, das die Rekonstruktionsfehler berücksichtigt. Damit bringt die Rekonstruktion des Arbeitsraumes unter den gegebenen Annahmen keinen Vorteil.

Dieses Ergebnis ändert sich dann, wenn Korrespondenzen zwischen den verschiedenen Bildern gefunden werden können und auf diese Weise eine Unterscheidung zwischen verschiedenen Hindernissen möglich ist. Damit die Korrespondenz zuverlässig gefunden werden kann, darf der Basisabstand und die Winkeldifferenz der Kameras nicht zu groß sein. Damit steigt die Zahl der erforderlichen Kameras stark an.

5.4 Experimente

Die Ergebnisqualität der Kollisionstests hängt stark von der Qualität der verwendeten Silhouetten von Roboter und Szene ab. Im Grenzfall kann bereits ein einziges falsches Pixel das Ergebnis des Kollisionstests verändern. Um die Eigenschaften der Kollisionstests ohne den Einfluss der Bildverarbeitung und der Robotermodellierung zu untersuchen, werden die Kollisionstests in einem Simulator getestet.

5.4.1 Versuchsaufbau

Eine Darstellung der simulierten Roboterzelle ist in Abbildung 53 zu sehen. Sowohl der Roboter als auch das Hindernis werden durch Kugeln dargestellt. Jedes Gelenk des Roboters wird durch fünf in gleichem Abstand entlang des Gelenks angeordneten, überlappende Kugeln approximiert.

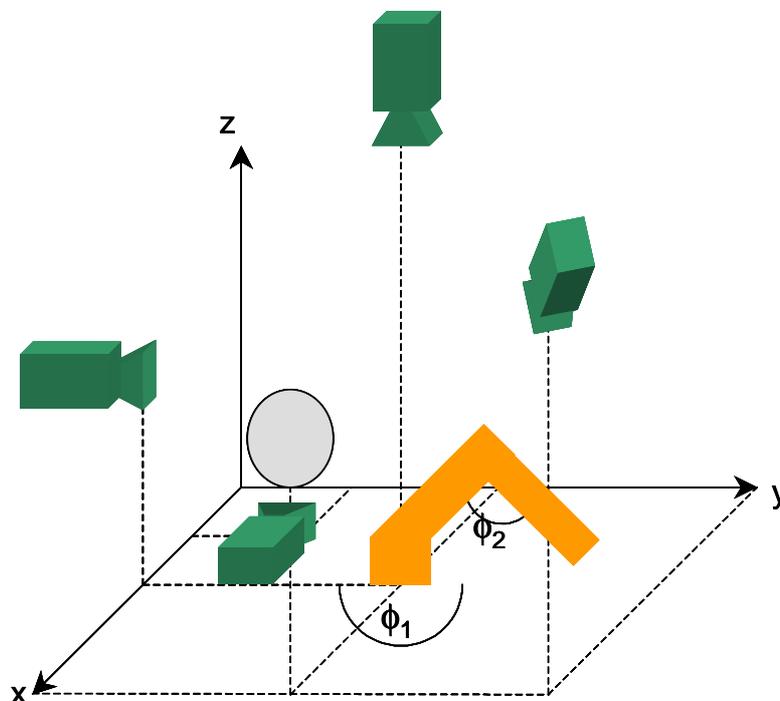


Abbildung 53: Arbeitszelle in der Simulation. Der Roboter besitzt zwei Freiheitsgrade f_1 und f_2 . Sowohl Roboter als auch Hindernis werden durch Kugeln approximiert.

Die Kugeln werden mittels einer Zentralprojektion in die Ebenen der Kameras auf Kreise abgebildet. Die Kreise werden entsprechend der Kameraauflösung in eine Pixeldarstellung umgewandelt. Der Roboterarm kann sich in zwei Gelenken f_1 und f_2 jeweils um 360° drehen. Interne Kollisionen oder Kollisionen mit dem Boden der Arbeitszelle werden nicht betrachtet. Der Konfigurationsraum besteht aus 32×32 Konfigurationen.

5.4.2 Bewertung der Kollisionserkennung

Ein Gütemaß zur Bewertung der Kollisionserkennung ist die Anzahl der falsch klassifizierten Roboterkonfigurationen in einer gegebenen Szene, die von einer gegebenen Kamerakonfiguration abhängt. Eine Kamerakonfiguration definiert alle Parameter, die für die Bildverarbeitung relevant sind, zum Beispiel: Anzahl der Kameras, die Positionierung im Raum und die Einstellung des Öffnungswinkels (Zoom).

Die beiden möglichen Fehlerarten sind *fehlende Konfigurationen* (Fehler 1. Art) und *zusätzliche Konfigurationen* (Fehler 2. Art). Fehlende Konfigurationen werden von dem untersuchten Verfahren zur Kollisionserkennung als nichtkollidierend klassifiziert. Tatsächlich findet aber eine Kollision statt. Zusätzliche Konfigurationen werden als kollidierend klassifiziert, sind in Wirklichkeit aber nichtkollidierend.

5.4.3 Versuchsdurchführung

Insgesamt werden vier Kollisionstests verglichen.

Der Referenztest untersucht direkt die Kugeln von Hindernis und Roboter auf Schnitt. Dieser Test dient zum Ermitteln des tatsächlich belegten und freien Raumes und damit als Indikator für Fehler 1. Art und Fehler 2. Art der anderen Kollisionstests.

Der zweite Test realisiert den Grundalgorithmus. Hierzu wird als Eingabe ein Bild verwendet, in das nur die Hinderniskugeln projiziert wurden. Dieser Test dient zur Messung des Einflusses der Verdeckungen von Hindernissen durch den Roboter.

Der dritte Test realisiert den Algorithmus für Szenen mit Robotern bei unkalibrierten Kameras. Der Kollisionstest erhält als Eingabe eine Silhouette, in welche die Kugeln von Roboter und Hindernis projiziert wurden.

Der vierte Test realisiert den Algorithmus für Szenen mit Robotern bei kalibrierten Kameras. Der Kollisionstest erhält als Eingabe eine Silhouette, in welche die Kugeln von Roboter und Hindernis projiziert wurden. Durch die oben beschriebene Methode mit Hilfe der Epipolarlinien versucht der Kollisionstest die Hindernissilhouette zu rekonstruieren.

Insgesamt wurden 14 verschiedene Kamerakonfigurationen untersucht, jeweils 4 Konfigurationen mit 3 bzw. 4 Kameras und 6 Konfigurationen mit 5 Kameras.

Der Schwellwert q_{Cam} gibt beim Verfahren für Hindernissilhouette und Szenensilhouette an, wie viele Kameras einen leeren Schnitt erkennen müssen, damit die getestete Konfiguration als frei klassifiziert wird.

Beim Verfahren mit Silhouettenrekonstruktion gibt der Schwellwert an, wie viele Kameras bei der Rekonstruktion der Hindernissilhouette aus dem Zwischenbild keinen Beitrag liefern müssen, damit das untersuchte Pixel dennoch als Objekt klassifiziert wird. Bei dieser Anzahl wird die untersuchte Kamera mitgezählt. Ein Schwellwert von 1

bedeutet daher, dass auf den Epipolarlinien von allen anderen Kameras ein Objekt zu sehen sein muss. Bei diesem Verfahren wird beim Kollisionstest immer ein Schwellwert von 1 verwendet.

Die mittlere Anzahl von fehlenden und zusätzlichen Konfigurationen in Abhängigkeit von Kameraanzahl und Schwellwert ist in Tabelle 11 dargestellt.

		Schwellwert q_{Cam}											
		1			2			3			4		
Anzahl Kameras		3	4	5	3	4	5	3	4	5	3	4	5
Zusätzliche Konfigurationen	Hindernissilhouette	74,5	59,0	61,7	126,0	101,5	86,3	295,0	156,0	114,3	-	253,4	154,0
	Szenensilhouette	22,5	16,5	28,7	79,5	61,5	56,3	259,0	129,0	89,0	-	235,0	131,0
	Rekonstruierte Silhouette	22,5	16,5	28,7	69,8	102,3	94,7	148,3	127,5	146,0	-	373,5	166,5
Fehlende Konfigurationen	Hindernissilhouette	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	-	0,0	0,0
	Szenensilhouette	9,0	11,0	18,7	0,0	0,0	1,3	0,0	0,0	0,7	-	0,0	0,0
	Rekonstruierte Silhouette	9,0	11,0	18,7	0,0	0,0	1,3	0,0	0,0	0,0	-	0,0	0,0

Tabelle 11: Einfluss von Kameraanzahl und Schwellwert auf die Qualität der Kollisionserkennung. Dargestellt ist die mittlere Anzahl der fehlenden bzw. zusätzlichen Konfigurationen im Vergleich zu einem Referenztest, der auf dem Schnitt von Kugeln arbeitet.

Das Ergebnis des Versuches zeigt, dass mit steigendem Schwellwert bei allen Verfahren die Anzahl der zusätzlichen Konfigurationen steigt. Dieses Ergebnis entspricht der Erwartung. Wenn der Roboter in der Nähe des Hindernisses ist, werden nur wenige Kameras die verbleibende Lücke erkennen. Je größer der Schwellwert ist, desto weiter muss der Roboter vom Hindernis entfernt sein, damit eine genügend große Anzahl von Kameras die Lücke sehen können.

Weiterhin ist zu erkennen, dass mit steigender Anzahl der Kameras die Anzahl der zusätzlichen Konfigurationen im allgemeinen abnimmt. Es gibt jedoch auch Ausnahmen, zum Beispiel einen Anstieg beim Verfahren mit rekonstruierter Hindernissilhouette beim Übergang von 3 auf 4 Kameras und dem Schwellwert 2.

Die Anzahl der fehlenden Konfigurationen ist beim Verfahren mit Hindernissilhouette stets 0. Dies entspricht der Erwartung, da in der Silhouette der Roboter nicht enthalten ist. Daher können auch keine Verdeckungen auftreten.

Die Anzahl der fehlenden Konfigurationen bei den Verfahren mit Szenensilhouette und rekonstruierter Silhouette ist sehr ähnlich. Wenn der Roboter ein Voxel aus der Sicht von maximal k Kamera verdecken kann, ist zu erwarten, dass auf den Epipolarlinien zu einem unbekanntem Pixel des Zwischenbildes mindestens $k-1$ Objektpixel zu finden sind, wenn das unbekanntem Pixel tatsächlich zu einem durch den Roboter verdeckten Hindernis gehört. Daher entspricht das Ergebnis den Erwartungen.

Der Vergleich der verschiedenen Verfahren zeigt, dass bezüglich der fehlenden Konfigurationen das Verfahren mit Hindernissilhouette am besten ist.

Die Anzahl der zusätzlichen Konfigurationen ist beim Verfahren mit Hindernissilhouette schlechter als bei den anderen Verfahren. Dieses Ergebnis überrascht zuerst, kann aber durch die Eigenschaften der Rekonstruktion erklärt werden. Die Objektvergrößerungen durch die Rekonstruktion führen zu einer Anzahl von zusätzlichen, belegten Konfigurationen. Die Verfahren mit Szenensilhouette und rekonstruierter Hindernissilhouette rekonstruieren das Objekt unvollständig. Dadurch steigt nicht nur die Anzahl der fehlenden Konfigurationen, sondern es fällt auch ein Teil der zusätzlichen Konfigurationen weg.

5.4.4 Unbewegliches Hindernis

Ziel dieses Versuches ist die Untersuchung der Eigenschaften des Kollisionstests im realisierten System. Das Versuchssystem ist in Kapitel 7.1 beschrieben. Ein schwarzer Block steht an bekannter Position im Raum, das Silhouettenbild wird durch eine Projektion in die Kameras erzeugt, damit mögliche Fehler der Bilderverarbeitung das Ergebnis nicht beeinflussen. Mit Hilfe des Kollisionstests für statische Hindernisse (siehe Kapitel 5.6) werden die tatsächlich verbotenen Konfigurationen ermittelt. Aus dem tatsächlichen Freiraum werden zufällig Messkonfigurationen ausgewählt und angefahren. Wenn der Roboter an diesen Messkonfigurationen angekommen ist, wird der gesamte Konfigurationsraum rekonstruiert.

Für jede Testkonfiguration werden die Abstände zwischen Testkonfiguration und Messkonfiguration, zwischen Testkonfiguration und nächstgelegener, tatsächlicher Hinderniskonfiguration und zwischen Messkonfiguration und nächstgelegener tatsächlicher Hinderniskonfiguration bestimmt. Bei der Abstandsberechnung werden auch Diagonalschritte mit Abstand 1 bewertet.

Die Anzahl der Fehler 1. Art (fehlende, belegte Konfigurationen) und Fehler 2. Art (zusätzliche belegte Konfigurationen) in Abhängigkeit der drei Abstände wird bestimmt. Da die absolute Anzahl der Konfigurationen in einer bestimmten Entfernung nicht konstant ist, werden alle Fehler relativ zur Anzahl der Konfigurationen in diesem Abstand bestimmt. Die Fehler geben also an, welcher Anteil der Konfigurationen in einem bestimmten Abstand falsch klassifiziert wird.

Als Kollisionstest wird der mit der Szenensilhouette für unkalibrierte Kameras verwendet. Der Schwellwert q_{cam} ist 2. Dieser Versuch wird für 8 verschiedene Hindernis- und Kamerapositionen durchgeführt. Die Ergebnisse sind in den Abbildung 54 bis Abbildung 56 dargestellt und werden im folgenden besprochen.

Der Verlauf der Klassifikationsfehler in Abhängigkeit der Entfernung der aktuellen Roboterkonfiguration zur nächstgelegenen tatsächlichen Hinderniskonfiguration ist in Abbildung 54 zu sehen. Es zeigt sich, dass für viele Szenen der Anteil der fehlenden Konfigurationen in der Nähe von tatsächlichen Hindernissen größer ist als weiter entfernt. Dies kann dadurch erklärt werden, dass das Robotermodell den Roboter konservativ approximiert. Bei der Konstruktion des Zwischenbildes gehen dadurch Hindernispixel, die eigentlich neben dem Roboter liegen, verloren. Der Anteil der zusätzlichen Konfigurationen steigt aus dem gleichen Grund mit zunehmendem Abstand zum Hindernis leicht an. Im rechten Teilbild von Abbildung 54 ist ein untypischer Verlauf dargestellt. In der Szene ist das Objekt so positioniert, dass häufig keine Kamera die Lücke zwischen Roboter und Hindernis erkennen kann. Daher ist der Anteil der zusätzlichen Konfigurationen sehr groß.

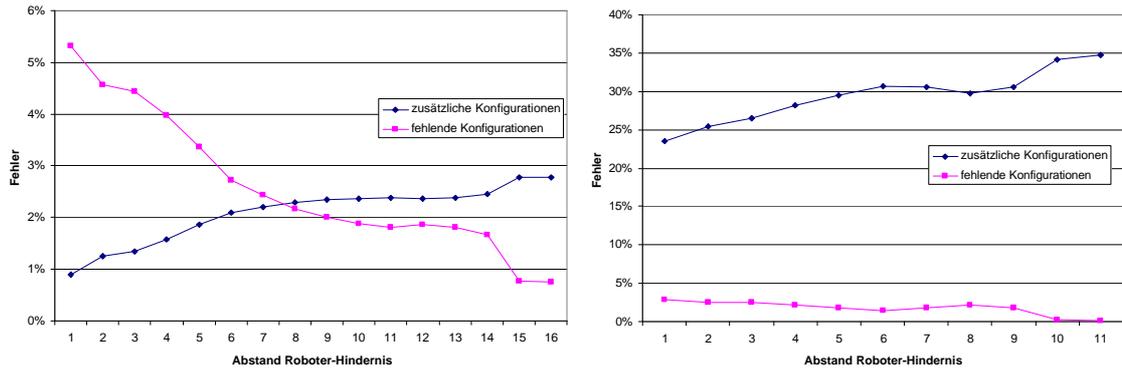


Abbildung 54: Klassifikationsfehler in Abhängigkeit der Entfernung der aktuellen Roboterkonfiguration zur nächstgelegenen tatsächlichen Hinderniskonfiguration. (Links: Typischer Verlauf, rechts: ungewöhnlicher Verlauf)

Der Verlauf der Klassifikationsfehler in Abhängigkeit der Entfernung der aktuellen Roboterkonfiguration zur getesteten Konfiguration ist in Abbildung 55 zu sehen. Es zeigt sich, dass für viele Szenen der Anteil der zusätzlichen Konfigurationen in einem mittleren Abstand maximal wird und mit größerer Entfernung stark abnimmt. Ein ähnlicher Verlauf bei deutlich geringeren Größen ist beim Anteil der fehlenden Konfigurationen zu sehen. Ein ungewöhnliches Ansteigen des Anteils der fehlenden Konfigurationen ist im rechten Teil von Abbildung 55 zu sehen. Die Ursache für diesen Verlauf ist unbekannt.

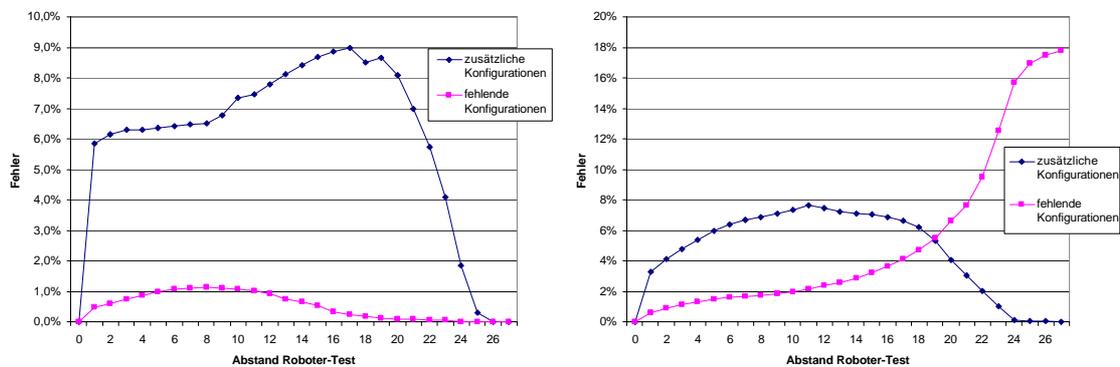


Abbildung 55: Klassifikationsfehler in Abhängigkeit der Entfernung der aktuellen Roboterkonfiguration zur getesteten Konfiguration. (Links: Typischer Verlauf, rechts: ungewöhnlicher Verlauf)

Der Verlauf des Klassifikationsfehlers für zusätzliche Konfigurationen in Abhängigkeit der Entfernung der getesteten Konfiguration zur nächstgelegenen tatsächlichen Hinderniskonfiguration ist in Abbildung 56 zu sehen. Der Anteil der fehlenden Konfigurationen gibt keine sinnvolle Darstellung, da per Definition der Abstand von allen fehlenden Konfigurationen zur nächstgelegenen Hinderniskonfiguration 0 ist. Es zeigt sich, dass für die meisten Szenen die zusätzlichen Konfigurationen mit zunehmendem Abstand stark fallen. Dies kommt daher, dass durch die Rekonstruktionsfehler Hindernis und Roboter etwas vergrößert werden. In der Nähe von Hindernissen werden daher Kollisionen erkannt, obwohl keine auftreten. Der Verlauf, der im rechten Teilbild von Abbildung 56 zu sehen ist, ist ungewöhnlich, da ein sehr großer Anteil der Konfigurationen bei relativ großer Entfernung zum Hindernis fälschlich als belegt klassifiziert wird. Dies kommt daher, dass in der Szene das Hindernis so positioniert war, dass der Rekonstruktionsfehler sehr groß geworden ist.

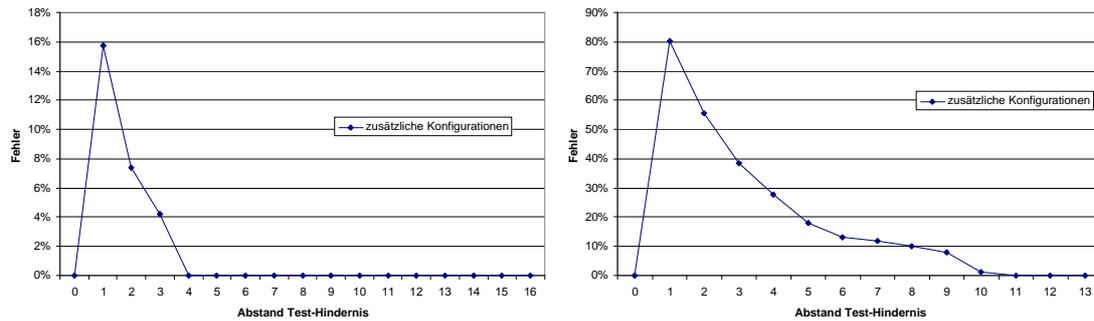


Abbildung 56: Klassifikationsfehler in Abhängigkeit der Entfernung der getesteten Konfiguration zur nächstgelegenen tatsächlichen Hinderniskonfiguration. (Links: Typischer Verlauf, rechts: ungewöhnlicher Verlauf)

5.5 Gleichzeitiger Kollisionstest für mehrere Konfigurationen

Bei den bisher vorgestellten Verfahren wird jeweils eine einzelne Konfiguration auf Kollision mit Hindernissen überprüft. Wenn mehrere Konfigurationen gleichzeitig getestet werden könnten, ließe sich die notwendige Anzahl der Kollisionstests verringern. Dadurch wäre eine Beschleunigung der auf dem Kollisionstest aufbauenden Kollisionsvermeidung möglich.

5.5.1 Gleichzeitiges Testen von Punkt-zu-Punkt Bewegungen

Der größte Zeitverbrauch bei der Bahnplanung liegt im Kollisionstest. Die Anzahl zu testender Konfigurationen bestimmt die Zeit. Bei Bahnplanern, die die gesamte Nachbarschaft einer Zelle untersuchen, wie zum Beispiel Best-First oder HD-I, wächst die Anzahl der zu testenden Konfigurationen bei gleichbleibender Diskretisierung exponentiell mit der Anzahl der Dimensionen des Konfigurationsraums.

Dadurch, dass die Roadmap-Bahnplaner auf die Untersuchung der gesamten Nachbarschaft verzichten, erreichen sie die Aufhebung des exponentiellen Wachstums der Anzahl zu testender Konfigurationen mit der Anzahl der Dimensionen des Planungsraumes. Bei der Suche nach der direkten Verbindung zwischen zwei Konfigurationen spielt allerdings die vorgegebene Diskretisierung eine Rolle. Unter der Annahme, dass jeder Konfigurationstest unabhängig von vorherigen Tests durchgeführt wird, wächst der Aufwand umgekehrt proportional zur Diskretisierung. Durch das gleichzeitige Testen einer Teilbahn könnte dies umgangen werden.

Ansatz

Der untersuchte Ansatz beruht auf einer rekursiven Aufspaltung der Kollisionserkennung.

Anstelle der Silhouette einer einzelnen Roboterkonfiguration wird die Swept-Volume-Silhouette der ganzen Punkt-zu-Punkt-Bewegung verwendet. Die Erzeugung der Swept-Volume-Silhouette wurde in Kapitel 4.2.4 beschrieben.

Diese Swept-Volume-Silhouette wird anstelle der üblichen Testsilhouette dem Kollisionstest übergeben. Wenn der Kollisionstest keinen Schnitt mit der Hindernissilhouette findet, ist die gesamte Punkt-zu-Punkt-Bahn frei.

Wird allerdings eine Kollision erkannt, so ist es möglich, dass die Bahn trotzdem kollisionsfrei ist. Zum Beispiel könnten in einer Silhouette die zu einer Konfiguration c_1 gehörende Robotersilhouette ein Hindernispixel in Kamera 1 schneiden, während Kamera 2 keinen Schnitt feststellt. Für die Konfiguration c_2 sei die Situation umgekehrt, Kamera 1 findet keine Schnittpixel, Kamera 2 erkennt einen Schnitt. Wenn dies die einzigen Kameras sind und der Grundalgorithmus verwendet wird, werden die Konfigurationen c_1 und c_2 dem Freiraum C_F zugeordnet. Wenn jedoch beide Konfigurationen in einer Punkt-zu-Punkt-Bahn vorkommen, führt der Schnitt mit der Swept-Volume-Silhouette zu einem Schnitt in Kamera 1 und 2 und somit zu einer (vorläufigen) Zuordnung beider Konfigurationen zum Objektraum C_O .

Aus diesem Grund wird eine Punkt-zu-Punkt -Bahn, die als belegt klassifiziert wurde, in der Mitte in zwei Teilbahnen unterteilt. Beide Teilbahnen werden wiederum getestet.

Andererseits ist die Swept-Volume-Silhouette bei zu komplexen Bewegungen eventuell nicht genau genug. In der realisierten Methode der Bounding-Box kann dies auf Grund der Approximation der Roboterbewegung durch Polynome 4. Grades der Fall sein. Eine Kriterienfunktion entscheidet, ob die Silhouette für die aktuelle Bahn genau genug ist. Im realisierten System wird als Kriterium der Abstand von Start- und Zielkonfiguration verwendet. Ist die Silhouette nicht genau genug, wird die Bahn in der Mitte unterteilt und die beiden Teilbahnen einzeln getestet.

Die Rekursion bricht ab, wenn eine minimale Bahnlänge erreicht wird. Dann wird ein Kollisionstest für eine einzelne Konfiguration verwendet, um Start- und Zielkonfiguration der Teilbahn auf Kollision zu untersuchen.

Eigenschaften

Angenommen, jeder Kollisionstest dauert gleich lang, egal ob nur eine Konfiguration oder eine ganze Teilbahn getestet wird.¹⁵ Dann beträgt der Aufwand der rekursiven Aufspaltung im Worst-Case den doppelten Aufwand im Vergleich zur linearen Aufspaltung. Der Worst-Case ist eine Bahn, bei der alle zu testenden Teilbahnen als belegt erkannt werden.

Da beim Randomized-Roadmap-Bahnplaner ein bedeutender Anteil der Zeit darauf verwendet wird, Teilbahnen zu erzeugen, die weit von den Hindernissen entfernt sind, wird erwartet, dass der Average-Case deutlich beschleunigt werden kann.

Algorithmus

In Algorithmus 2 ist das Verfahren zum rekursiven Test einer Punkt-zu-Punkt-Bahn dargestellt. Als Kriterium, ob die Swept-Volume-Silhouette genau genug ist, wird der Abstand zwischen Start- und Zielkonfiguration genommen. Ist der Abstand zu groß, wird die Bahn unterteilt.

¹⁵ Da ein anderes Robotermodell verwendet wird, ist diese Annahme nicht zwingend erfüllt. Ebenso ist es möglich, dass auf Grund der größeren Anzahl von Vordergrundpixel bei Swept-Volume-Silhouette der Kollisionstest früher eine Kollision erkennt und daher der Test kürzer dauert als bei einer einzelnen Konfiguration.

```

bool collisionTestPTP (configuration cmin, cmax, unsigned float
mindist, maxdist)

IF distance(cmax,cmin) < mindist
  Silhouette smin = GetSilhouetteFromConfiguration(cmin)
  Silhouette smax = GetSilhouetteFromConfiguration(cmax)
  RETURN ( TestSilhouette(smin) AND TestSilhouette(smax) )
ELSE
  Silhouette svol = GetSweptVolumeSilhouette(cmin,cmax)
  IF ( distance(cmax,cmin) > maxdist ) OR
    NOT(TestSilhouette(svol)) )
    cmid = (cmin + cmax) / 2
    RETURN ( collisionTestPTP(cmin, cmid, mindist, maxdist)
AND
    collisionTestPTP(cmid, cmax, mindist, maxdist)
  ELSE
    RETURN TRUE
  END IF
END IF
END collisionTestPTP

```

Algorithmus 2: Rekursiver Kollisionstest für eine Punkt-zu-Punkt-Bahn.

Experimente

Der Kollisionstest für PTP-Bahnen wird in drei verschiedenen Hindernisszenen untersucht. Jede Szene wird in einem großen und in einem kleinen Konfigurationsraum realisiert.

Die Testaufgaben bestehen aus einer Fahrt im leeren Raum, sowie einer Fahrt um ein statisches Hindernis und der Fahrt um einen Menschen im Arbeitsraum. Die beiden letzten Experimenten sind in Abbildung 57 zu sehen.

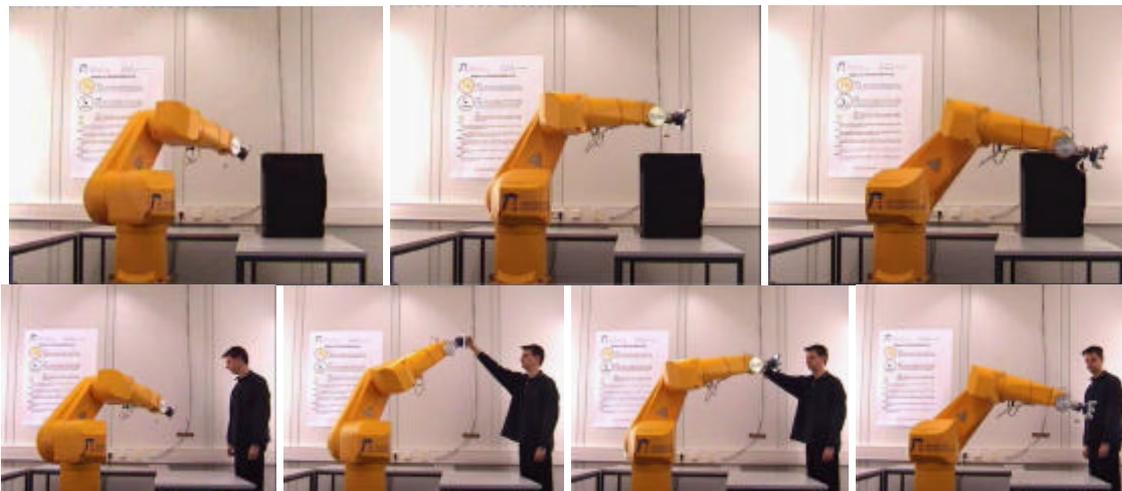


Abbildung 57: Experiment zum Kollisionstest für PTP-Bahnen. (Obere Reihe: Statisches Hindernis, Untere Reihe: Mensch) ([Urbanczik03])

Die relative Laufzeit des Kollisionstests für Punkt-zu-Punkt-Bahnen im Vergleich zum Testen der einzelnen Konfigurationen der Bahn ist in Abbildung 58 dargestellt. Zu Er-

kennen ist, dass der PTP-Test im kleinen Konfigurationsraum mehr Zeit braucht, wenn dort Hindernisse vorhanden sind. Dies kommt daher, dass dann die Rekursion so tief ausgeführt werden muss, dass das Testen der einzelnen Konfigurationen schneller ist.

Wenn das Hindernis allerdings nur einen kleinen Teil eines relativ großen Konfigurationsraums einnimmt oder der Konfigurationsraum leer ist, dann ist der PTP-Test deutlich schneller als das Testen der Einzelkonfigurationen.

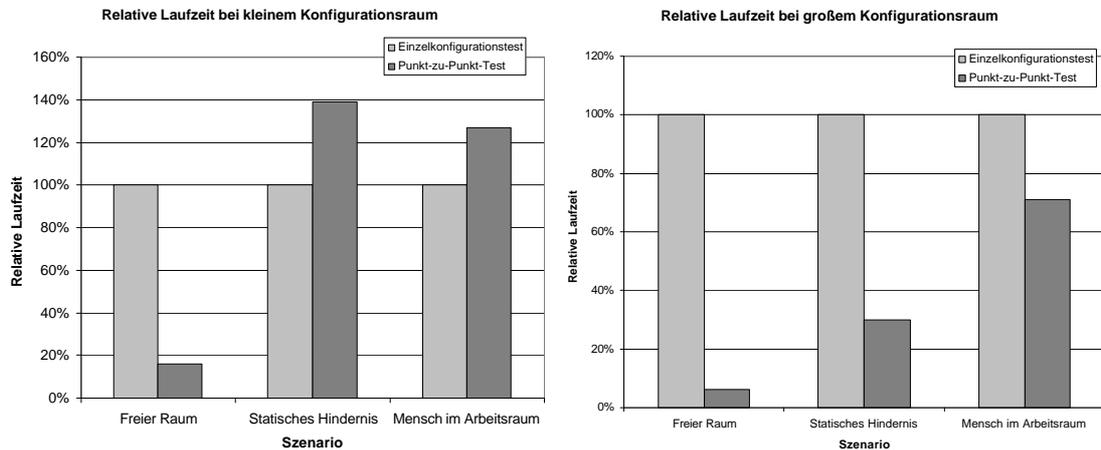


Abbildung 58: Relative Laufzeit des Tests für Punkt-zu-Punkt-Bahnen im Vergleich zum Test der Einzelkonfigurationen für einen kleinen Konfigurationsraum (links) und einen großen Konfigurationsraum (rechts). ([Urbanczik03])

Ergebnis

Das Problem bei der Verwendung des PTP-Tests sind Konfigurationsräume mit relativ großen Hindernissen. Auch die Verwendung einer konvexen Hülle, anstelle einer Bounding-Box, bringt nach [Urbanczik03] keine Verbesserung.

In großen Konfigurationsräumen mit kleinen Hindernissen ist der PTP-Test sehr leistungsfähig.

Da im Versuchssystem die Hindernisse im Vergleich zur Konfigurationsraumausdehnung sehr groß sind, bringt dort der PTP-Test keine Vorteile. Dies liegt jedoch vor allem an der relativ kleinen Konfigurationsraumgröße. Bei realen Anwendungen scheint die Verwendung des PTP-Tests sinnvoll zu sein.

Eine Verbesserung des Algorithmus scheint durch die genauere Darstellung der Bounding-Box erreichbar. Während in der aktuellen Version die Extremwerte für alle Gelenke bestimmt werden, wird die Bounding-Box nur aus deren Maximum und Minimum bestimmt. Durch eine Zerlegung der Bounding-Box in mehrere einzelne Boxen für jedes Glied scheint eine genauere Approximation möglich, ohne dass der Berechnungsaufwand steigt. Dieser Ansatz wurde aus Zeitgründen bis jetzt noch nicht realisiert.

5.5.2 Tabellenbasierte Rekonstruktion des gesamten Konfigurationsraumes

Eine weitere Verbesserung der Leistungsfähigkeit der Kollisionserkennung kann erzielt werden, wenn ein möglichst großer Teil des Rechenaufwandes vorgezogen wird. Dann

wird zur Laufzeit auf die vorher berechneten Daten zurückgegriffen und die eigentliche Kollisionserkennung arbeitet schneller.

Die bisher vorgestellten Ansätze beruhen auf einer Berechnung aller notwendigen Schritte zur Laufzeit. Der im Folgenden vorgestellte und in [Ebert01] veröffentlichte Ansatz speichert die wesentlichen Informationen in einer großen Tabelle ab, auf die zur Kollisionserkennung zurückgegriffen wird.

Ansatz

Beim untersuchten Problem der Rekonstruktion des gesamten Konfigurationsraumes besteht die Eingabe aus den Silhouetten. Die Ausgabe besteht in einem Feld, das für jede Konfiguration angibt, ob sie kollisionsfrei ist oder nicht.

Durch die Diskretisierung des Konfigurationsraumes in einzelne Konfigurationen, die Diskretisierung der Silhouetten in einzelne, binäre Pixel und der endlichen Anzahl an Kameras, ist das Problem der Rekonstruktion des gesamten Konfigurationsraumes endlich. Damit kann im Prinzip jede Funktion, welche die Eingaben auf die Ausgaben abbildet, durch einen großen Speicher realisiert werden.

Gegeben seien wiederum l Kameras mit jeweils p Pixel. Der Konfigurationsraum C bestehe aus c einzelnen Konfigurationen.

Unter diesen Annahmen können insgesamt 2^p verschiedene Szenen auftreten. Im Worst-Case muss für jede Szene der gesamte Konfigurationsraum gespeichert werden. Der Speicheraufwand ist daher $O(c \cdot 2^p)$, allerdings erfolgt die Rekonstruktion in konstanter Zeit.

Der Speicheraufwand sprengt bei praktisch relevanten Werten für p, l, c jede realisierbare Größe. Daher muss der Speicheraufwand auf Kosten der Rechenzeit reduziert werden.

Hierzu wird für jede der Kameras eine eigene Tabelle angelegt, in der jedem Pixel die Konfigurationen zugeordnet sind, die möglicherweise eine Kollision verursachen. Eine solche Tabelle besteht aus p Zeilen. In jeder Zeile stehen im Worst-Case c Einträge. Insgesamt gibt es l solcher Tabellen. Der Speicheraufwand beträgt also $O(c \cdot l \cdot p)$.

Die Rekonstruktion des Konfigurationsraumes erfolgt durch eine Vereinigung der Einträge aller Pixel einer Kamera und den Schnitt dieser Vereinigungen von allen Kameras. Die Vereinigung der Einträge einer Kamera hat den Zeitaufwand $O(c \cdot p)$ und ist l mal durchzuführen. Der anschließende Schnitt von l vereinigten Konfigurationsraumeinträgen erfordert einen Aufwand von $O(c \cdot l)$. Damit wird der Gesamtaufwand $O(c \cdot l \cdot p + c \cdot l) = O(c \cdot l \cdot (p + 1)) = O(c \cdot l \cdot p)$.¹⁶

Konstruktion der Tabellen

Die Tabelle zur Rekonstruktion der Konfigurationsraumbelegung kann mit Hilfe des Robotermodells erhalten werden. Für jede Konfiguration werden die Robotersilhouetten für alle Kameras erzeugt. Für jedes gesetzte Pixel der Silhouetten wird die aktuelle Konfiguration in der zugehörigen Tabelle eingetragen.

¹⁶ Durch diese Verwendung der Tabellen kann allerdings nicht mehr jede mögliche Abbildung der Eingangssilhouetten auf die Konfigurationsraumbelegung realisiert werden.

Algorithmische Umsetzung

In der algorithmischen Umsetzung ist es meistens leichter mit Bitfeldern als mit Mengen zu operieren. Auf Grund der besseren Lesbarkeit wird in Algorithmus 3 allerdings die Mengendarstellung verwendet. Bei der Aufwandsbetrachtung ist aber zu berücksichtigen, dass der Schnitt oder die Vereinigung zweier Konfigurationsräume einen linearen Aufwand in der Anzahl der Konfigurationen hat.

```

ConfigurationSpace reconstructConfigSpace(silhouette sceneSilhouette)

ConfigurationSpace resultSpace, camSpace, tempSpace
resultSpace = allConfigurations
FOR camera = 0 TO maxcamera
    camSpace =  $\emptyset$ 
    FOR row = 0 TO maxrow
        FOR column = 0 TO maxcolumn
            IF scenesilhouette[kamera][column][row]
                tempSpace = Table[kamera][column][row]
                camSpace = camSpace  $\cup$  tempSpace
            END IF
        END FOR column
    END FOR row
    resultSpace = resultSpace  $\cap$  camSpace
END FOR camera
RETURN resultSpace

END reconstructConfigSpace

```

Algorithmus 3: Rekonstruktion des gesamten Konfigurationsraumes, wenn der Roboter nicht in der Szene zu sehen ist.

Ebenso wie der Grundalgorithmus kann auch der Algorithmus zur Rekonstruktion des gesamten Konfigurationsraumes so erweitert werden, dass der Roboter in der Silhouette zu sehen sein darf. Hierzu dürfen diejenigen Pixel, die schon allein auf Grund der Anwesenheit des Roboters im Bild gesetzt sind, nicht betrachtet werden. Allerdings muss ebenso wie im Verfahren für unkalibrierte Kameras, der Schwellwert so angepasst werden, dass er der Anzahl der Kameras entspricht, aus deren Sicht ein Voxel verdeckt sein kann.

Dieses erweiterte Verfahren ist in Algorithmus 4 zu sehen.

Eigenschaften

Die tabellenbasierte Rekonstruktion des gesamten Konfigurationsraumes hat den Vorteil, dass der eigentliche Kollisionstest durch einen Tabellenzugriff realisiert wird.

Allerdings ist der Aufwand der Rekonstruktion, der mit jeder neuen Sensoreingabe aufgewendet werden muss, im Worst-Case genauso groß wie wenn der Grundalgorithmus für jede Konfiguration angewendet würde. Ein leichter Vorteil besteht darin, dass das Robotermodell implizit durch die Tabelle der verbotenen Konfigurationen realisiert wird und schneller ist als eine Berechnung der Robotersilhouetten auf Anfrage. Dies könnte aber durch ein tabellenbasiertes Robotermodell ausgeglichen werden.

Der eigentliche Vorteil liegt darin, dass der Aufwand für die Vereinigung der Teilkonfigurationsräume nur für die gesetzten Silhouettenpixel erfolgt. Die Experimente mit dem realisierten System zeigen, dass dies im Mittel etwa 10% aller Pixel sind. Der Average-Case wird daher deutlich beschleunigt.

Ein Nachteil des tabellenbasierten Ansatzes ist die Eigenschaft, dass, unabhängig von der Anzahl der angefragten Konfigurationen, der gesamte Konfigurationsraum rekonstruiert wird. Im Gegensatz dazu werden beim Grundalgorithmus nur der durch die Bahnplanung angefragte Teil des Konfigurationsraumes rekonstruiert.

Der Hauptnachteil des tabellenbasierten Ansatzes ist der hohe Speicherbedarf. Schon im Versuchssystem mit 4 Kameras, deren Silhouetten aus jeweils 4096 Pixeln bestehen und einem relativ grob diskretisierten Konfigurationsraum mit etwa 20.000 Konfigurationen beträgt der Speicheraufwand (bei 1 Byte pro Konfiguration) etwa 312 MBytes.¹⁷

```

ConfigurationSpace reconstructConfigSpaces(silhouette sceneSilhouette,
configuration currentConfiguration)

ConfigurationSpace result, camSpace, tempSpace
ConfigurationSpaceInt intSpace
FOR camera = 0 TO maxcamera
  FOR row = 0 TO maxrow
    FOR column = 0 TO maxcolumn
      IF scenesilhouette[kamera][column][row]
        tempSpace = Table[kamera][column][row]
        IF not(currentConfiguration isIn tempSpace)
          camSpace = camSpace  $\cup$  tempSpace
        END IF
      END IF
    END FOR column
  END FOR row
  intSpace = intSpace + camSpace
END FOR camera
FOR configuration = 0 TO maxconf
  IF intSpace[configuration] > threshold
    resultSpace[configuration] = TRUE
  END IF
END FOR configuration
RETURN resultSpace

END reconstructConfigSpaces

```

Algorithmus 4: Rekonstruktion des gesamten Konfigurationsraumes, wenn der Roboter in der Szene zu sehen ist.

¹⁷ Im Prinzip würde 1 Bit pro Konfiguration ausreichen. Die bitweise Speicherung (Bit-Stuffing) bringt aber beim Zugriff einen deutlich höheren Zeitverbrauch als die byte- oder wortweise Speicherung. Aus dem selben Grund werden auch die Silhouetten im Rechner durch Byte-Arrays realisiert.

5.6 Berücksichtigung statischer Hindernisse

Die in der Szene vorhandenen statischen Hindernisse können nicht durch den normalen Kollisionstest erkannt werden, da sie nicht in der Szene auftauchen. Wenn ihre Position bekannt ist, können sie jedoch mit Hilfe des geometrischen Robotermodells berücksichtigt werden.

Bei jeder getesteten Konfiguration werden alle Punkte des Robotermodells auf Kollision mit den statischen Hindernissen untersucht. Je nach verwendetem Modell handelt es sich dabei um eine Mehrpunkt- oder Ganzarmerkennung.

Im realisierten System werden alle vorhandenen, statischen Hindernisse modelliert. Beim Start des Systems wird der Abstand aller Gelenkpunkte des kinematischen Modells in jeder Roboterkonfiguration zu den Hindernissen berechnet. Ist der Abstand zu klein, wird die Konfiguration als belegt betrachtet.

In einem Feld wird für alle Konfigurationen gespeichert, ob sie durch statische Hindernisse belegt sind. Bevor der silhouettenbasierte Kollisionstest aufgerufen wird, wird geprüft, ob die Konfiguration durch ein statisches Hindernis belegt ist.

Da im realisierten System nur die Abstände zu den Eckpunkten des kinematischen Modells verwendet werden, handelt es sich, bezüglich der statischen Hindernisse, um eine Mehrpunkterkennung *ME*.

5.7 Zusammenfassung

Der im realisierten System verwendete Kollisionstest baut keine Repräsentation des 3D-Arbeitsraumes auf, sondern arbeitet ausschließlich auf den Silhouettenbildern von Szene und Roboter.

Die Algorithmen zur Kollisionserkennung bestehen im wesentlichen aus drei Teilen:

1. Klassifizierung der einzelnen Silhouettenpixel mit Hilfe des Robotermodells in Hintergrund, Hindernis und Roboter
2. Schnitt der Robotersilhouette in der Testkonfiguration mit den Hindernissen für jede Kamera
3. Ableiten des Gesamtergebnisses aus den Ergebnissen für jede Kamera (zum Beispiel durch Schwellwertbildung).

Es gibt verschiedene Verfahren für einzelne Konfigurationen, je nachdem, ob der Roboter in der Szene zu sehen ist oder nicht. Weiterhin gibt es Verfahren, die mehrere Konfigurationen auf einmal testen. Zum einen ein Verfahren, das eine Punkt-zu-Punkt-Bewegung des Roboters rekursiv untersucht. Zum anderen ein Verfahren zur schnellen Rekonstruktion des gesamten Konfigurationsraumes.

Kapitel 6

Systemintegration

Im vorherigen Kapitel wurde das zentrale Element des realisierten Systems zur sicheren Transferbewegung, der bildbasierte, globale Ganzarmkollisionstest vorgestellt. Im folgenden werden die weiteren Komponenten beschrieben, die notwendig sind, damit dieser Kollisionstest zur sicheren Mensch-Roboter-Koexistenz verwendet werden kann.

6.1 Bahnplanung

6.1.1 Verwendete Bahnplaner

Damit die Vorteile der globalen Kollisionserkennung ausgenutzt werden können, ist ein globales Bahnplanungsverfahren notwendig. Bei lokalen Verfahren kann der Roboter in lokalen Minima festsitzen, obwohl es einen freien Weg zum Ziel gibt.

Es werden die Bahnplanungsalgorithmen Best-First und HD-I, ein Randomized-Roadmap-Planer sowie ein Wellenausbreitungsverfahren realisiert. Der Bahnplanungsalgorithmus HD-I und der verwendete Best-First-Algorithmus stammen aus [Noborio99]. Der Best-First realisiert eine Tiefensuche. Als Randomized-Roadmap-Planer wurde das Verfahren aus [Yu99] verwendet. Die Implementierungen sind in [Heintz02] beschrieben.

Der Randomized-Road-Map Planer wurde später durch einen Bahnglätter ergänzt, der die in [Siegert99] beschriebene Methode verwendet. Die Implementierung hierzu ist in [Urbanczik03] beschrieben. Außerdem wurde das in [Yu99] verwendete Verfahren zur Ermittlung des kürzesten Weges in der aufgebauten Roadmap verändert. Der im Original verwendete Dijkstra-Algorithmus wurde durch das allgemeinere A*-Verfahren ersetzt.

6.1.2 Mehrstufige Bahnplanung

Die Bewertung der verschiedenen Verfahren zeigte, dass keiner der betrachteten Planer wirklich optimal geeignet ist (siehe Kapitel 3.3.5). Durch eine mehrstufige Bahnplanung kann die Gesamtleistung des Systems eventuell verbessert werden.

Die Hauptschwierigkeit ist, die Zeit für die Bahnplanung so weit zu senken, dass eine flüssige Roboterbewegung auch bei großen Konfigurationsräumen möglich ist. Dies wird dadurch erreicht, dass der eigentliche Bahnplaner nur verwendet wird, wenn die

aktuelle Bahn nicht kollisionsfrei ist. Hierzu werden die nächsten Konfigurationen der aktuellen Bahn auf Kollision getestet. Neben dem Geschwindigkeitsgewinn hat dies auch den Vorteil, dass sich die Bahn des Roboters nicht mit jeder Umweltveränderung ändert, sondern nur dann, wenn es notwendig ist. Der Nachteil ist, dass der Roboter unter Umständen auch dann noch um Hindernisse herumfährt, wenn diese bereits nicht mehr vorhanden sind.

Da die Umwelt nicht statisch ist, muss eine einmal gefundene Bahn während der Ausführung immer wieder auf Kollisionen überprüft werden. Dies führt zu einer Aufspaltung der Bahnplanung in Navigator und Pilot, ähnlich wie bei mobilen Robotern. Der Pilot überprüft die Bahn ständig auf Kollisionen und fordert, falls notwendig, vom Navigator eine neue Bahn an.

Wenn der eigentliche Bahnplaner keine Bahn zum Ziel findet, wird auf der direkten Verbindung zum Ziel solange gefahren, bis ein Hindernis im Weg ist.

Bei der bildbasierten Kollisionserkennung kann durch Vergrößern der Hindernissilhouettenbilder erreicht werden, dass ein bestimmter Mindestabstand zu den Hindernissen eingehalten wird. Dadurch braucht der Bahnplaner den Mindestabstand nicht mehr zu garantieren.

Bei der Verwendung des Randomized-Roadmap-Planers kann durch die Punkt-zu-Punkt-Kollisionserkennung (siehe Kapitel 5.5.1) und eines Glätters sowohl die Anzahl der erforderlichen Kollisionstests als auch die Güte der Bahn verbessert werden.

Wenn es keinen Weg zwischen Start und Ziel gibt, dauert es sehr lange, bis der Randomized-Roadmap-Bahnplaner terminiert. Durch den Abbruch des Planungsvorganges nach einer bestimmten Zeitschranke kann dieses Problem beseitigt werden.

Experimente

In einem typischen Szenario werden die realisierten Bahnplaner getestet. Damit die Planungsdauer der einzelnen Planer exakt gemessen werden kann, wird der lokale Planer deaktiviert. In jedem Planungsschritt des Systems wird der untersuchte globale Planer verwendet. Damit die Ergebnisse reproduzierbar sind, wird das Hindernis über den Kollisionstest für statische Hindernisse modelliert.

Das Ergebnis des Versuchs ist in Abbildung 59 zu sehen. Das Wellenausbreitungsverfahren und das HD-I Verfahren brauchen deutlich länger als der Best-First und Randomized-Roadmap-Planer. Der Einfluss des Glätters auf die Laufzeit des Randomized-Roadmap-Verfahrens ist sehr gering.

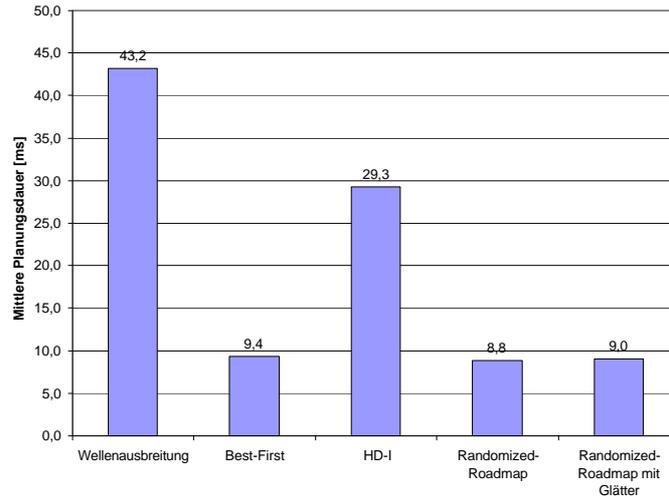


Abbildung 59: Mittlere Planungsdauer der verschiedenen Bahnplaner im realisierten System.

Ergebnis

Die mehrstufige Bahnplanung auf der Basis des Randomized-Roadmap Verfahrens wird nach den gleichen Kriterien bewertet wie die anderen Verfahren in Kapitel 3.3.5. Diese Bewertung ist in Tabelle 12 zu sehen. Zum Vergleich sind die Bewertungen der anderen Verfahren noch einmal aufgeführt. Die Bewertung des Best-First-Verfahrens bei der Anzahl der Kollisionstests wurde allerdings auf Grund der Experimente von „-“ nach „+“ verändert. In den typischen Szenarien führt eine Tiefensuche sehr schnell mit wenigen Kollisionstests zum Ziel.

Insgesamt zeigt sich, dass der optimierte Randomized-Roadmap-Planer mit anschließender Glättung den besten Kompromiss darstellt.

Kriterium	Bahnplaner				
	Wellenausbreitung	Best-First (Tiefensuche)	HD-I	Randomized-Roadmap	Mehrstufige Bahnplanung
Anzahl der Kollisionstests	-	+	-	+	+
Schnelle Terminierung, wenn kein Weg vorhanden	-	o	o	-	o
Erzeugung von Teilbahn, wenn kein Weg vorhanden	-	o	o	o	+
Bahn in Mitte des Freiraumes	-	-	-	o	-
Vorsehbare und intuitive Bahn	+	o	o	-	o
Kurze und glatte Bahn	+	o	o	-	o
Gesamtbewertung	-2	±0	-2	-2	+1

Tabelle 12: Bewertung der betrachteten Bahnplanungsverfahren

6.2 Erzeugung kontinuierlicher Bewegungen

6.2.1 Problem

Bei einer ersten prototypischen Realisierung des Systems zur Sicherung der Transferbewegung zeigte sich in Versuchen eine deutlich ungleichmäßige Bewegung, bei der die Geschwindigkeit des Roboters ständig zwischen der Maximalgeschwindigkeit und kurzen Phasen des Stillstandes pendelte. Eine Analyse ergab, dass die Ursache im sequentiellen Ablauf des Systems lag.

6.2.2 Sequentieller Kontrollfluss

Man kann das System aus Computer mit Bildverarbeitung und Bahnplanung auf der einen Seite sowie Steuerung mit Roboter auf der anderen Seite als ein parallel arbeitendes System betrachten. Die zeitliche Verzahnung der beiden Teilsysteme beim sequentiellen Ablauf ist in Abbildung 60 dargestellt.

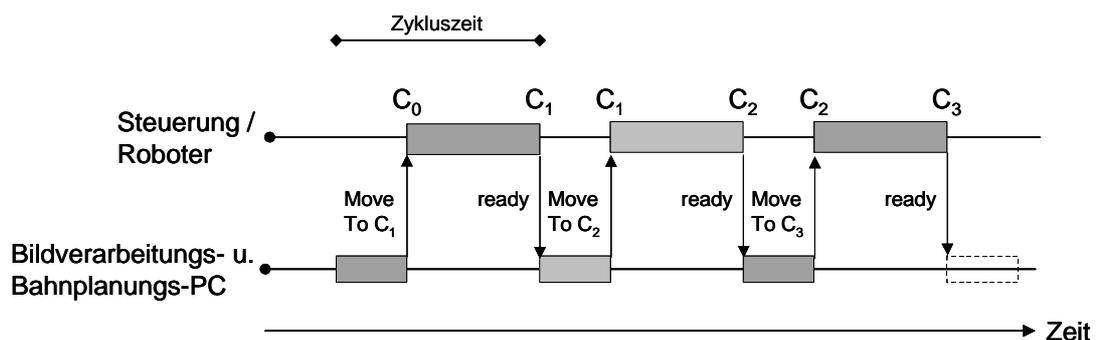


Abbildung 60: Diagramm der zeitlichen Verzahnung von Roboter und PC beim sequentiellen Kontrollfluss. Der PC schickt die nächste Konfiguration an die Steuerung und wartet auf das Signal, dass der Roboter an dieser Konfiguration angekommen ist. Dann verarbeitet der PC die aktuellen Kamerabilder und plant die nächste Konfiguration. (ähnlich [Gecks03])

Es zeigt sich, dass sowohl Roboter als auch PC Stillstandzeiten haben. Die Wartezeiten sind abhängig von der Rechenleistung des Computers und der Verfahrensgeschwindigkeit des Roboters.

6.2.3 Paralleler Kontrollfluss

Durch eine Parallelisierung des Kontrollflusses lassen sich die Wartezeiten des Roboters vermeiden, solange die Bewegung zur nächsten Konfiguration länger dauert als die Bildverarbeitung und Bahnplanung zusammen. Die zeitliche Verzahnung dieses Verfahrens ist in Abbildung 61 dargestellt.

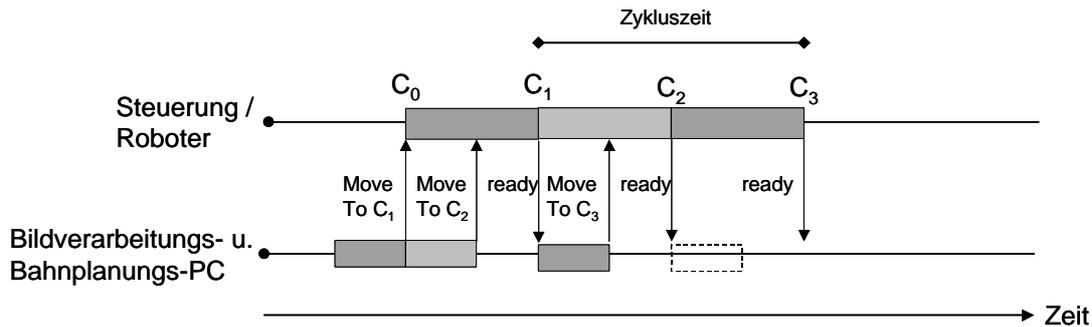


Abbildung 61: Diagramm der zeitlichen Verzahnung von Roboter und PC beim parallelen Kontrollfluss. Der PC schickt bereits die nächste Konfiguration an die Steuerung, während sich der Roboter noch in der Bewegung auf die vorherige Konfiguration befindet. Sobald die Steuerung die Ankunft des Roboters an einer der Konfigurationen meldet, verarbeitet der PC die aktuellen Kamerabilder und plant die übernächste Konfiguration. (ähnlich [Gecks03])

Die Grundidee besteht darin, bereits dann die Folgekonfiguration zu planen, solange sich der Roboter noch auf dem Weg zur vorherigen Konfiguration befindet.

Der Vorteil dieser Methode ist, dass der Roboter nicht abzubremesen braucht, solange die Bewegung länger dauert, als der PC für die Bildverarbeitung und Bahnplanung zusammen braucht. Durch Überschleifen der Bewegung zwischen den Konfigurationen kann eine flüssige Roboterbewegung erzielt werden.

Wenn sich der Roboter zu schnell bewegt, dann kommt der PC nicht mehr mit der Berechnung neuer Konfigurationen nach. Die Folge ist eine stockende Bewegung. Diese ist zwar nicht wünschenswert, aber aus Sicherheitsgründen deutlich besser als eine unüberwachte Bewegung.

Der Nachteil des parallelen Kontrollflusses ist, dass die Zeitspanne zwischen Aufnahme der Kamerabilder und Ankunft an der auf diesen Kamerabildern geplanten Konfiguration größer ist als beim sequentiellen Kontrollfluss. Diese Zeitspanne, die Zykluszeit, ist in Abbildung 60 und Abbildung 61 eingezeichnet.

Weitere Ansätze, um das beschriebene Problem zu lösen, wurden in [Gecks03] untersucht.

6.2.4 Experimente

Ziel des Versuches ist die Ermittlung der maximalen Robotergeschwindigkeit, bei der noch ein kontinuierliches Fahren des Roboters bei der Verwendung des parallelen Kontrollflusses möglich ist. Wie in oben beschrieben, werden die Abläufe auf der Steuerung und dem PC durch Signale synchronisiert. Bewegt sich der Roboter zu schnell, kommt der PC nicht mehr mit der Verarbeitung der Sensordaten und der Erzeugung des nächsten Bahnpunktes nach. Die Folge ist eine stockende Roboterbewegung.

Verschärft tritt das Problem auf, wenn eine neue Bahn durch den globalen Bahnplaner geplant werden muss. Dies kann dadurch geschehen, dass die bisher geplante Bahn nicht mehr frei ist oder dass ein neuer Zielpunkt angefahren wird.

Im ersten Teilversuch wird untersucht, ab welcher Robotergeschwindigkeit die Bewegung stockend wird, ohne dass eine globale Bahnplanung erforderlich ist. Der Roboter

bewegt sich von einer Startkonfiguration c_S am Rande des Konfigurationsraum zu einer möglichst weit entfernten Zielkonfiguration c_Z . Über eine Statusfunktion der Steuerung kann gemessen werden, ob der Roboter gleichförmig fährt oder mit einer Abbremsbewegung beginnt. Über einen Timer wird dieser Status 70-150 mal pro Sekunde ausgelesen¹⁸. In Abbildung 62 ist zu sehen, wie der prozentuale Anteil Bremsbewegung mit der Robotergeschwindigkeit zusammenhängt.

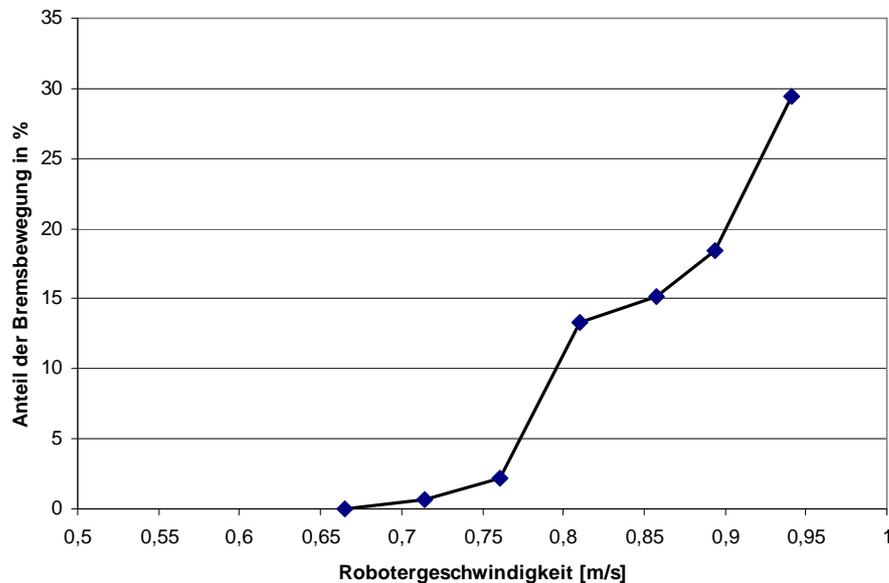


Abbildung 62: Diagramm des Anteils der Bremsbewegung. Dargestellt ist der Mittelwert aus jeweils 10 Messungen.

Bis zu einer Geschwindigkeit des Werkzeugmittelpunktes von 0,7 m/s beträgt der Anteil der Bremsbewegungen unter 1% der Gesamtfahrtzeit. Die Roboterbewegung ist flüssig. Mit steigender Geschwindigkeit wächst der Anteil der Bremsbewegungen. Ab etwa 0,76 m/s werden die Abbremsbewegungen des Roboters für das Auge wahrnehmbar. Bei mehr als 0,9 m/s wird die Bewegung sehr stark stockend.

Im zweiten Teilversuch wird der Einfluss der globalen Bahnplanung untersucht. Der Roboter fährt wieder zwischen den Konfigurationen c_S und c_Z . Allerdings ist die Strecke in 10 Teilstrecken unterteilt. Die Zwischenkonfigurationen liegen auf der Bahn zwischen c_S und c_Z . Eigentlich könnte der Roboter ohne Halt und Änderung der Bewegungsrichtung weiterfahren. Die mittlere Zeit für eine Bewegung von c_S nach c_Z wird gemessen und in die Geschwindigkeit des Werkzeugmittelpunktes umgerechnet. Die mittlere Zeit für eine Bewegung bei einer Teilstrecke und der Bewegung bei 10 Teilstrecken bei gleicher eingestellter Robotergeschwindigkeit werden einander gegenübergestellt. Dies ist in Abbildung 63 zu sehen.

Bis zu einer Geschwindigkeit des Werkzeugmittelpunktes von 0,7 m/s ist die Geschwindigkeit unabhängig von der Streckenunterteilung. Dies kommt daher, dass der PC schnell genug ist, um auch die globale Bahnplanung abzuschließen, bevor der Robo-

¹⁸ Der genaue Anteil hängt von der Robotergeschwindigkeit ab. Bei schnellerer Roboterbewegung braucht der Trajektoriengenerator mehr Rechenzeit der Steuerungs-CPU. Daher steht für das Anwendungsprogramm, das die Statusinformation ausliest, weniger Rechenzeit zur Verfügung.

ter an der vorherigen Konfiguration ankommt. Bei größeren Geschwindigkeiten verursacht die globale Bahnplanung eine niedrigere mittlere Geschwindigkeit.

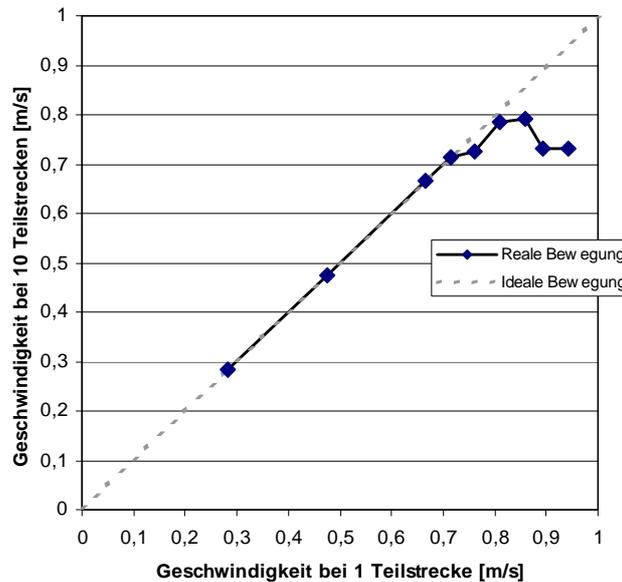


Abbildung 63: Diagramm des Geschwindigkeitsverhältnisses. Aufgetragen ist die mittlere Geschwindigkeit aus 10 Versuchen.

Dadurch, dass der Konfigurationsraum keine Hindernisse enthielt, ist der Zusatzaufwand für die globale Bahnplanung im Experiment klein. Dieser Zusatzaufwand kann bei entsprechender Belegung des Konfigurationsraumes mit Hindernissen sehr groß werden. Die Worst-Case Konfigurationsräume, bei denen fast alle Konfigurationen getestet werden müssen, um einen freien Weg zu finden, kommen im realen System aber nur sehr selten vor.

6.3 Sicherheit

Die Sicherheit des Menschen im Arbeitsraum des Roboters ist nur dann gewährleistet, wenn das System zur Sicherung der Transferbewegung des Roboters korrekt arbeitet. Da ein Fehler des Systems schwere Auswirkungen haben kann, muss sichergestellt sein, dass das System einwandfrei arbeitet.

Für Hard- und Softwarefehler stehen (im Prinzip) Verfahren zur Verfügung, die das Auftreten der Fehler erkennen können. Die klassischen Methoden haben bei komplexen Systemen jedoch Schwierigkeiten, die notwendigen Parameter zu bestimmen. Sobald der Mensch in das System mit einbezogen werden muss, wird die Analyse noch schwieriger, da das Verhalten des Menschen nicht vorhersehbar ist.

6.3.1 Stand der Forschung

Die Sicherheit spielt im Bereich der Industrierobotik eine große Rolle. Allerdings beschäftigen sich die meisten Autoren mit der Ausfallsicherheit eines „klassischen“ Industrieroboters, die hauptsächlich durch die Hardware bestimmt ist. Beispielsweise wird in [Khodabandehloo96] die Fehlermöglichkeit und Einflussanalyse (FMEA) und die Fehlerbaumanalyse exemplarisch für einen Industrieroboter durchgeführt. Das Problem

bei dieser Art von Analyse ist die Schwierigkeit, zu den einzelnen Fehlern die Eintrittswahrscheinlichkeiten zu bestimmen.

In [Gaskill96] wird festgestellt, dass der einzig gangbare Weg zur Gewährleistung der Sicherheit bei interaktiven Maschinen darin besteht, zu gewährleisten, dass die Maschine immer sicher arbeitet und kein zufälliger Hardware- oder systematischer Fehler zu einer gefährlichen Situation führt. Allerdings wird auch festgestellt, dass durch die zunehmende Komplexität der Maschinen die Anzahl der möglichen Fehlerzustände so groß ist, dass sie nicht mehr einzeln erfasst und überprüft werden können.

In [Bertagnolli02] werden die Voraussetzungen untersucht, die erfüllt sein müssen, damit ein autonomer mobiler Roboter in Europa in den Verkehr gebracht werden darf. Hauptaugenmerk ist die Einhaltung des Sicherheitsabstandes, der durch Sensoren überwacht wird. Die Absicherung gegen Fehler des überwachenden Systems wird nicht betrachtet.

6.3.2 Risiko- und Ursachenanalyse

Risikoanalysen dienen zur Erfassung der quantitativen Gefahr, die von einer Einrichtung ausgeht. Für die Zulassung von Geräten, die für den Menschen gefährlich werden können, sind sie meist gesetzlich vorgeschrieben. Die Risikoanalyse nach DIN 19250 definiert hierzu vier Risikoparameter, mit deren Hilfe die Systeme in eine Anforderungsklasse eingeordnet werden können. Die vier Risikoparameter sind Schadensausmaß S , Aufenthaltsdauer A , Gefahrenabwendung G und Eintrittswahrscheinlichkeit W .

Die Anwendung der Risikoanalyse auf ein System zur sicheren Mensch-Roboter-Koexistenz führt durch die folgenden Äste des Entscheidungsbaumes: $S2$ – schwere irreversible Verletzung einer Person, $A2$ – häufiger Aufenthalt im Gefahrenbereich, $G1$ – Gefahrenabwendung möglich unter bestimmten Bedingungen, $W2$ – Eintrittswahrscheinlichkeit gering. Dieser Entscheidungsbaum ist in Abbildung 64 zu sehen.

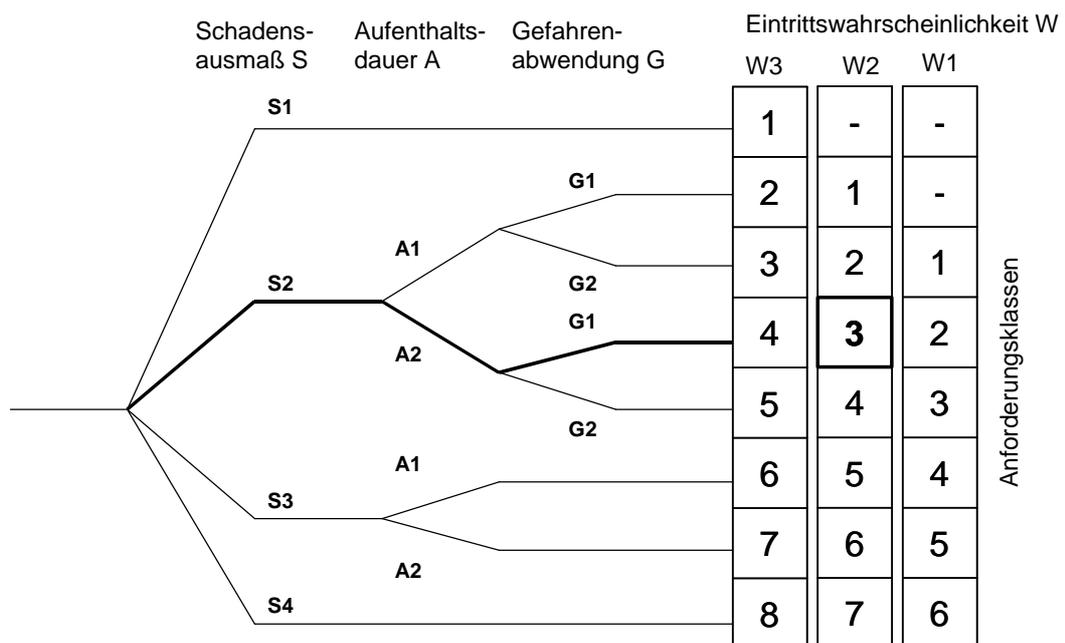


Abbildung 64: Risikograph für ein System zur sicheren Mensch-Roboter-Koexistenz (nach DIN 19250)

Das System zur sicheren Mensch-Roboter-Koexistenz wird folglich in die Anforderungsklassen 3 eingeordnet. Dies stimmt mit dem Ergebnis aus [Heiligensetzer02a] für ein Kleinrobotersystem zur sicheren Mensch-Roboter-Kooperation überein.

6.3.3 Problematik der Sensorfehler

Bei sensorbasierten Systemen kommt zu den klassischen Hard- und Softwarefehlern noch das Problem der Sensorfehler hinzu. Sensorfehler bedeutet, dass die von der Sensorik gelieferten Daten, nicht mit den erwarteten Daten übereinstimmen. Sensorfehler können durch Messfehler, Hardwarefehler in der Sensorik oder durch nicht berücksichtigte Umweltzustände entstehen. Da das System auf der Grundlage der Sensordaten Entscheidungen trifft, können Sensorfehler zu Fehlentscheidungen führen.

Messfehler sind zufällige Schwankungen der gemessenen Sensorwerte um den wahren Wert. Die Schwankungsbreite wird auch als Sensorrauschen bezeichnet. Bei einem kleinen Teil der Messwerte (Ausreißer) kann der Fehler sehr groß werden, ohne dass die genaue Ursache im Einzelfall zu finden ist.

Hardwarefehler können auf vielfältige Art zu Sensorfehlern führen. Je nachdem welches Bauteil betroffen ist, kann die Auswirkung von einer Vergrößerung des Sensorrauschens bis hin zum Totalausfall gehen.

Systeme mit Sensoren stellen Annahmen über die Umwelt auf. Wenn diese Annahmen nicht erfüllt sind, können auch korrekt arbeitende Systeme Fehlentscheidungen treffen.

6.3.4 Erkennung von Sensorfehlern

Sensorfehler können auf verschiedene Weise erkannt werden, je nachdem, welche Ursache sie haben.

Die einfachste Methode ist die wiederholte Messung mit demselben Sensor. Der Nachteil liegt darin, dass jede Messung Zeit kostet.

Redundante Sensoren vermeiden dies, allerdings sind die Kosten deutlich größer. Zudem kann es im Einzelfall schwierig sein, die Sensoren so zu platzieren, dass sie die gleichen Ergebnisse liefern.

Sensorfehler lassen sich auch dadurch erkennen, dass die erhaltenen Sensordaten mit Hilfe eines Modells überprüft werden. Ein Modell kann entweder das Verhalten der erwarteten Sensordaten oder das Verhalten von aus den Sensordaten abgeleiteten Größen beschreiben. Beide Modellarten können entweder Aussagen für jeden einzelnen Wert oder für die statistische Verteilung aller Messwerte treffen. Die einfachsten Modelle geben Rahmengrößen für die einzelnen Messwerte an, wie Mittelwert oder Maximum. Das komplexeste Modell besteht in einer vollständigen Simulation der Umwelt und dem Sensor. Aus der simulierten Umwelt und den Eigenschaften des Sensors werden die erwarteten Sensorwerte berechnet. Zum Beispiel könnten Kamerabilder einer Roboterzelle mit gerenderten Bildern des CAD-Modells der Roboterzelle verglichen werden.

Messfehler können bereits durch mehrfache Messung mit demselben Sensor erkannt werden. Ebenso ist dies mit redundanten Sensoren und durch einen Vergleich der Sensordaten mit einem Modell möglich.

Hardwarefehlern können durch eine wiederholte Messung nicht immer erkannt werden. Eine Erkennung durch redundante Sensoren oder einem Vergleich der Sensordaten mit einem Modell ist jedoch möglich.

Unerwartete Umweltzustände können weder durch wiederholte Messung noch durch redundante Sensoren erkannt werden. Die Erkennung mittels eines Modells ist prinzipiell möglich, führt aber zu weiteren Annahmen über den Umweltzustand, die ebenfalls überprüft werden müssten.

Die Eignung der Erkennungsmethoden bei verschiedenen Fehlerursachen ist in Tabelle 13 zusammengefasst.

		Fehlerursache		
		Messfehler	Hardwarefehler im Sensor	Unerwarteter Umweltzustand
Erkennungsmethode	Wiederholte Messung	X	-	-
	Redundante Sensoren	X	X	-
	Vergleich mit Modell	X	X	X

Tabelle 13: Methoden zur Erkennung von Sensorfehlern und Anwendbarkeit bei verschiedenen Fehlerursachen.

6.3.5 Fehler durch unerwartete Umweltzustände

Die Fehler durch unerwartete Umweltzustände sind die am schwierigsten zu erkennenden Sensorfehler.

Systeme mit Sensoren stellen Annahmen über die Umwelt auf. Wenn diese Annahmen nicht erfüllt sind, können auch korrekt arbeitende Systeme Fehlentscheidungen treffen. Aus diesem Grund müssten alle Annahmen, die ein System über die Umwelt aufstellt, überprüft werden. Zur Überprüfung dieser Annahmen müsste jedoch wiederum Sensorik eingesetzt werden. Auch diese Sensorik stellt Annahmen über die Umwelt auf, welche wiederum überprüft werden müssten. Letztendlich ist es unvermeidlich, unüberprüfte Annahmen über die Umwelt aufzustellen.

Damit ein System zur Mensch-Roboter-Koexistenz als sicher gelten kann, muss die Wahrscheinlichkeit P_{Fehler} minimiert werden, dass Sensorfehler unbemerkt auftreten. Ein auf Grund von nicht berücksichtigten Umweltzuständen verursachter Sensorfehler tritt mit der Wahrscheinlichkeit P_{Umwelt} auf, mit der die Umwelt der Annahme nicht genügt. Allerdings besteht die Wahrscheinlichkeit $P_{\text{Entdeckung}}$, dass der Benutzer den Fehler erkennt und daher kein Schaden eintritt.

$$P_{\text{Fehler}} = P_{\text{Umwelt}} * (1 - P_{\text{Entdeckung}}) \quad (40)$$

Damit ein System zur Mensch-Roboter-Koexistenz sicher ist, dürfen nur unüberprüfte Annahmen über Umweltzustände getroffen werden, die den folgenden Forderungen genügen:

- Die Wahrscheinlichkeit des Nichtzutreffens der Annahme ist klein
- Die Entdeckungswahrscheinlichkeit ist groß

Damit der Benutzer die Verletzung der Annahme über die Umwelt überhaupt erkennen kann, muss er wissen, welche Annahmen das System trifft. Dies macht erforderlich, dass alle Annahmen dokumentiert sind.

6.3.6 Kamerafehler

Die Problematik der Sensorfehler wurde am Beispiel der Kamerafehler des Systems zur sicheren Mensch-Roboter-Koexistenz untersucht. Einige Fehler sind in Abbildung 65 zu sehen.



Abbildung 65: Sensorfehler bei Kamerabildern. Erwartetes Bild, verrauschtes Bild, falsche Beleuchtung, falsche Zoomeinstellung, falsche Kameraposition, teilweise blockierter Sichtbereich (von links nach rechts und von oben nach unten)

Folgende Annahmen wurden, bislang zum großen Teil implizit, über die Sensoren getroffen:

1. Anfragen liefern neue, aktuelle Bilder
2. Bilder mit gutem Signal-Rausch-Verhältnis
3. Bekanntes Beleuchtungsverhältnis
4. Bekannte Kameraeinstellung (Zoom, Focus, Blende,...)
5. Bekannte Kamerapositionierung im Raum
6. Kameras erfassen den gemeinsamen Arbeitsbereich
7. Der Sichtbereich der Kameras ist nicht durch Objekte eingeschränkt

Die Annahmen 1 und 2 lassen sich durch Veränderungen der Sensordaten überprüfen.

Die Annahme 3 kann direkt durch die Sensordaten (z.B. mittlerer Energiegehalt des Grauwertbildes) überprüft werden, wobei aber Annahmen über die Objekte im Bildbe-

reich und über die Kameraeinstellungen, die die Bildhelligkeit betreffen, mit eingehen (Blende, Shutter, Gain).

Die Annahme 4 kann nicht geschlossen überprüft werden. Im Zweifel muss jede Einstellungsmöglichkeit einzeln überprüft werden. Allerdings lassen sich die Teilannahmen, die die Bildhelligkeit (Blende, Shutter, Gain) betreffen, nur zusammen mit Annahmen über die Objekte im Bildbereich und über die Beleuchtung überprüfen. Die Annahme über die Zoomeinstellung kann nur zusammen mit der Kamerapositionierung geprüft werden. Bei einer Überprüfung der Annahme über die Focuseinstellung, muss die Einwirkung der Blendeneinstellung berücksichtigt werden.

Die Annahme 5 lässt sich durch die Suche von markanten Punkten an bestimmten Bildkoordinaten überprüfen, wobei allerdings die Annahmen über die gesehenen Objekte und über die Kameraeinstellungen, die den Sichtbereich betreffen, mit eingehen (Zoom, Linsenverzerrung).

Die Annahme 6 lässt sich rechnerisch über die Kameraeinstellungen und Kamerapositionierung überprüfen, wenn die Grenzen des gemeinsamen Arbeitsbereiches im kartesischen Raum bekannt sind.

Die Annahme 7 erfordert eine Objekterkennung, die die räumliche Position der Objekte ermittelt. Hierzu sind mehrere Kameras und eine aufwändige Bildverarbeitung erforderlich.

Die einfachste Möglichkeit, die Beleuchtung zu überprüfen, liegt in der Verwendung eines eigenen Sensors (Belichtungsmesser). Die Überprüfung der Annahme, dass der Sichtbereich der Kameras nicht verdeckt ist, sprengt den Rahmen dieser Arbeit.

Eine Überprüfung der Kamerapositionierung und die Untersuchung der Kamerabilder auf Veränderungen beseitigt einen Großteil der Unsicherheit bezüglich der Umweltannahmen. Im folgenden werden diese beiden Verfahren näher betrachtet.

6.3.7 Erkennung von Bildrauschen und Standbildern

Die Erkennung von Bildrauschen und Standbildern erfolgt auf der Basis von Annahmen über die Veränderung der Pixelwerte eines Bildes.

Es wird davon ausgegangen, dass auf Grund des Kamerarauschens es immer Veränderungen eines Bildes zu seinem Vorgänger gibt. Allerdings ist die Veränderung auf Grund des Rauschens sehr klein (z.B. ± 4 Einheiten). Außerdem wird davon ausgegangen, dass der überwiegende Teil des Bildes ruhende Objekte zeigt.

Diese Annahmen werden nicht überprüft. Die Annahme über das Kamerarauschen ist sehr wahrscheinlich. Allerdings ist sie vom Bediener nicht direkt überprüfbar. Die Annahme, dass der überwiegende Teil des Bildes ruhende Objekte zeigt, ist bei einem System, das stationäre Kameras zur Überwachung einer Roboterzelle einsetzt, sehr wahrscheinlich erfüllt. Außerdem kann die Annahme leicht vom Bediener durch Betrachten der Kamerabilder überprüft werden. Daher ist diese Annahme unproblematisch.

Unter den obigen Annahmen kann aus der Veränderung der Pixelwerte zwischen zwei Bildern Bildrauschen und Standbilder erkannt werden. Ist die Anzahl der Pixel, deren Wert sich stärker als das Kamerarauschen verändert hat, zu groß, kann von einem veräuschten Bild ausgegangen werden. Ist die Anzahl der Pixel, deren Wert sich nicht verändert hat, zu klein, kann von einem Standbild ausgegangen werden.

Zur Beschleunigung der Berechnung wird nicht jedes Pixel überprüft, sondern nur eine Teilmenge.

Mit einer kleinen, aber positiven Wahrscheinlichkeit können alle Pixel gleich sein, auch wenn das System noch in Ordnung ist. Werden große Objekte im Bild bewegt (z.B. direkt vor der Kamera) wird dies als Fehler erkannt. Diese Probleme beruhen darauf, dass die Annahmen über die Umwelteigenschaften verletzt werden. Allerdings ist in beiden Fällen das Ergebnis, dass das System einen Fehler erkennt, der nicht vorhanden ist (Fehler 2. Art). Dies ist aus Sicherheitsüberlegungen heraus weniger kritisch als ein unerkannt bleibender Fehler, der zu einer Kollision führen könnte.

6.3.8 Falsche Kamerapositionierung

Die Erkennung einer falschen Kamerakonfiguration erfolgt durch die Suche nach einem Schachbrettmuster im Bild an einer bekannten Position. Zur Schachbrettsuche wird der gleiche Algorithmus verwendet wie bei der Kamerakalibrierung in Kapitel 4.1.7.

Dabei wird von folgenden Annahmen ausgegangen: Der Roboter trägt an einer bekannten Stelle ein Schachbrettmuster mit einer vorgegebenen Form, im Sichtbereich der Kameras ist kein weiteres Schachbrettmuster zu sehen und der Roboter befindet sich in einer vorgegebenen Konfiguration.

Wenn im Bild der Kameras an der erwarteten Position ein Schachbrettmuster gefunden wird, kann unter den obigen Annahmen geschlossen werden, dass sich die Kameras mit hoher Wahrscheinlichkeit an der richtigen Position befinden.

Die kritischen Kamerapositionen befinden sich auf der Geraden zwischen Schachbrett und Kamera. Der Zoom muss so eingestellt sein, dass die scheinbare Größe des Schachbretts gleich bleibt. In der Abbildung 66 ist das Problem der nicht erkennbaren, fehlerhaften Kamerapositionierung schematisch dargestellt. In Abbildung 67 sind die Auswirkungen in der Realität zu sehen.

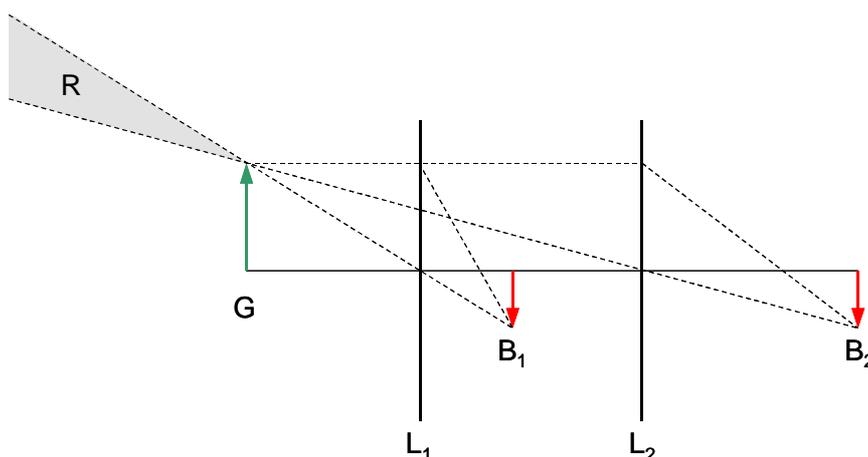


Abbildung 66: Fehlerhafte Kamerapositionierung. Das am Roboter befestigte Schachbrettmuster sei G . Bei der erwarteten Kamerapositionierung wird das Bild B_1 durch die Linse L_1 erzeugt. Das Bild B_2 der fehlerhaften Kamerapositionierung ist nicht vom erwarteten Bild B_1 zu unterscheiden. Jedoch ist nun der schraffierte Raumbereich R nicht mehr im Bild und wird nicht auf Kollisionen überwacht.

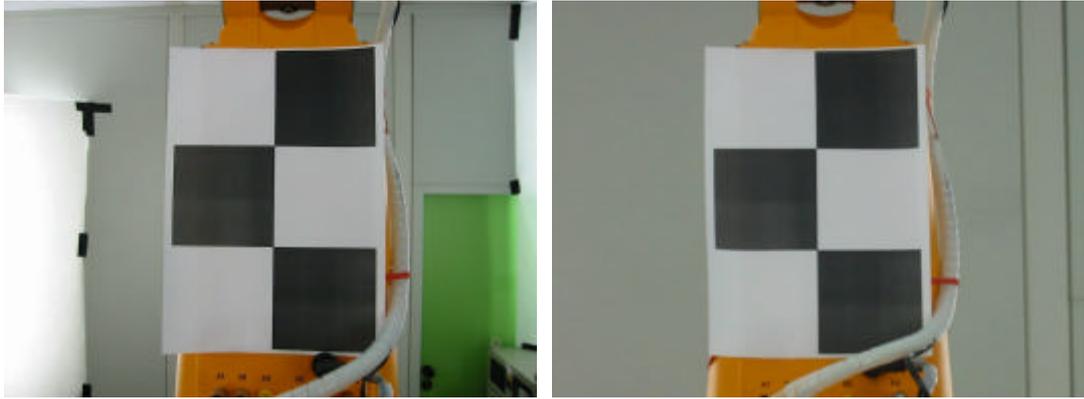


Abbildung 67: Fehlerhafte Kamerapositionierung. Die inneren Punkte der Schachbretter in beiden Fotos liegen (ungefähr) an den gleichen Bildkoordinaten. Wenn das linke Foto der erwarteten Kamerapositionierung entspricht, wird bei der fehlerhaften Kamerapositionierung im rechten Bild der Raumbereich zwischen Roboter und Beleuchtung nicht überwacht.

Bei der gewählten Ausführung des Schachbrettes gibt es einen weiteren Freiheitsgrad auf einer Kreisbahn um den Mittelpunkt des Schachbrettes. Dies ließe sich aber durch ein Schachbrett mit vier inneren Ecken vermeiden.

Das Problem, dass es verschiedene Kamerapositionen gibt, die die gleichen Messwerte erzeugen, tritt dann nicht auf, wenn die Kameras nur einen kleinen Bewegungsspielraum in der Orientierung besitzen. Dies kann zum Beispiel durch eine Befestigung mit Kugelgelenk erreicht werden, wie sie viele Überwachungskameras haben. Eine andere Möglichkeit wäre die Überprüfung von Punkten, deren Abstand zur Kamera unterschiedlich sind. Dies könnte durch ein dreidimensionales Objekt oder mehrere Schachbrettmuster an verschiedenen Stellen realisiert werden.

Das Vorhandensein von mehreren Schachbrettern im Bild kann im Prinzip erkannt werden. Problematisch ist die Situation, bei der ein anderes Schachbrett an der erwarteten Position erkannt wird, während sich das eigentlich gesuchte Schachbrett auf Grund der Positionsänderung außerhalb des Bildes befindet.

Die Wahrscheinlichkeit, dass in der Zelle ein (falsches) Schachbrett so angebracht ist, dass die Koordinaten der inneren Punkte vom System akzeptiert werden, sind deutlich kleiner als die Wahrscheinlichkeit, dass sich die Kameraorientierung oder -position verändert hat. Ein menschlicher Bediener kann deutlich leichter erkennen, ob ein Schachbrettmuster im Sichtbereich der Kameras ist, als die genaue Orientierung der Kameras an Hand der Kamerabilder zu überprüfen.

Aus diesen Gründen erhöht die gewählte Form der Kamerapositionsprüfung die Betriebssicherheit des Systems.

6.3.9 Zusammenfassung

Im realisierten System stellen die betrachteten Fehler im Normalbetrieb keine große Gefahrenquelle dar. Wenn das aktuelle Bild gestört ist, zeigt sich dies normalerweise in einer starken Abweichung zum Referenzbild. Dadurch werden mehr Hindernisse erkannt als tatsächlich da sind. Dies kann dazu führen, dass sich der Roboter nicht mehr bewegt, weil der Raum voller Hindernisse ist. Die Auswirkung eines verwaschenen Bildes auf die Differenzbilderstellung ist in Abbildung 68 zu sehen.

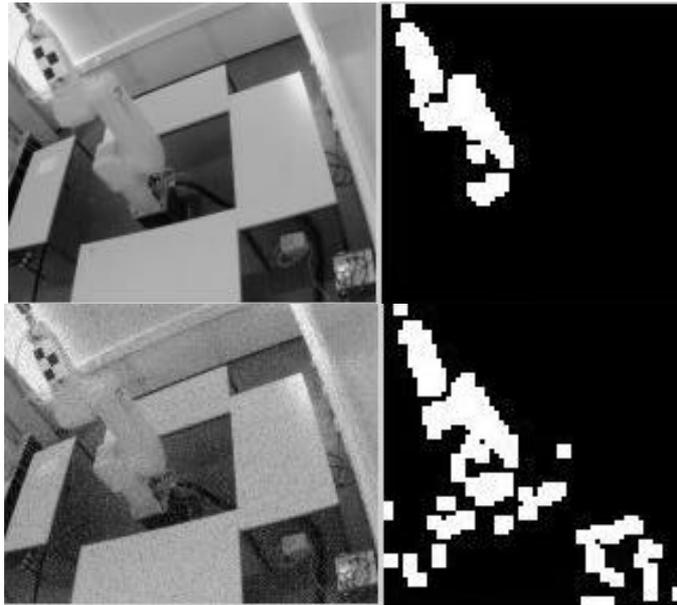


Abbildung 68: Auswirkung von Bildrauschen auf die Differenzbilderstellung. Unverraushtes Kamerabild und zugehöriges Differenzbild (obere Reihe), Verrauschtes Kamerabild und zugehöriges Differenzbild (untere Reihe) ([Wenzel03])

Eine Kollision kann nur dann eintreten, wenn Hindernisse im Raum nicht erkannt werden. Dies kann zum Beispiel dadurch geschehen, dass eine Kamera permanent ein Standbild liefert und das Hindernis nicht im Standbild zu erkennen ist.

Problematischer ist der Einrichtungsbetrieb. Werden die Referenzbilder erstellt, während die Kameras den gemeinsamen Arbeitsbereich von Mensch und Roboter nicht sehen können, wird im anschließenden Normalbetrieb mit hoher Wahrscheinlichkeit jede Konfiguration als frei erkannt, unabhängig von der Position der Hindernisse.

Die Überwachung der Pixelfehler wird durch ein eigenes Modul realisiert, das transparent in den Datenfluss der Bildverarbeitung eingehängt ist. Die Überwachung läuft parallel zum Betrieb des Systems. Die Überprüfung der Kamerapositionierung ist ein eigener Betriebsmodus, der vor der Erstellung der Referenzbilder ausgeführt wird. Die Realisierung der Verfahren zur Überwachung auf Pixelfehler und der Kamerapositionierung ist in [Ebert03] und [Wenzel03] beschrieben.

6.4 Verwendung des Systems

Ziel ist die einfache Integration des Systems zur sicheren Mensch-Roboter-Koexistenz in konventionelle Industrierobotersysteme. Dies wird durch einen skillbasierten Ansatz erreicht, der die Funktionalität der sensorgestützten Bewegung kapselt. Es wird ein weiterer Bewegungsbefehl zur Steuerungssprache hinzugefügt, zum Beispiel MOVE_SAFE genannt. Dieser Befehl zur Ausführung einer gesicherten Transferbewegung wird aus Sicht des Anwenders genauso benutzt wie der Befehl zur normalen, ungesicherten Roboterbewegung.

6.4.1 Ausführungsablauf des Befehls zur sicheren Transferbewegung

Wenn ein Roboterprogramm auf der Steuerung den MOVE_SAFE Befehl aufruft, übergibt die Steuerung alle notwendigen Parameter über eine Netzwerkverbindung an den

Bildverarbeitungs- und Bahnplanungsrechner.¹⁹ Dabei werden alle notwendigen Parameter übergeben. Diese sind Start- bzw aktuelle Konfiguration, Zielkonfiguration und Parameter für die Bahnausführung wie zum Beispiel Verfahrensgeschwindigkeit.

Der Bahnplanungsrechner überprüft alle übergebenen Parameter auf Gültigkeit und prüft, ob alle beteiligten Module betriebsbereit sind. Falls ja, nimmt er die Anfrage an. Von jetzt an wird die Verbindung zwischen Bahnplanungsrechner und Steuerung auf Ausfall und Latenz überwacht und bei Überschreiten von Zeitschranken ein Nothalt ausgelöst.

6.4.2 Umgang mit Konfigurationen außerhalb des Planungsraumes

Start- und Zielkonfiguration der angeforderten Bewegung liegen nicht zwingend auf Gitterpunkten des Planungsraumes und können daher nicht direkt auf Kollision getestet werden. Dieses Problem besteht auch bei den Roadmap-Ansätzen in der Bahnplanung, wie sie zum Beispiel in [Latombe96] vorgestellt sind.

Die Lösung besteht in einer Dreiteilung der abzufahrenden Bahn. Die erste Teilbewegung führt von der Startkonfiguration zu einem Punkt des Planungsraumes. Anschließend wird innerhalb des Planungsraumes bis in die Nähe des Zieles gefahren. Die dritte Teilbewegung verläuft zwischen dem letzten Punkt des Planungsraumes und dem Ziel. Diese Lösung ist in Abbildung 69 dargestellt und wird im folgenden beschrieben.

Die Bahn innerhalb des Planungsraumes verläuft zwischen diskreten Konfigurationen. Jede dieser Konfigurationen kann auf Kollision getestet werden. Durch den Kollisionstest muss sichergestellt sein, dass der Weg zwischen zwei benachbarten Konfigurationen kollisionsfrei ist, wenn beide Konfigurationen kollisionsfrei sind.

Im einfachsten Fall wäre nur die mittlere Teilbewegung sensorisch überwacht. Die Bewegung zum ersten Punkt des Planungsraumes und vom letzten Punkt zum Ziel würde unüberwacht erfolgen.

Liegen Start oder Ziel innerhalb der Grenzen des Planungsraumes, kann durch eine Prüfung der Nachbarkonfigurationen sichergestellt werden, dass auch Start bzw. Ziel kollisionsfrei sind²⁰. Liegt Start oder Ziel außerhalb der Grenzen des Planungsraumes, kann nur die nächstgelegene Konfiguration auf Kollision getestet werden.

¹⁹ Bei ausreichender Rechenleistung auf der Robotersteuerung könnte die Berechnung natürlich auch auf der Steuerung selbst erfolgen. Die aktuellen, kommerziellen Steuerungen sind zur Zeit aber dazu nicht in der Lage.

²⁰ Wenn es Hindernisse gäbe, die kleiner als eine Gittergröße sind und der Kollisionstest den Roboter ohne Sicherheitszuschlag testet, könnte es theoretisch sein, dass zwar die benachbarten Gitterpunkten kollisionsfrei sind, aber dennoch eine Kollision bei einer Konfiguration im inneren der Gitterzelle vorkommt. Im realisierten System besteht dieses Problem aber nicht.

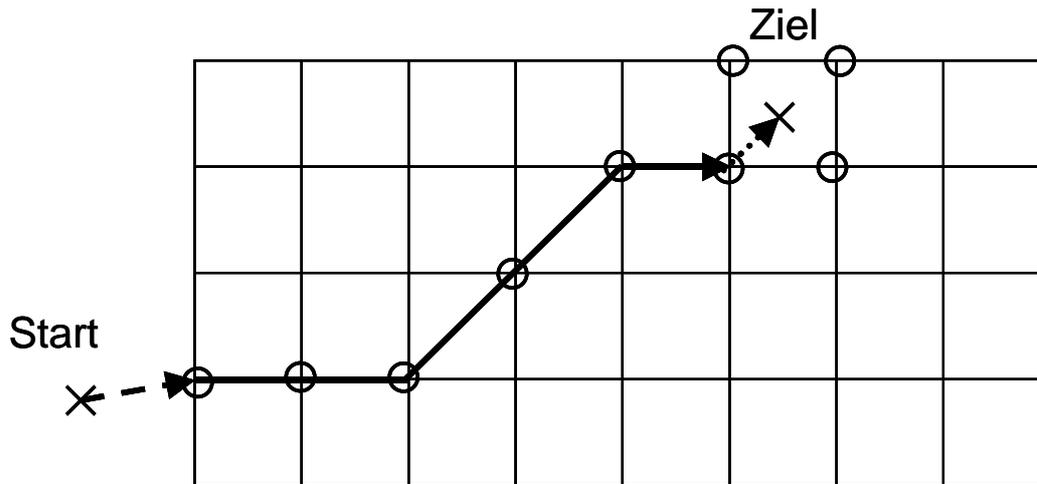


Abbildung 69: Zeichnung des abgefahrenen Pfades. Die Gitterlinien stellen den diskretisierten Konfigurationsraum des Roboters dar. Bei einer Bewegung vom Start zum Ziel werden die eingekreisten Konfigurationen auf Kollision untersucht. Da Start und Ziel nicht notwendigerweise auf Gitterpunkten liegen, spaltet sich eine Bahn in drei Teilstücke auf. Das erste Teilstück führt vom Start zum in Zielrichtung nächstgelegenen Gitterpunkt. Das dritte Teilstück führt vom in Startrichtung nächstgelegenen Gitterpunkt zum Ziel. Dazwischen wird der Roboter von Gitterpunkt zu Gitterpunkt verfahren. Liegen Start oder Ziel innerhalb des Konfigurationsraumes kann durch das Testen aller Nachbarkonfigurationen sichergestellt werden, dass Start bzw. Ziel kollisionsfrei ist. In der Zeichnung ist dies am Beispiel des Zieles dargestellt.

Die obige Funktionalität ist im beschriebenen System bereits integriert worden, wurde aber bislang noch nicht verwendet.

Kapitel 7

Experimente

In den vorherigen Kapiteln wurden bei der Beschreibung der einzelnen Komponenten des Systems zur Sicherung der Transferbewegung auch Experimente zu deren Eigenschaften vorgestellt. Im folgenden werden nun die Eigenschaften des Systems als ganzes untersucht.

7.1 Versuchsaufbau

Der Aufbau der Roboterzelle ist in Abbildung 70 und Abbildung 71 zu sehen. Die verwendete Hardware besteht aus einem Stäubli RX130 6-Achs Industrieroboter, der auf einem Sockel in der Mitte der Roboterzelle montiert ist. Der Roboter wird durch eine Stäubli CS7B Robotersteuerung kontrolliert. Auf der Robotersteuerung läuft ein Programm in der steuerungseigenen Sprache V+, das die Kontrolle des Roboters über ein Netzwerk ermöglicht [Gecks01]. Das eigentliche Softwaresystem läuft auf einem Standard-PC mit Athlon XP 2200+ Prozessor und 512 MB Hauptspeicher.

In diesem PC sind zwei DFG/BW1 Framegrabber (baugleich zu ELTEC/PCEYE4) eingebaut. An diese Framegrabber sind insgesamt vier DMK 73/C Grauwertkameras mit Vario-Focus-Objektiven angeschlossen. Die Vario-Focus-Objektive ermöglichen ein manuelles Einstellen von Sichtwinkel (Zoom) und Focus, wobei im Unterschied zu echten Zoom-Objektiven aber Sichtwinkel und Focus nicht unabhängig voneinander eingestellt werden können. Die genauen technischen Daten der Framegrabber und Kameras sind bei [8] und [11] zu finden.

Die Grauwertkameras sind an folgenden Positionen angebracht: Kamera 0 ist an der Decke oberhalb des überwachten Arbeitsraumes angebracht und blickt im Wesentlichen senkrecht nach unten. Kamera 1 ist in der linken, hinteren, oberen Ecke (aus Sicht eines vor der Zelle stehenden Betrachters) angebracht. Kamera 2 befindet sich in der rechten, vorderen oberen Ecke. Kamera 3 ist an der in der rechten, vorderen, unteren Ecke an der Robotersteuerung befestigt. Die Kameraposition ist in Abbildung 70 skizziert.

Die Beleuchtung erfolgt durch Leuchtstoffröhren, die über elektronische Vorschaltgeräte betrieben werden. Dadurch ist die Frequenz der Leuchtstoffröhren deutlich größer als die Bildaufnahmefrequenz der Kameras und es entstehen keine Störungen. Jeweils eine Lampe mit zwei Leuchtstoffröhren befindet sich in den vier Ecken der Arbeitszelle. Außerdem befinden sich drei Lampen mit je einer Leuchtstoffröhre an der Zellendecke. Vor allen Lampen ist zur Erzielung einer diffuseren Beleuchtung Papier gespannt.

Die Software auf Seite der Steuerung ist in der proprietären Sprache V+ geschrieben, auf dem PC wird C++ unter Windows NT 4.0 verwendet. Steuerung und PC sind durch ein Netzwerk (TCP/IP über Ethernet) verbunden.

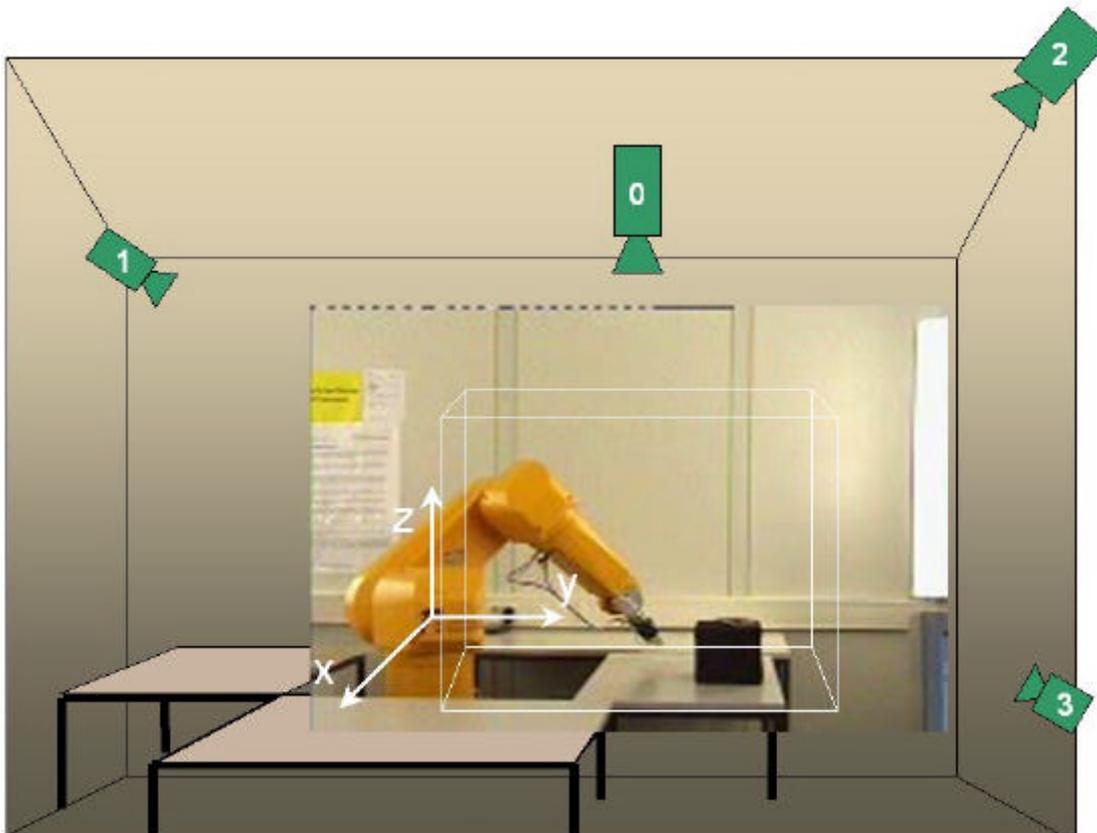


Abbildung 70: Kamerapositionierung im Versuchssystem.

Der Ursprung des Weltkoordinatensystems liegt im Schnittpunkt der 1. und 2. Achse des Roboters. Die positive x-Achse respektive y-, z-Achse, zeigen nach vorne bzw. rechts, oben. Das Koordinatensystem und die ungefähren Abmessungen des gemeinsamen Arbeitsraumes von Mensch und Roboter sind in der Abbildung 70 eingezeichnet.

Die Weltkoordinaten der Kamerapositionen sind für Kamera 0 (-443 mm, 985 mm, 1692 mm), für Kamera 1 (-1620 mm, -2080 mm, 1862 mm), für Kamera 2 (1507 mm, 2146 mm, 1868 mm) und für Kamera 3 (1683 mm, 1651 mm, -26 mm).

Der Konfigurationsraum des Roboters ist mit Hilfe der MaxMove-Methode aus [Henrich98] diskretisiert. Es wurde ein MaxMove von 10 cm verwendet. Der resultierende Winkel zwischen zwei benachbarten Konfigurationen beträgt $4,4^\circ$ im ersten, $4,2^\circ$ im zweiten Gelenk und $7,8^\circ$ im dritten Gelenk. Der Konfigurationsraum besteht aus $28 \times 20 \times 24 = 13\,440$ Konfigurationen. Der resultierende Konfigurationsraum umfasst einen Winkelbereich von $118,8^\circ$, $79,8^\circ$ bzw. $179,4^\circ$ in den Gelenken 1,2 bzw. 3.

Neben den untersuchten dynamischen Kollisionstests wird der statische Kollisionstest eingesetzt. Auf diese Weise wird verhindert, dass ein Gelenk des Roboters tiefer als 75cm ab Boden kommen kann. Dadurch werden Kollisionen mit den Tischen (Höhe 70cm) verhindert.

Ein Foto der Roboterzelle in der verwendeten Konfiguration und die zugehörigen Kamerabilder sind in Abbildung 71 zu sehen.

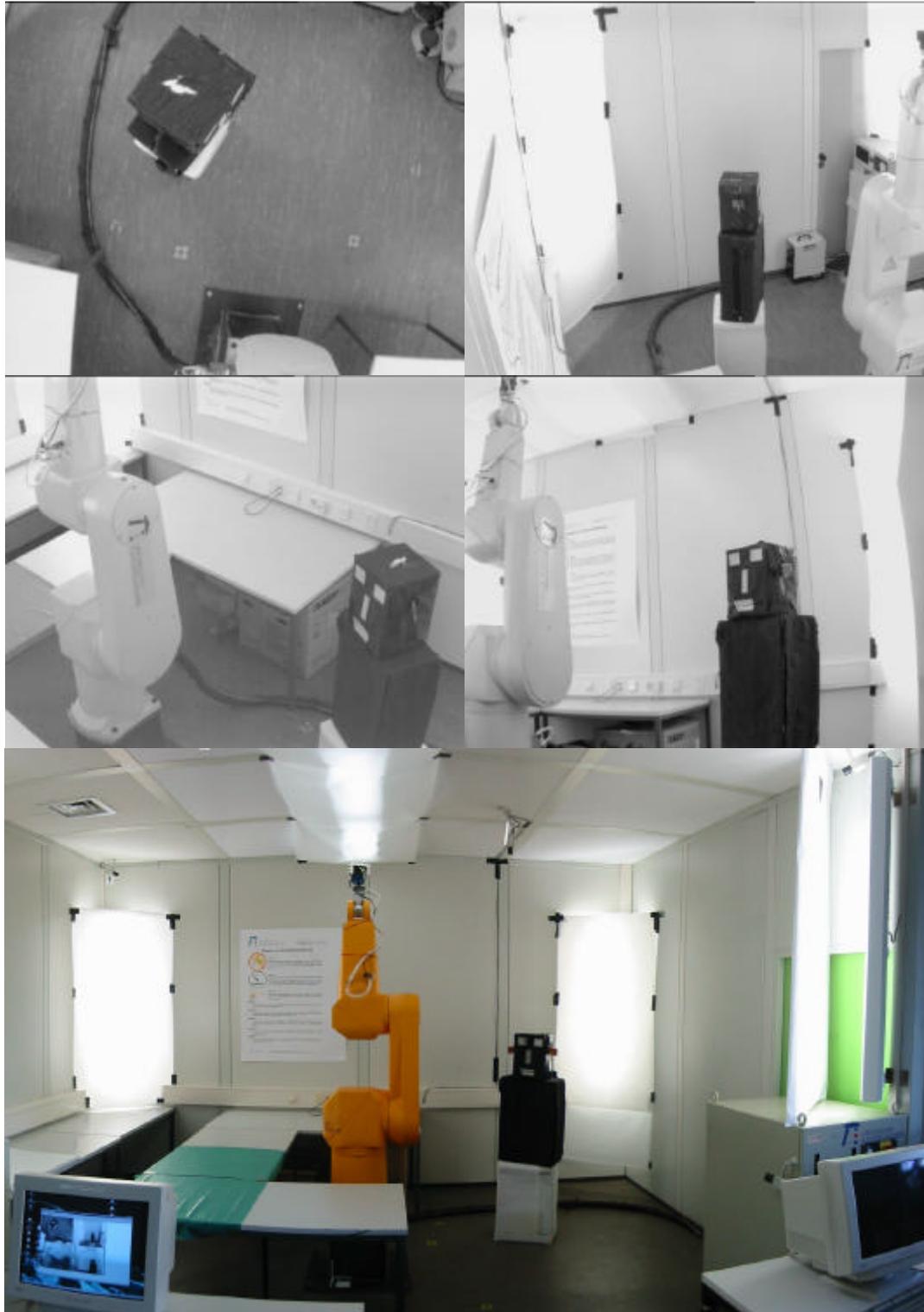


Abbildung 71: Versuchssystem. Weitwinkelaufnahme der ganzen Zelle (unten). Sicht der Kameras 0 – 3 von oben links nach Mitte rechts.

7.2 Versuchsdurchführung

Ziel des Systems zur Sicherung der Transferbewegung ist die Ermöglichung der Koexistenz von Mensch und Roboter in einem gemeinsamen Arbeitsraum. Das Problem bei Versuchen mit Menschen ist aber, dass deren Verhalten und Bewegungen nur in Grenzen wiederholbar sind.

Die betrachtete Aufgabe soll eine mögliche Anwendung bei der Kleinserienproduktion auf abstrakte Weise realisieren. An einem Rand des Arbeitsraumes symbolisiert ein Tisch ein Förderband, auf dem die zu bearbeitenden Werkstücke transportiert werden. In der Mitte des Arbeitsraumes symbolisiert ein anderer Tisch eine Maschine zur Bearbeitung der Werkstücke.

Der Roboter bewegt sich zu einer Aufnahmekonfiguration oberhalb des „Förderbandes“ und fährt dann zur Bearbeitungskonfiguration bei der „Maschine“. Danach fährt er in eine Wartekonfiguration. Von dort fährt er wieder in die Bearbeitungskonfiguration und dann weiter in eine Ablagekonfiguration an einer anderen Stelle des „Förderbandes“. Anschließend beginnt er Zyklus von neuem.

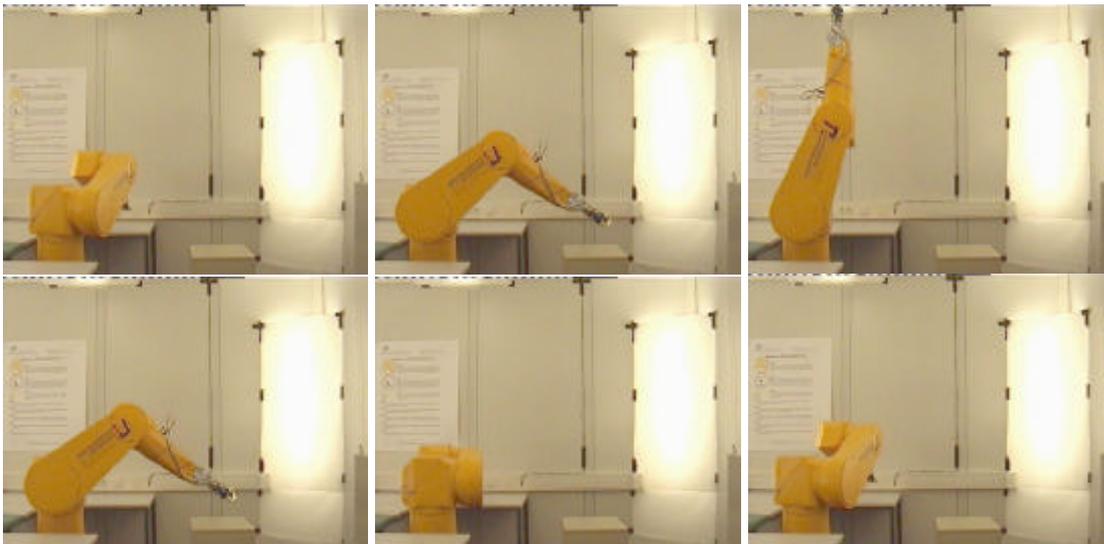


Abbildung 72: *Simulierte Roboter Aufgabe. Der Roboter bewegt sich von der Werkstückaufnahme (oben, links) zur „Maschine“ (oben, Mitte). Anschließend fährt er in eine Warteposition (oben, rechts) und dann wieder zur Maschine (unten, links). Nach der Bewegung zur Ablageposition (unten, Mitte) fährt der Roboter wieder zur Werkstückaufnahme (unten, rechts bzw. oben, links).*

Ausgehend von dieser Situation werden drei Aufgaben definiert:

- Passieren innerhalb der Roboterzelle
- Vorbeugende Wartung der Maschine
- Aufräumen der Zelle

Diese Aufgaben werden im folgenden beschrieben.

7.2.1 Passieren in der Roboterzelle

Die erste Aufgabe simuliert den Betrieb der Maschine in einem vom Menschen genutzten Raum. Der Mensch bewegt sich zwischen den Grenzen des Arbeitraumes hin und her. Es wird aber darauf geachtet, dass die Zielkonfigurationen des Roboters immer frei sind. Dabei wird die Veränderung der Zykluszeit des Roboters im Vergleich zur leeren Zelle gemessen.

Ein Versuchsdurchlauf für die Aufgabe „Passieren“ ist in Abbildung 73 zu sehen. Die einzelnen Bilder haben einen zeitlichen Abstand von 2 Sekunden.

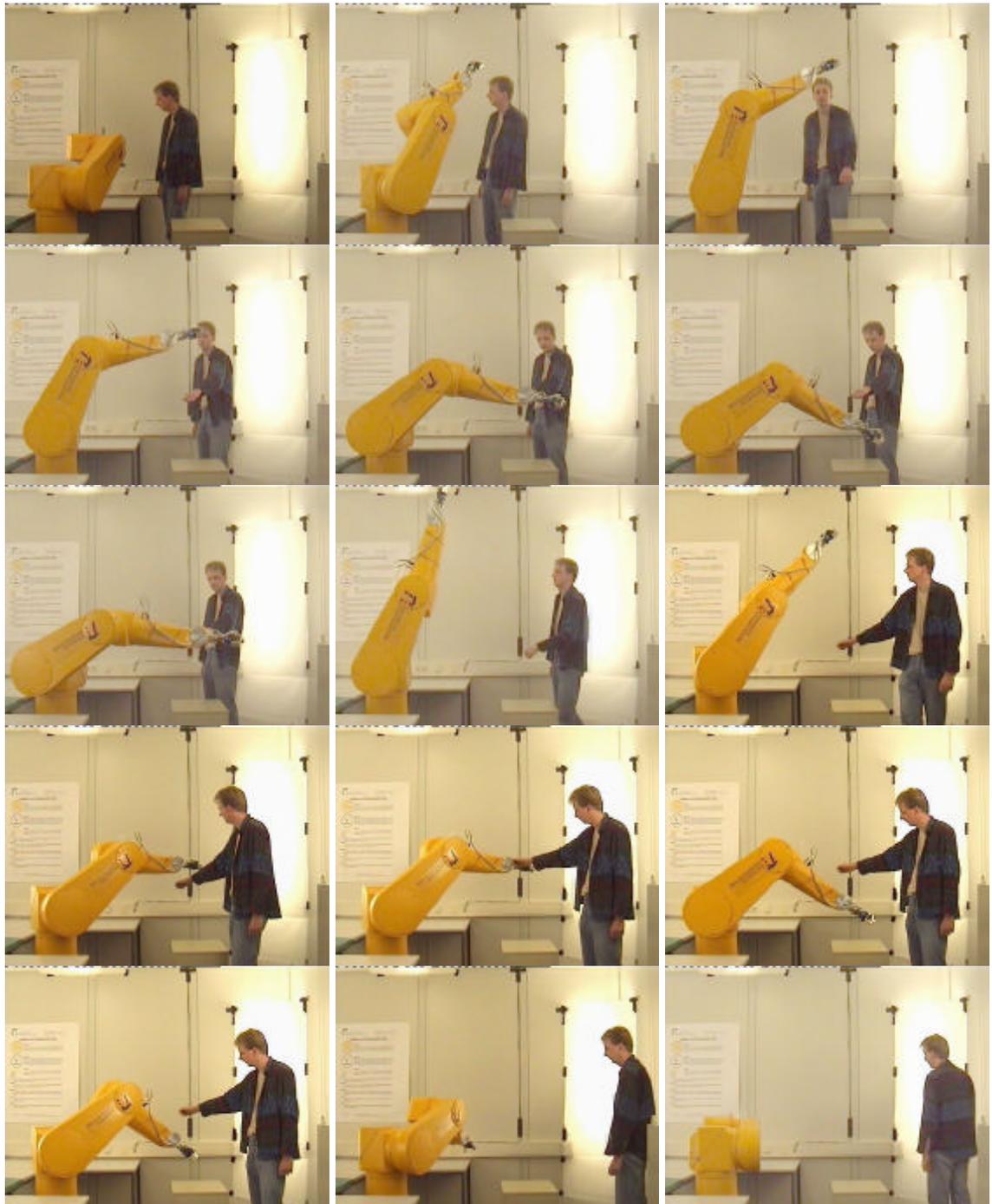


Abbildung 73: *Passieren im Arbeitsraum. Der Mensch betritt während des Roboterbetriebes die Zelle. Er läuft im Arbeitsraumes entlang, allerdings werden die Zielkonfigurationen des Roboters nicht blockiert (Passieren ohne Beschränkung des Bewegungsraumes).*

7.2.2 Wartung der Maschine

Die zweite Aufgabe simuliert die (vorbeugende) Wartung der Maschine. Der Mensch kommt während des Roboterbetriebs in die Zelle. Der Roboter muss erkennen, dass die Bearbeitungskonfiguration der Maschine belegt ist und warten, bis der Mensch mit der Wartung fertig ist.

Die Bildsequenz in Abbildung 74 zeigt eine Durchführung dieser Aufgabe. Die Bilder haben einen zeitlichen Abstand von einer Sekunde.



Abbildung 74: Vorbeugende Wartung einer Maschine im Arbeitsraum. Der Mensch betritt während des Roboterbetriebs die Zelle. Er führt die Wartungsaufgabe durch (hier: Putzen des Tisches) und verlässt den Arbeitsraum wieder. Der Roboter erkennt, dass die Zielkonfiguration bei der Maschine belegt ist und wartet, bis der Mensch aus dem Weg ist und setzt dann seine Arbeit fort.

7.2.3 Aufräumen in der Arbeitszelle

Die dritte Aufgabe ist das Aufräumen in der Arbeitszelle. Im Beispiel trägt der Mensch einen Kartons heraus. Der Roboter muss erkennen, welcher Raumbereich durch Mensch und Objekt belegt ist. Wenn es keinen Weg zum Ziel gibt, muss er anhalten, bis der Weg frei ist und dann seine Arbeit fortsetzen.

Die Abbildung 75 zeigt eine Durchführung dieser Aufgabe. Die Bilder haben einen zeitlichen Abstand von einer Sekunde.



Abbildung 75: Aufräumen der Arbeitszelle. Der Mensch betritt den Arbeitsraum während des Roboterbetriebes. Er führt seine Aufgabe durch und blockiert dabei zeitweilig verschiedene Zielkonfigurationen des Roboters. Der Roboter versucht sein Ziel so gut wie möglich zu erreichen. Nachdem der Mensch die Zelle verlassen hat, setzt der Roboter die Arbeit fort.

7.2.4 Ergebnis

Die Zyklusdauer für eine Ausführung der Aufgabe des Roboters in der leeren Zelle und bei der Aufgabe 1 (Passieren in der Zelle) wird ermittelt. Die Aufgabe wird in zwei Varianten durchgeführt. Bei der ersten Variante wird der dem Roboter zur Verfügung stehende Raum nicht eingeschränkt. Bei der zweiten Variante wird der Roboter durch den Menschen im Arbeitsraum in seiner Bewegungsfreiheit eingeschränkt.

Bei den Aufgaben 2 und 3 hängt die Auswirkung auf die Zykluszeit des Roboters vor allem davon ab, wie lange der Mensch den Weg des Roboters blockiert. Aus diesem Grund werden für diese Aufgaben keine Zykluszeiten ermittelt.

Der Einfluss der Störungen ist in Abbildung 76 zu sehen. Es zeigt sich, dass allein die Präsenz des Menschen im Arbeitsraum zu einer Vergrößerung der Zykluszeit führt. Passt der Mensch innerhalb des Bewegungsraums des Roboters, so wird die Zykluszeit des Roboters noch größer.

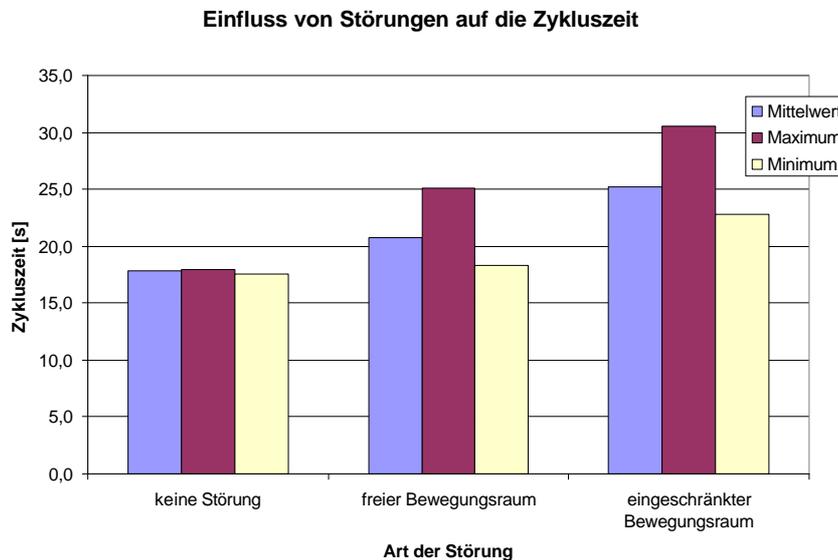


Abbildung 76: Einfluss von Störungen auf die Zyklusdauer. Dargestellt sind Mittelwert, Maximum und Minimum von 5 (keine Störung) bzw. 14 (Passieren ohne und mit Einschränkung des Bewegungsraumes) Versuchen.

7.3 Diskussion der erzielten Ergebnisse

Im Folgenden werden die erzielten Ergebnisse diskutiert und bewertet. Dabei werden alle Aspekte des realisierten Systems zur Sicherung der Transferbewegung betrachtet.

7.3.1 Einschränkungen des Systems

Prinzipielle Einschränkungen

Auf Grund der Rückprojektion werden Hindernisse durch ihre visuelle Hülle angenähert. Schmale Passagen werden durch die Hindernisvergrößerung meistens blockiert. Daher eignet sich das System nicht zur Navigation in engen Räumen. Hierzu sind weitere Sensoren notwendig. Das vorgestellte System eignet sich für die Szenarien mit relativ

wenigen und kompakten Hindernissen wie beim Benchmark „SIMPLE“ aus [Hwang92].

Bedingt durch die Verwendung von Silhouetten reagiert das System empfindlich auf Veränderungen des Hintergrundes. Diese Veränderungen können durch Bewegungen von Objekten außerhalb des gemeinsamen Arbeitsbereiches von Mensch und Roboter entstehen. Daher ist es notwendig, dass der Sichtbereich der Kameras auf die Arbeitszelle beschränkt bleibt.

Das System kann nicht in Kontakt zur Umwelt treten. Jedes Hindernis ist eine verbotene Zone. Damit ist eine aktive Mensch-Roboter-Kooperation nicht möglich, da der Roboter nicht auf den Menschen zu fahren kann, sondern nur in seine Nähe kommt. Aus dem selben Grund ist auch das Aufnehmen und Ablegen von Werkstücken nicht abgesichert möglich. Im realisierten System wird nur die Transferbewegung abgesichert. Greifen und Ablegen muss ungesichert erfolgen.

Einschränkungen durch die gewählte Realisierung

Im Moment kann das System noch keine großen Objekte transportieren, die den Roboter aus Sicht der Kameras verändern. Dies ließe sich durch ein Anpassen des Robotermodells leicht ändern. Allerdings müssten die Abmessungen der zu transportierenden Objekte bekannt sein.

Bedingt durch die Verwendung von Grauwertkameras reagiert das System empfindlich auf Veränderungen der Beleuchtung und auf Schattenwurf. Dies ließe sich durch die Verwendung von Farbkameras und einer Transformation der Pixel in einen störungsärmeren Farbraum verringern.

Ebenso eine Folge der Verwendung von Grauwertkameras ist die Tatsache, dass Objekte leichter dem Hintergrund ähneln können. Ein Beispiel hierfür ist der Roboter selbst. Sein Grauwert ist dem des Hintergrundes ähnlich, obwohl der Hintergrund grau und der Roboter orange-gelb ist. Diese Problematik ließe sich durch die Verwendung von Farbkameras entschärfen (aber nicht ganz vermeiden).

Die Roboterbewegung wird ab einer TCP-Geschwindigkeit von 0,7 m/s nicht mehr glatt. Dies liegt zum einen an der begrenzten Rechenleistung des Bildverarbeitungsrechners, zum anderen an der Latenz des Netzwerkes. Durch einen schnelleren Rechner und ein schnelles, echtzeitfähiges Netzwerk könnte die Robotergeschwindigkeit deutlich gesteigert werden.

Das bisherige System überprüft noch nicht alle wesentlichen Umweltannahmen. Zur Zeit wird noch nicht überprüft, ob die Kameras so ausgerichtet sind, dass sie den ganzen gemeinsamen Arbeitsraum überwachen. Bei kalibrierten Kameras und bekannten Dimensionen des gemeinsamen Arbeitsraumes, lässt sich der Anteil des überwachten Bereiches aus den Kameraparametern ableiten.

Der statische Kollisionstest überprüft nur die Gelenkpunkte des Roboters. Eine Überprüfung des gesamten Roboters ist aber kein prinzipielles Problem. Durch die Verwendung eines Speichers für die statische Konfigurationsraumbelegung wären während des Betriebs keine Auswirkungen auf die Laufzeit zu erwarten.

Weitaus aufwendiger ist die Überprüfung, ob der Sichtbereich der Kameras durch Objekte verdeckt ist. Auch bei korrekter Positionierung könnte der Sichtbereich einer Kamera zum Beispiel durch ein von der Decke hängendes Schild verdeckt werden. Diese

Kamera würde keine Veränderung in der Zelle bemerken. Dadurch würden Konfigurationen als kollisionsfrei klassifiziert, die in Wirklichkeit zu Kollisionen führen.

7.3.2 Leistung des Systems

In den durchgeführten Aufgaben wird gezeigt, dass das System erfolgreich die Transferbewegung eines Industrieroboters absichert. Das Versuchssystem verwendet vier einfache Grauwertkameras und einen einzigen Standard-PC zur Bildverarbeitung, Kollisionserkennung und Bahnplanung.

Trotzdem wird eine relativ zuverlässige Erkennung von Hindernissen erreicht und der Roboter kann um die meisten Hindernisse eng herumfahren. Die erreichbare Verfahrensgeschwindigkeit von 0,7 m/s des Roboters genügt, dass sich Menschen im Arbeitsraum unwohl fühlen und zusammenzucken, wenn der Roboter nahe an ihnen vorbeifährt.

Das angestrebte Ziel, ein System zu realisieren, das die Transferbewegung eines Industrieroboters mit Hilfe von Kameras absichert und dazu eine Ganzarmerkennung und eine globale Ausweichbewegung verwendet wurde bezüglich der dynamischen Objekte erreicht. Bezüglich der statischen Objekte wird eine Mehrpunkterkennung benutzt. Dies zu Verändern ist aber kein prinzipielles Problem.

Die Bewertung des realisierten Systems nach den selben Kriterien wie die in Kapitel 2 betrachteten Systeme ist in Tabelle 14 zu sehen.

	Globale Ausweichbewegung		Realisiertes System (statische Objekte)	Realisiertes System (dynamische Objekte)
Reaktionsstrategie	Lokale Ausweichbewegung		[Bischoff99] [Yu99] *)	[Feddema94] [Martín99] *) [Stettmer94]
	Geschwindigkeitsregelung	[Meisel94] *) [Morhard02] [Zettl02]		[Heiligensetzer02a] [Spingler02] [Zelinsky99]
	Kollisionsentschärfung			[Morhard02] [Heiligensetzer02a] [Lim00] [Zelinsky99]
		Schutzraum	Einzelpunkt	Mehrpunkt
	Erkennungsstrategie			

Tabelle 14: Einordnung des realisierten Systems und der Systeme aus der Literatur. Systeme die mehrere Strategien verfolgen, sind in allen zutreffenden Klassen aufgeführt. Bei Systemen, zu denen mehrere Publikationen existieren, ist nur die jeweilige Hauptquelle aufgeführt.

**) Autor betrachtet andere Aufgabenstellung. Einordnung ist anhand der technischen Möglichkeiten erfolgt.*

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

Aus Sicherheitsgründen müssen zur Zeit die Arbeitsräume von Industrierobotern und Menschen durch Schutzeinrichtungen getrennt werden. Die strengen Anforderungen, die Normen wie zum Beispiel die Industrierobotersicherheitsnorm ISO10218 vorschreiben, kommen daher, dass die Roboter ihre Umwelt nicht wahrnehmen können. Für viele Bereiche ist es wünschenswert, Mensch und Roboter im selben Arbeitsraum arbeiten zu lassen. Dies sind zum Beispiel Hol- und Bringdienste, bei denen der Roboter einen Menschen von zeitraubenden Tätigkeiten entlastet.

Eine Bewertung der verschiedenen Roboterbewegungen zeigt, dass die für einen Menschen im Arbeitsraum gefährlichste Bewegung, die freie Transferbewegung ist. Denn dabei werden schnelle Bewegungen über größere Distanzen ausgeführt. Die betrachtete Aufgabe besteht daher darin, die Transferbewegungen eines Manipulators kollisionsfrei durchzuführen. Von einer gegebenen Startkonfiguration aus soll der Manipulator auf einer möglichst kurzen, kollisionsfreien Bahn in eine gegebene Zielkonfiguration fahren.

Das realisierte System gliedert sich in die vier Hauptkomponenten: Bildverarbeitung, Robotermodellierung, Kollisionserkennung und Bahnplanung.

Aufgabe der Bildverarbeitung ist es, ein Silhouettenbild der aktuellen Szene zu erstellen. Ein Silhouettenbild ist ein binäres Bild, in dem alle Vordergrundpixel gesetzt und alle Hintergrundpixel gelöscht sind. Die robuste und schnelle Erstellung eines Silhouettenbildes aus Grauwertbildern ist ein relativ schwieriges Problem. Der Grund hierfür ist der eindimensionale Farbraum der Kameras. Das zu messende Signal (echter Grauwert eines Szenenpunktes) ist direkt mit allen Störgrößen (z.B. Schatten) überlagert. Im realisierten System wird ein kachelbasierter Ansatz verwendet. Die Grauwertbilder werden in Kacheln unterteilt und für jede dieser Kacheln werden Merkmale berechnet. Ein Klassifikator vergleicht die Merkmale der aktuellen Szene mit den Merkmalen der selben Kachel im Referenzzustand. Anhand der Differenz wird entschieden, ob die Kachel Vordergrund oder Hintergrund ist.

Die Aufgabe der Robotermodellierung besteht darin, die Kollisionserkennung mit Wissen über die Geometrie des Roboters zu versorgen. Im realisierten System wird eine bildbasierte Kollisionserkennung verwendet. Daher muss hier die Robotermodellierung zu einer angefragten Konfiguration ein Silhouettenbild des Roboters liefern. Dieses Silhouettenbild wird durch die Berechnung von mehreren Modellen erzeugt. Zuerst wird die Lage der Roboterglieder im Raum durch das kinematische Modell berechnet. An-

schließlich wird auf dieser Basis das Volumen des Roboters durch Pyramidenstümpfe approximiert. Diese Pyramidenstümpfe werden in die Kameras projiziert. Dadurch entsteht die Robotersilhouette.

Die Aufgabe der Kollisionserkennung ist es, zu entscheiden, ob eine bestimmte Konfiguration zu einer Kollision mit einem Objekt führen könnte, wenn der Roboter diese einnimmt. Dabei wird lediglich diese angefragte Konfiguration betrachtet, egal ob es möglich ist diese Konfiguration aus der gegenwärtigen Konfiguration zu erreichen. Der im realisierten System verwendete Kollisionstest baut keine Repräsentation des 3D-Arbeitsraumes auf, sondern arbeitet ausschließlich auf den Silhouettenbildern von Szene und Roboter. Es gibt verschiedene Verfahren für unkalibrierte und kalibrierte Kameras. Außerdem werden Verfahren beschrieben, die mehr als eine Konfiguration auf einmal testen können. Alle diese Kollisionstests basieren auf dem Schnitt von Szenensilhouette, aktueller und zu testender Robotersilhouette.

Die Bahnplanung hat die Aufgabe, ausgehend von einer gegebenen Startkonfiguration, eine kollisionsfreie Bahn zu einer gegebenen Zielkonfiguration zu finden. Dieses Problem ist in der Robotik gut untersucht. Bislang wurden jedoch hauptsächlich Probleme betrachtet, in denen der betrachtete Raum statisch und vollständig bekannt ist. Im Anwendungsszenario ist die Umgebung unbekannt und ändert sich ständig. Dies stellt besondere Anforderungen an die Bahnplanung. In der klassischen Bahnplanung kann dem Ergebnis der Kollisionserkennung blind vertraut werden. Bei einem System, das die Umwelt mit Sensoren erfasst, ist dies nicht so, da in der Sensorerfassung immer Fehler auftreten können. Da die Umwelt außerdem nicht statisch ist, muss eine einmal gefundene Bahn während der Ausführung immer wieder auf Kollisionen überprüft werden. Wenn keine Bahn zum Ziel möglich ist, sollte möglichst nahe zum Ziel gefahren werden. Dies hat zum einen den Grund, dass unter Umständen das Ziel nur wegen einer fehlerhaften Kollisionserkennung nicht erreicht wird, falls die Qualität der Kollisionserkennung abnimmt mit der Entfernung der getesteten Konfiguration zur aktuellen Konfiguration. Zum anderen kann das Hindernis, das den Weg zum Ziel blockiert, ein kooperierender Mensch sein. Nur wenn der Mensch erkennt, dass er dem Roboter im Weg ist, wird er den Weg freimachen.

Das oben beschriebene Systemkonzept wurde realisiert. Die verwendete Hardware besteht aus einem konventionellen 6-Achs Industrieroboter, einem Standard-PC zur Bildverarbeitung und Bahnplanung. Die Versuche zeigen, dass der Ansatz zur Lösung der gestellten Aufgabe geeignet ist. Mögliche Erweiterungen des Systems sind die direkte Mensch-Roboter-Kooperation oder das gemeinsame Arbeiten an Werkstücken.

8.2 Ausblick

Zukünftige Arbeiten lassen sich in zwei Bereiche unterteilen. Zum einen kann das realisierte System verbessert werden. Hierzu sind im vorherigen Kapitel einige Möglichkeiten vorgestellt. Diese Veränderungen machen das realisierte System robuster und bringen es näher an eine praktische Anwendbarkeit. Sie stellen aber keinen prinzipiellen Fortschritt dar.

Andererseits sind auf der Basis des realisierten Systems auch weitergehende Erweiterungen möglich, die aber eine grundlegende Änderung des Systems nach sich ziehen.

Eine wesentliche Einschränkung des Systems ist, dass es keinen sicheren Kontakt zu Objekten oder Menschen herstellen kann. Das vorgestellte System sichert die Transferbewegung eines Industrieroboters ab. Greif-, Ablege- und Montageoperationen sind

damit nicht gesichert möglich. Auch wenn das Risiko einer Kollision mit dem Menschen bei diesen Operationen geringer ist als bei der Transferbewegung, sollte das System so erweitert werden, dass es auch diese Bewegungsarten absichert.

Dieser direkte Kontakt erfordert aber eine Sensorik, die zwischen Hindernissen unterscheiden kann. Mit einer solchen Sensorik ließen sich ein direkter Roboter-Werkstück-Kontakt und ein direkter Roboter-Mensch-Kontakt sicher realisieren. Um dies mit Kameras zu realisieren müsste aber die einfache Silhouettenerzeugung aufgegeben werden.

Eventuell kann der Kontakt aber auch durch die Verwendung von zusätzlichen Sensoren abgesichert werden. Dies wäre zum Beispiel durch eine künstliche Haut mit Infrarotabstandssensoren wie in [Lumelsy98] möglich. Ein weiterer Vorteil dieser zusätzlichen Sensoren wäre die Navigation in engen Räumen. Das realisierte System kann nur unter speziellen Randbedingungen in engen Räumen fahren.

Während der Arbeiten am realisierten System zeigte sich, dass die automatisierte Kamera- und Beleuchtungspositionierung ein noch wenig bearbeitetes Gebiet ist. Zur Zeit werden die Kameras und Lichtquellen der meisten Systeme „nach Gefühl“ positioniert. Dies ist auch im realisierten System der Fall. Aus wissenschaftlicher Sicht wäre eine grundlegende Lösung dieses Problems sehr interessant. Außerdem gäbe es für einen Ansatz, der das Kamera- und Beleuchtungsproblem löst, eine Vielzahl von Anwendungsmöglichkeiten.

Ebenso wurde während der Arbeit am System, ein schneller, echtzeitfähiger Bahnplaner vermisst. Das Problem dabei ist, dass der Planungsvorgang des einzelnen Planers sehr lange dauern kann. Das Problem tritt dann auf, wenn nach außen hin eine harte Zeitschranke einzuhalten ist. Dieses Problem tritt auch bei anderen Anwendungen auf. Eine allgemeine Lösung des Problems wäre sehr wünschenswert.

Anhang A: Literaturverzeichnis

- [Abmayr94] W. Abmayr, „Einführung in die digitale Bildverarbeitung“, Teubner, Stuttgart, 1994
- [Ameling96] Ameling, W. (Hrsg.): „Flexible Handhabungsgeräte im Maschinenbau“; Ergebnisse aus dem Sonderforschungsbereich 208, VCH Verlag, 1996
- [Baginski97] Baginski, B.: „Efficient Dynamic Collision Detection using Expanded Geometry Models“, In: Proc. of IEEE/RSJ/GI International Conference on Intelligent Robots and Systems IROS'97, Grenoble, September 7-11, pp. 1714-1719, 1997
- [Baginski99] Baginski, B.: „Motion Planning for Manipulators with Many Degrees of Freedom - The BB-Method“, Dissertaion, Technische Universität München, Fakultät für Informatik, Dissertationen zur Künstlichen Intelligenz Bd. 195, 1999
- [Bertagnolli02] Bertagnolli, F., Ziegler, M., Dillmann, R.: „Mobile Roboter auf dem Weg zur CE-Zertifizierung“, In: VDI-Bericht 1679 - Tagungshandbuch zur Robotik 2002, Ludwigsburg, pp. 669-675, 2002
- [Bischoff99] Bischoff, A.: „Echtzeit Kollisionsvermeidung für einen Industrieroboter durch 3D-Sensorüberwachung“, Diplomarbeit Fernuniversität Hagen, 1999
- [Diehl01] Diehl, T.: „SIMERO – Verfahren zur Erzeugung stabiler Differenzbilder“, Projektarbeit, AG Eingebettete Systeme und Robotik, FB Informatik, Universität Kaiserslautern, 2001
- [Dornier93] Dornier GmbH: „Range Imaging Camera (EBK)“, Produktionsinformation, Dornier GmbH, Optoelectronics VFB 317, 1993
- [Duda73] Duda, R., Hart, P.: „Pattern Classification and Scene Analysis“, John Wiley & Sons, New York, 1973
- [Ebert01] Ebert, D., Henrich, D.: „Safe Human-Robot-Cooperation: Problem Analysis, System Concept and Fast Sensor Fusion" In: IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 239-244, Baden-Baden, Germany, August 20 - 22, 2001
- [Ebert02a] Ebert, D., Henrich, D.: „SIMERO - Sicherheitsstrategien für die Mensch-Roboter-Kooperation" In: „OTS-Systeme in der Robotik – Roboter Ohne Trennende Schutzeinrichtungen“, Reihe BKM Berichte, Herbert Utz Verlag, S. 5.1-5.17, München, 25. Juni 2002

- [Ebert02b] Ebert, D., Henrich, D.: „Safe Human-Robot-Cooperation: Image-based collision detection for Industrial Robots" In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1826-1831, Lausanne, October 2 - 4, 2002
- [Ebert03] Ebert, D., Henrich, D.: „SIMERO - Sichere Mensch-Roboter-Koexistenz“, Angenommen für „2. Workshop für OTS-Systeme in der Robotik – Mensch und Roboter ohne trennende Schutzsysteme“, Stuttgart, 24. Juni 2003
- [Eckert00] Eckert, G.: „Automatic Shape Reconstruction of Rigid 3-D Objects from Multiple Calibrated Images“, Eusipco 2000 Proceedings, Tampere, Finland , 2000
- [Eisert00] Eisert, P., Steinbach, E., Girod, B.: „Automatic Reconstruction of Stationary 3-D Objects from Multiple Uncalibrated Camera Views", In: IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on 3D Video Technology, vol. 10, no. 2, pp. 261-277, March 2000
- [Feddema94] Feddema, J.T., Novak, J.L.: „Whole Arm Obstacle Avoidance for Teleoperated Robots“, In: IEEE Robotics and Automation Proceedings, pp.3303 – 3309, 1994
- [Ferber99] Ferber, J.: „Multi-Agent-Systems An Introduction to distributed artificial intelligence“, Addison-Wesley, New York, Bonn, Tokyo, ISBN 0-201-36038-9, 1999
- [Fiegiert96] Fiegiert, M.: „Experimentelle Untersuchung zur Umweltmodellierung mit einem 3D-Laserscanner“, Diplomarbeit, Institut für Prozeßrechenetechnik & Robotik, Universität Karlsruhe, 1996
- [Fujimura91] Fujimura, Kikuo: “Motion Planning in Dynamic Environments”, Springer Verlag, Tokyo, Berlin, New York, 1991
- [Gaskill96] Gaskill, S.P., Went, S.R.G.: „Safety issues in modern applications of robots“, In: Reliability Engineering and System Safety 53, Elsevier Science Limited, Northern Ireland, pp. 301-307, 1996
- [Gecks01] Gecks, T.: „SIMERO – Software System Design and Implementation“, Projektarbeit, AG Eingebettete Systeme und Robotik, FB Informatik, Universität Kaiserslautern, 2001
- [Gecks03] Gecks, T.: „SIMERO - Erzeugung von flüssigen, schnellen Roboterbewegungen“, Diplomarbeit, AG Eingebettete Systeme und Robotik, FB Informatik, Universität Kaiserslautern, 2003
- [Gupta98] Gupta Kamal, Angel del Pobil (ed.): “Practical Motion Planning in Robotics; Current Approaches and Future Directions”, John Wiley & Sons Ltd, Chichester, England, 1998
- [Hamfeld02] Hamfeld, H.: „Aktive Stereoskopie – Neue Verfahren zur dreidimensionalen Vermessung von Objekten“, Schriftenreihe Informatik Band 4, Verlag Universität Kaiserslautern, 2002

- [Henrich98] Henrich, D., Wurll, Ch., Wörn, H.: „On-line path planning with optimal C-space discretization”, In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98), Victoria, Canada, Oct. 12-16, 1998
- [Heiligensetzer02a] Heiligensetzer, P.: „Sichere Mensch-Roboter Kooperation für Roboter im niedrigen Traglastbereich“ In: OTS-Systeme in der Robotik, Reihe BKM Berichte, Herbert Utz Verlag, München, 2002
- [Heiligensetzer02b] Heiligensetzer, P.: „Sichere Mensch-Roboter Kooperation in der industriellen Fertigung“, In: Robotik 2002, VDI-Berichte Nr. 1679, pp. 197-202, 2002
- [Heinzen03] Heinzen, F.: „SIMERO – Robuste und Schnelle Erzeugung von Silhouetten aus Grauwertbildern“, Diplomarbeit, AG Eingebettete Systeme und Robotik, FB Informatik, Universität Kaiserslautern, 2003
- [Heinzmann99] Heinzmann, J., Zelinsky, A.: „A Safe-Control Paradigm for Human-Robot Interaction“, In: Journal of Intelligent and Robotic Systems, Issue 25, 1999, Kluwer Academic Publishers, pp. 295-310, 1999
- [Heintz02] Heintz, M.: „SIMERO : Implementierung und Vergleich von Bahnplanungsverfahren im Konfigurationsraum“, Projektarbeit, AG Eingebettete Systeme und Robotik, FB Informatik, Universität Kaiserslautern, 2002
- [Horprasert00] Horprasert, T., Harwood, D., Davis, L.S.: „A Robust Background Subtraction and Shadow Detection”, In: Proc. ACCV'2000 Taipei, Taiwan, January 2000
- [Hwang92] Hwang, Y., Ahuja, N.: „Gross Motion Planning: A Survey“, In: ACM Computing Surveys, 24(3), pp. 219-291, 1992
- [Ivanov00] Ivanov, Y., Bobick, A., Liu, J.: „Fast Lighting Independent Background Subtraction”, In: International Journal of Computer Vision 37(2): 199-207, June 2000
- [ISO10218] ISO 10218, EN 775: „Manipulating industrial robots – Safety”, (ISO 10218 modified), 1992
- [Khodabandehloo96] Khodabandehloo, K.: „Analyses of robot systems using fault and event trees: case studies“, In: Reliability Engineering and System Safety 53, Elsevier Science Limited, Northern Ireland, pp. 247-264, 1996
- [Körte03] Körte, J.: „SIMERO : Implementierung und Analyse von bildbasierten Kollisionstests“, Projektarbeit, AG Eingebettete Systeme und Robotik, FB Informatik, Universität Kaiserslautern, 2003
- [Latombe96] Latombe, J-C.: „Robot motion planning”; 4th print.; Boston : Kluwer Academic Publishing, 1996.
- [Leou91] Leou, J. J., Chang, Y. L., Wu, J.S.: „Robot operation monitoring for collision avoidance by image sequence analysis“, In: Pattern Recognition, Vol. 25, No. 8, pp. 855-867, 1992

- [Lim99] Lim, H., Tanie, K.: „Collision-Tolerant Control of Human-Friendly Robot With Viscoelastic Trunk“, In: IEEE/ASME Transactions on Mechatronics, Vol. 4 No. 4, December 1999, Published by IEEE Periodicals Transactions/Journal Department, USA, pp.417 – 427, 1999
- [Lim00] Lim, H., Tanie, K.: „Human Safety Mechanisms of Human-Friendly Robots: Passive Viscoelastic Trunk and Passively Movable Base“, In: The International Journal of Robotics Research Vol. 19 No. 4 April 2000 pp.307-335, Sage Publications, 2000
- [Lumelsky93] Lumelsky, V.; Cheung, E.: “Real-Time Collision Avoidance in Teleoperated Whole-Sensitive Robot Arm Manipulators”, IEEE Transactions on Systems, Man and Cybernetics, Vol.23 No.1, pp.194-203,1993
- [Lumelsky98] Lumelsky, V.; Cheung, E.: “Real-Time Collision Avoidance in Teleoperated Whole-Sensitive Robot Arm Manipulators”, In: “Practical Motion Planning in Robotics; Current Approaches and Future Directions”, reprint of: IEEE Transactions on Systems, Man and Cybernetics, Vol.23 No.1, pp.194-203, John Wiley & Sons Ltd, Chichester, England, 1998
- [Maeda02] Maeda, Y., Ishido, N., Kikuchi, H., Arai, T.: „Teaching of Grasp/Graspless Manipulation for Industrial Robots by Human Demonstration“ In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1523-11528, Lausanne, October 2 - 4, 2002
- [Martín99] Martín, P., Millán, J.: „Learning of Sensor-Based Arm Motions while Executing High-Level Descriptions of Tasks“, In: Autonomous Robots Vol. 7, Aug. 1999, Kluwer Academic Publishers, pp. 57-75, 1999
- [MarquesDeSá01] MarquesdeSá, J. P.: „Pattern Recognition – Concepts, Methods and Applications“, Springer-Verlag, Berlin 2001
- [Matusik00] Matusik, W., Buehler, C., Raskar, R., McMillan, L., Gortler, S.: „Image-Based Visual Hulls.“ In Proceedings of SIGGRAPH, 2000.
- [Morhard02] Morhard, D.: „Realisierung eines OTS-Systems bei einem Zulieferer der Kunststoffindustrie“, In: OTS-Systeme in der Robotik, Reihe BKM Berichte, Herbert Utz Verlag, München, pp. 8.1 – 8.12, 2002
- [Meisel91] Meisel, A., Föhr, R., Ameling, W.: „3D-Kollisionsschutzsensor auf der Basis von CCD-Kameras“, In SENSOR 91, Seiten 157-170, Nürnberg, Mai 1991
- [Meisel94] Meisel, A.: „3D-Bildverarbeitung für feste und bewegte Kameras“, Vieweg Verlag, Reihe Fortschritte der Robotik Nr. 21, 1994
- [Niem97] Niem, W.: „Error Analysis for Silhouette-Based 3D Shape Estimation from Multiple Views“, Proc. on Int. Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging, Rhodos, September 1997

- [Noborio92] Noborio, H., Katahira, Y.: „A Stereo Volume Intersection Method for Reconstruction of 3D Multiple Objects“, Proceedings of the 1992 Second International Conference on Automation, Robotics and Computer Vision, pp.CV-5.3.1-CV.5.3.5, Sept. 1992
- [Noborio98] Noborio, H., Yoshioka, T.: „Sensor-Based Navigation of a Mobile Robot under Uncertain Conditions“, In: “Practical Motion Planning in Robotics; Current Approaches and Future Directions”, John Wiley & Sons Ltd, Chichester, England, 1998
- [Noborio99] Noborio H., Maeda, Y., Kenji Urakawa: „Three or More Dimensional Sensor-Based Path-Planning Algorithm HD-I“ IEEE International Conference on Intelligent Robots and Systems, pp.1699-1706 1999
- [Noborio01] Noborio, H., Nishino, Y.: “Image-based Path-Planning Algorithm on the Joint Space”, In: IEEE International Conference on Robotics and Automation, pp. 1180-1187, Seoul, 2001
- [Novak92a] Novak, J.L.; Feddema, J.T.: “Collision avoidance during teleoperation using whole arm proximity sensors coupled to a virtual environment”, In: SPIE Proceedings, Vol. 2057, pp. 2 – 9, 1992
- [Novak92b] Novak, J.L.; Feddema, J.T.: ”A Capacitance-Based Proximity Sensor for Whole Arm Obstacle Avoidance”, IEEE Proceedings of the Intl. Conf. on Robotics and Automation, pp. 1307-1314, 1992
- [Quick96] Quick, G., Müller, P.: „A Novel Approach to Collision Avoidance of Robots Using Image Planes“, In: Robotics and Manufacturing – Recent Trends in Research and Applications, Vol. 6, pp. 567-573, 1996
- [Rosin95] Rosin, P.L., Ellis, T.: „Image difference threshold strategies and shadow detection“, 6th British Machine Vision Conf., Birmingham, pp. 347-356 1995.
- [Rosin98] Rosin, P.L.: „Thresholding for Change Detection“, Int. Conf. Computer Vision, pp. 274-279, 1998
- [Schraft98] Schraft, R.D., Schmierer, G.: „Serviceroboter: Produkte, Szenarien, Visionen“, Springer Verlag, Berlin, Heidelberg, New York, Barcelona, Budapest, Hongkong, London, Mailand, Paris, Santa Clara, Singapur, Tokio, 1998
- [Siegert99] Siegert, H.-J.: „Weg- und Trajektorienplanung“, Skript zur Vorlesung, AG Echtzeitsysteme und Robotik, TU München, 1998
- [Spingler02] Spingler, J., Thiemermann, S.: „Direkte Mensch-Roboter Kooperation in der flexiblen Montagezelle“, In: Robotik 2002, VDI-Berichte Nr. 1679, pp. 191-195, 2002
- [Stettmer94] Stettmer, J.: „Sensorgestützte Kollisionsvermeidung bei Industrierobotern“, Reihe: Berichte aus der Produktionstechnik; Bd. 23/94, Shaker Verlag; Aachen, 1994

- [Souccar98] Souccar, K., Coelho, J., Connolly C., Grupen, R.: „Harmonic Functions for Path Planning and Control“, In: “Practical Motion Planning in Robotics; Current Approaches and Future Directions”, John Wiley & Sons Ltd, Chichester, England, 1998
- [Thiemermann02] Thiemermann, S.: „team@work – Direkte Mensch-Roboter Kooperation“, In: OTS-Systeme in der Robotik, Reihe BKM Berichte, Herbert Utz Verlag, München, pp. 4.1 – 4.5, 2002
- [Um98] Um, D., Lumelsky, V.: „Component Redundancy-Based Fault Tolerance System for a Modular Sensitive Skin“, Technical Report, Robotics Laboratory, University of Wisconsin-Madison, 1998
- [Urbanczik03] Urbanczik, Ch.: „SIMERO: Bildbasierte Kollisionserkennung und Bahnglättung im Konfigurationsraum“, Diplomarbeit, AG eingebettete Systeme und Robotik, FB Informatik, Universität Kaiserslautern, 2003
- [Wegerif92] Wegerif, D.: “Sensor-based whole-arm obstacle avoidance for kinematically redundant robots”, In: Proceedings of SPIE Vol 1828 Sensor Fusion V, pp.417-426, 1992
- [Wenzel03] Wenzel, P.: „SIMERO: Überwachung der Funktion eines Systems zur sicheren Mensch-Roboter-Kooperation“, Projektarbeit, AG eingebettete Systeme und Robotik, FB Informatik, Universität Kaiserslautern, 2003
- [Yu99] Yu, Y., Gupta, K.: „Sensor-Based Roadmaps for Motion-Planning for Articulated Robots in Unknown Environment: Some Experiments with an Eye-in-hand System“ IEEE International Conference on Intelligent Robots and Systems, pp.1707-1714, 1999
- [Zelinsky99] Zelinsky, A., Matsumoto, Y., Heinzmann, J., Newman, R.: "Towards Human Friendly Robots: Vision-based Interfaces and Safe Mechanisms", In Experimental Robotics VI (Lecture Notes In Control and Information Sciences 250), Peter Corke and James Trevelyan (Eds), Springer, pp.487-498, 1999.
- [Zettl02] Zettl, H.: „Verwirklichte OTS-Systeme und Gedanken zur Weiterentwicklung“, In: OTS-Systeme in der Robotik, Reihe BKM Berichte, Herbert Utz Verlag, München, pp. 9.1 – 9.13, 2002

Anhang B: Sonstige Quellen

- [1] http://www.thetech.org/exhibits_events/online/robotics/universal/breakout_p01_rur.html
- [2] <http://ascii24.com/news/i/hard/article/2002/03/19/634534-000.html?top>
- [3] <http://www.uwec.edu/academic/curric/jerzdg/RUR/>
- [4] <http://www.wikipedia.org/wiki/Robot>
- [5] http://www.fraunhofer.de/german/press/md/md2001/md04-2001_fo4.html
- [6] <http://www.elan.de/tweb/download/ba7db9693f1.pdf>
- [7] <http://www.merritt-systems.com> (Zum Zeitpunkt des Drucks sind die verwendeten Abbildungen nicht mehr vorhanden)
- [8] <http://www.theimagingsource.com/prod/cam/dmk73/dmk73.htm>
- [9] <http://www.sick.de/de/products/categories/safety/espe/laserscanner/pls/de.html>
- [10] <http://www.minolta3d.com/specs.html>
- [11] <http://www.theimagingsource.com/prod/grab/dfgbw1/dfgbw1.htm>

Lebenslauf des Verfassers



Dirk Martin Ebert wurde am 25. Dezember 1973 in Freiburg im Breisgau geboren. Er besuchte von 1980 bis 1984 die Grundschule in Sinsheim an der Elsenz und von 1984 bis 1986 das dortige Wilhelmi-Gymnasium. Von 1986 bis 1993 besuchte er das Eugen-Bolz Gymnasium in Rottenburg am Neckar. Dort machte er im Mai 1993 das Abitur. Nach dem Abitur leistete er bis 1994 Zivildienst in Sulz am Neckar.

Zum Wintersemester 1994/95 begann Dirk Ebert das Studium des Wirtschaftsingenieurwesens an der Universität Karlsruhe. Er bestand das Vordiplom im Jahr 1996. Während des Hauptstudiums legte Dirk Ebert einen Schwerpunkt in den Bereich Automation und Robotik. Er belegte in diesem Bereich Seminare, Studien- und Diplomarbeit sowie Hauptdiplomsprüfungen am Institut für Prozessrechentechnik, Automation und Robotik (IPR).

Nach Abschluss des Diploms im Februar 2000 begann Dirk Ebert am 1. März 2000 seine Arbeit als wissenschaftlicher Mitarbeiter an der Universität Kaiserslautern im Fachbereich Informatik in der Arbeitsgruppe „Eingebettete Systeme und Robotik (RESY)“ von Prof. Dr. Dominik Henrich. Seit Oktober 2000 ist er angenommener Doktorand des Fachbereiches Informatik.

Das Forschungsinteresse von Dirk Ebert liegt im Gebiet der Mensch-Roboter-Kooperation. Sein Ziel ist es, dem Menschen einen ungefährdeten Aufenthalt im Arbeitsbereich eines arbeitenden Roboters zu ermöglichen. Sein Konzept zur Lösung dieses Problems stellte er im Jahr 2001 auf der MFI-Konferenz²¹ vor. Im Jahr 2002 konnte er das entstehende System auf dem 1. OTS-Workshop²² vor einem Fachpublikum aus Industrie und Wissenschaft präsentieren. Im Herbst 2002 veröffentlichte er den Kernalgorithmus zur bildbasierten, globalen Ganzarmkollisionserkennung auf der großen, internationalen Konferenz IROS²³.

Im Juni 2003 wurde Dirk Ebert eingeladen auf dem 2. OTS-Workshop²⁴ über das im Rahmen seiner Promotion entstandene System zur sicheren Mensch-Roboter-Koexistenz zu referieren.

²¹ IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems ([Ebert01])

²² OTS-Systeme in der Robotik - Roboter Ohne Trennende Schutzeinrichtungen ([Ebert02a])

²³ IEEE/RSJ International Conference on Intelligent Robots and Systems ([Ebert02b])

²⁴ 2. Workshop für OTS-Systeme in der Robotik - Mensch und Roboter ohne trennende Schutzeinrichtungen ([Ebert03])