

Sicherheitsstrategien für die Mensch-Roboter-Kooperation – Das SIMERO-System

Dirk Ebert

Wissenschaftlicher Mitarbeiter, AG Eingebettete Systeme und Robotik, Universität
Kaiserslautern

Prof. Dr. Dominik Henrich

Leiter der Arbeitsgruppe Eingebettete Systeme und Robotik, Universität
Kaiserslautern

Universität Kaiserslautern, Erwin-Schrödinger-Straße, D-67663 Kaiserslautern

Zur Zeit haben Industrieroboter nur eine sehr begrenzte Wahrnehmung ihrer Umwelt. Wenn sich Menschen im Arbeitsraum des Roboters aufhalten sind sie daher gefährdet. Durch eine Einteilung der möglichen Roboterbewegung in verschiedene Klassen kann gezeigt werden, dass die für einen Menschen im Arbeitsraum gefährlichste Bewegung die freie Transferbewegung ist. Daher besteht die betrachtete Aufgabe darin, diese Transferbewegung eines Manipulators durchzuführen, ohne mit dynamischen Hindernissen, wie zum Beispiel Menschen, zu kollidieren. Das SIMERO-System gliedert sich in die vier Hauptkomponenten Bildverarbeitung, Robotermodellierung, Kollisionserkennung und Bahnplanung. Diese Komponenten werden im einzelnen vorgestellt. Die Leistungsfähigkeit des Systems und die weiteren Verbesserungen werden an einem Versuch exemplarisch gezeigt.

1 Motivation

Aus Sicherheitsgründen müssen zur Zeit die Arbeitsräume von Industrierobotern und Menschen durch Schutzeinrichtungen getrennt werden. Die strengen Anforderungen, die Normen wie zum Beispiel [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.

In der Abbildung 1 sind ist eine Unterteilung der Roboterbewegungen anhand der Art der Interaktion mit Objekten dargestellt. Es wird unterschieden zwischen Bewegungen, die bei denen sich die Kontaktsituation eines Objektes im Greifer des Roboters verändert und ob das Objekt im Greifer mit weiteren Objekten der Umwelt in Interaktion tritt. Die Gefährdung des Menschen durch den Roboter ist je nach Bewegungsart des Roboters unterschiedlich. Bei schnellen Bewegungen ist sie größer 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 und über kurze Entfernungen. Die für einen Menschen im Arbeitsraum gefährlichste Bewegung ist die freie Transferbewegung, denn dabei werden schnelle Bewegungen über größere Distanzen ausgeführt.

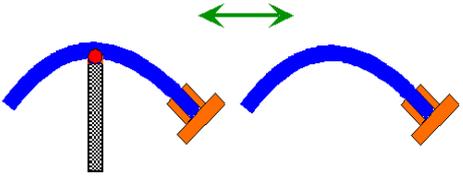
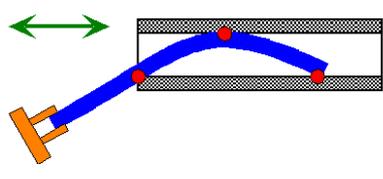
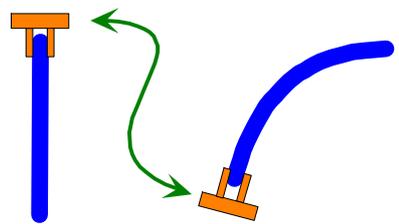
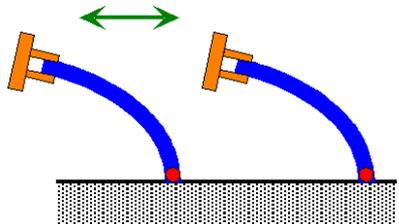
	Werkstück / Werkzeug (zeitweise) ohne Kontakt	Werkstück / Werkzeug mit Kontakt
Bewegung mit Änderung der Kontaktsituation	<p>Aufgreifen / Ablegen</p> 	<p>De-/Montieren</p> 
Bewegung ohne Änderung des Kontaktsituation	<p>Freier Transfer</p> 	<p>Bedingter Transfer</p> 

Abbildung 1: Klassifikation der Roboteroperationen am Beispiel von Montageaufgaben



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.

2 SIMERO-System

Das Anwendungsszenario ist folgendes: Ein Anwendungsprogramm, das eine abgesicherte Transferbewegung ausführen möchte, ruft die SIMERO-Bibliothek mit der gewünschten Start- und Zielkonfiguration des Roboters auf. Daraufhin startet ein Bahnplaner mit der Suche nach einer Bahn zwischen Start und Ziel. Dazu muss der Bahnplaner von einem Kollisionstest erfahren, welche Konfigurationen des Roboters kollisionsfrei sind. Der Kollisionstest erkennt dies, in dem er die aktuelle Szene und die Kinematik und Geometrie des Roboters berücksichtigt. Die aktuelle Szene wird durch die Bildverarbeitung ermittelt, das Wissen über die Geometrie des Roboters kommt von einem Robotermodell.

Das SIMERO-System gliedert sich entsprechend in die vier Hauptkomponenten Bildverarbeitung, Robotermodellierung, Kollisionserkennung und Bahnplanung. Die einzelnen Komponenten werden im folgenden detailliert vorgestellt.

2.1 Bildverarbeitung

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 Gegensatz dazu kann bei RGB-Farbkameras eine Transformation im Farbraum vorgenommen werden, bei der Signal und Störung sich unterschiedlich auf die verschiedenen Dimensionen verteilt. Ein Beispiel hierfür ist in [Horprasert00] zu finden. In [Rosin95,98] werden verschiedene Verfahren zur automatischen Berechnung eines Schwellwertes auf Grauwertbildern vorgestellt. Außerdem wird ein Ansatz gezeigt, der unter bestimmten Umständen Schatten als solche erkennen kann. In [Ivanov00] wird ein Verfahren vorgestellt, dass die Grauwertbilder von mehreren Kameras berücksichtigt.

Da die Anwendung keine hochauflösenden Silhouetten erfordert, wird im SIMERO-System ein Ansatz verwendet, bei dem jedes Pixel der Silhouette aus mehreren Grauwertpixeln berechnet wird. Hierzu wird das Grauwertbild in Kacheln zerteilt. Für jede Kachel werden Merkmale berechnet. Die Merkmale einer Kachel der aktuellen Szene werden mit den Merkmalen der gleichen Kachel in einem Referenzbildes ver-

gleichen. 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. Die Bildverarbeitung gliedert sich in die Module Merkmalsbilderstellung, Referenzbilderstellung, Differenzbilderstellung. Diese Module werden im folgenden vorgestellt.

2.1.1 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 SIMERO-System sind aus Gründen der Einfachheit alle Merkmale skalare Größen und die verwendete Metrik ist die absolute Differenz. Als Merkmale werden der Mittelwert, Varianz, Spannweite, Kontrast und Schwerpunkt der Grauwertpixel einer Kachel verwendet. Das Grauwertbild einer Kamera und das zugehörige Merkmalsbild, das aus fünf einzelnen Komponenten besteht, ist in der Abbildung 2 dargestellt.

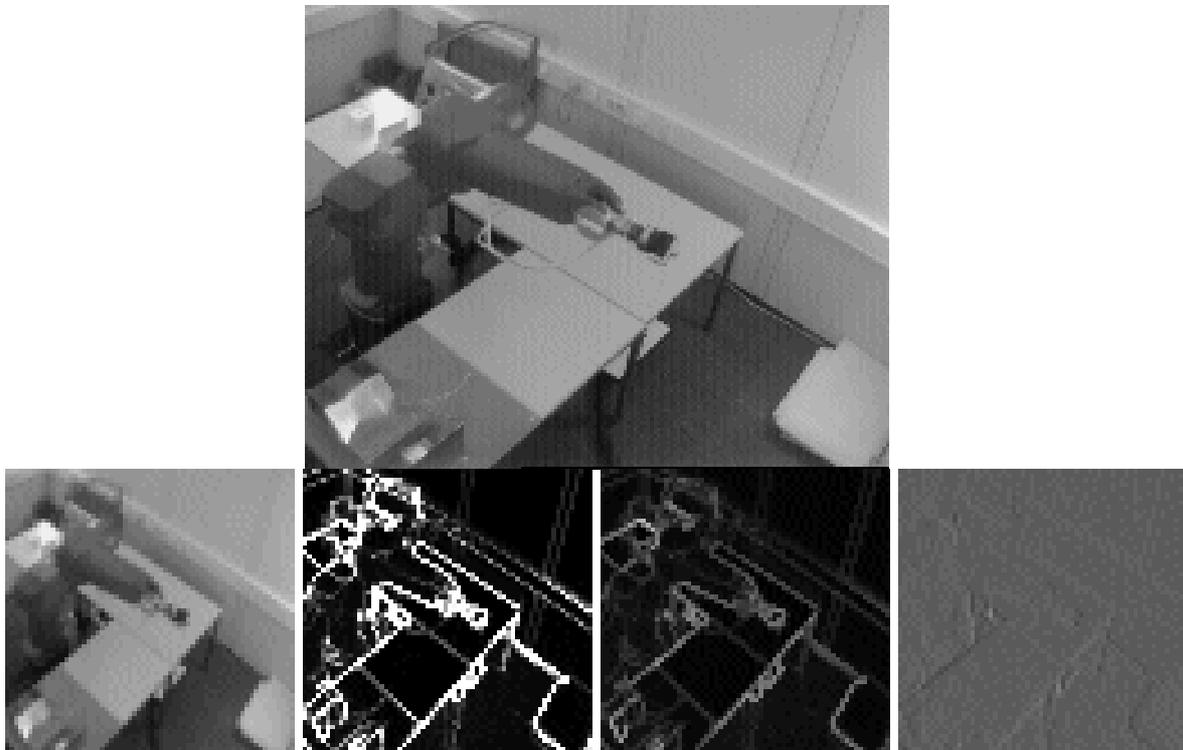


Abbildung 2: Grauwertbild (oben) und zugehöriges Merkmalsbild (unten), das aus den vier Merkmalen Mittelwert, Varianz, Spannweite und Grauwertschwerpunkt (von links nach rechts)

2.1.2 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. 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, entspricht zum Beispiel der Median dem Grauwert des Hintergrunds. Das Verfahren zeigt die Abbildung 3.

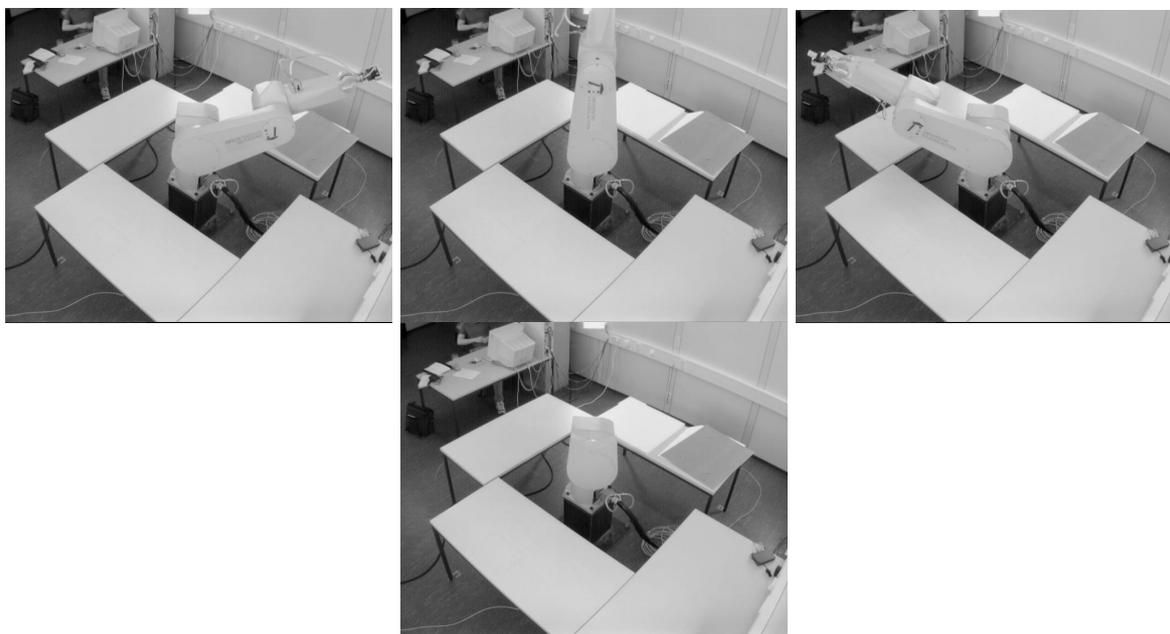


Abbildung 3: Referenzbilderstellung. Der Roboter wird verschiedene Konfigurationen gefahren und die Bilder gespeichert. Der Median aller Grauwerte einer Koordinate wird als Referenzgrauwert verwendet. Anschließend werden die Merkmale des entstandenen Grauwertreferenzbild berechnet.

Aus den Grauwertreferenzbildern für jede Kamera werden Merkmalsbilder berechnet. Nur diese Merkmalsreferenzbilder werden für die weiteren Schritte der Bildverarbeitung verwendet.

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 Sze-



nenbild 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.

Im SIMERO-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. 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.

2.1.3 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 SIMERO-System wird zur Zeit ein linearer Klassifikator verwendet. Ein statistischer Klassifikator wurde ebenfalls entwickelt. Die vom Klassifikator gelieferten Silhouetten werden durch die morphologischen Operatoren OPEN und CLOSE nachbearbeitet. Dies ist in der Abbildung 4 zu sehen.

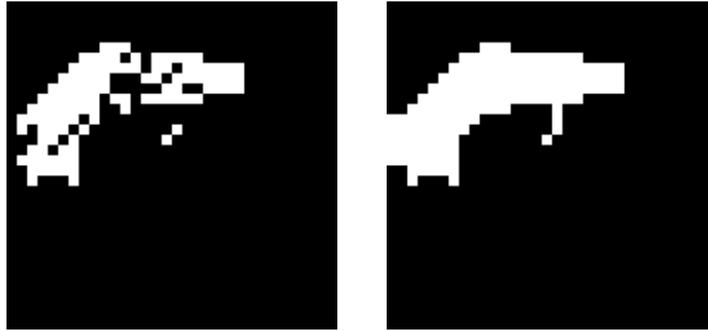


Abbildung 4: Silhouettenbild des Roboters vor und nach der Nachbearbeitung

2.2 Robotermodellierung

Die Aufgabe der Robotermodellierung besteht darin, die Kollisionserkennung mit Wissen über die Geometrie des Roboters zu versorgen. Im SIMERO-System wird eine bildbasierter Kollisionserkennung verwendet (siehe Abschnitt 2.3). Daher muss hier die Robotermodellierung zu einer angefragten Konfiguration ein Silhouettenbild des Roboters liefern. Diese Aufgabe kann auf verschiedene Arten gelöst werden. Zum einen können durch Berechnung und Rendern eines CAD-Modells des Roboters die Bilder synthetisch erzeugt werden. Zum anderen können die Bilder durch Aufnahmen und abspeichern von Kamerabildern an allen möglichen Abfragekonfigurationen erzeugt werden. Auch eine sensorgestützte Anpassung eines Modells ist möglich. Die Methoden und ihre Vor- und Nachteile werden im folgenden erläutert.

2.2.1 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 sie für einen bestimmten Robotertyp vergleichsweise schnell ist und das Roboterbild für eine beliebige Konfiguration ermitteln kann. Der Nachteil ist, dass das Modell nur für genau einen Roboter gilt und alle Modifikationen eine manuelle Anpassung des Modells erfordern.

2.2.2 Bilddatenbank

Bei der Robotermodellierung mittels Bilddatenbank wird der Roboter nacheinander in alle möglichen Abfragekonfigurationen bewegt. In jeder dieser Konfigurationen wer-



den 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. Der Nachteil ist, dass das Modell nur zu den gelernten Konfigurationen die Roboterbilder erzeugen kann und dass der Zeitaufwand zum Lernen bei einer großen Anzahl an Konfigurationen sehr hoch ist. Der Speicheraufwand der Roboterbilder ist ebenfalls relativ groß.

2.2.3 Anpassung eines Modells

Eine Art Mischform aus den beiden bisher vorgestellten Methoden ist die sensorgestützte Anpassung eines Modells. Anstatt das 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.

Im SIMERO-System wird zur Zeit eine Bilddatenbank verwendet. Eine reine CAD-Modellberechnung ist in der Entwicklung, eine sensorbasierte Modellanpassung in der Planung.

2.3 Kollisionserkennung

Die Aufgabe der Kollisionserkennung ist es, zu entscheiden, ob eine bestimmte Konfiguration zu einer Kollision mit der 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.

Grundsätzlich kann die Kollisionserkennung auf verschiedene Arten durchgeführt werden. Letztendlich muss aus den Kamerabildern eine Belegung des Arbeitsraumes abgeleitet werden. Falls der Roboter in dieser Belegung auftaucht, muss er herausgerechnet werden. Dann wird geprüft, ob der Roboter in der zu testenden Konfiguration mit einem Objekt des Arbeitsraumes kollidiert.

Den ersten Schritt, die Belegung des Arbeitsraumes zu erkennen, kann aus den Silhouettenbildern mittels einer Rückprojektion der Silhouetten in den Arbeitsraum er-



folgen. Dieses Verfahren, das auch unter den Namen „Shape from Silhouette“ oder „Volumenschnitt“ bekannt ist, wurde zum Beispiel in [Eckert00], [Noborio92] untersucht. Allerdings treten dabei Rekonstruktionsfehler auf, wie in [Niem97] untersucht wurde. Ein ähnliches Verfahren, das zur Kollisionserkennung bei einem Portalroboter verwendet wurde, ist in [Meisel94] vorgestellt worden.

Der zweite Schritt, das Herausrechnen des Roboters, erfordert entweder Wissen über das Aussehen, also ein Modell des Roboters, oder der Roboter muss anhand seines charakteristischen Aussehens ohne direkt aus dem Bild erkannt werden. Auch eine Kombination aus beiden Verfahren ist möglich.

Der dritte Schritt, der Test, ob der Roboter in seiner neuen Konfiguration mit der Umwelt kollidiert, erfordert zwingend ein Modell des Roboters.

Der im SIMERO-System verwendete Kollisionstest baut keine Repräsentation des 3D-Arbeitsraumes auf, sondern arbeitet ausschließlich auf den Silhouettenbildern von Szene und Roboter. Es gibt zwei verschiedene Verfahren, eines für unkalibrierte und eines für kalibrierte Kameras. Dieser bildbasierte Kollisionstest wird im Detail in [Ebert02] vorgestellt. Er basiert auf dem Schnitt von Szenensilhouette, aktueller und zu testender Robotersilhouette.

Die Abbildungsmatrix zur Berechnung des Ergebnisbildes ist in Tabelle 1 dargestellt. Von den acht möglichen Kombinationen sind fünf unproblematisch. Diese Pixel haben das Ergebnis „B“ für „Background“. Der Roboter kann nur dann mit einem Hindernis kollidieren, wenn er in der zu testenden Robotersilhouette ein Pixel überdeckt, das in der Szenensilhouette gesetzt ist und das nicht zum Roboter gehört. Dies entspricht in der Tabelle der 3. Zeile von unten. Problematisch sind die „Unknown“ Pixel, über die keine Aussage gemacht werden kann, da eventuell davor oder dahinter ein Hindernis sein könnte. Nur wenn die Kameras kalibriert sind, kann mit Hilfe der anderen Kameras versucht werden, diese Pixel als Hintergrund oder Hindernis zu erkennen. Die 3. und 4. Zeile von oben dürften bei korrektem Ablauf eigentlich nicht vorkommen, da der Roboter in der aktuellen Konfiguration Teil der Szene ist. Wenn der Roboter in der zu testenden Konfiguration dieses Pixel nicht überdeckt, droht keine Gefahr (Zeile 3), andernfalls aber muss eine Fehlerbedingung signalisiert werden.

Szenensilhouette	Aktuelle Robotersilhouette	Zu testende Robotersilhouette	Ergebnisbild
B(ackground)	B(ackground)	B(ackground)	B(ackground)
B(ackground)	B(ackground)	R(obot)	B(ackground)
B(ackground)	R(obot)	B(ackground)	B(ackground)
B(ackground)	R(obot)	R(obot)	E(rror)
F(oreground)	B(ackground)	B(ackground)	B(ackground)
F(oreground)	B(ackground)	R(obot)	O(bstacle)
F(oreground)	R(obot)	B(ackground)	B(ackground)
F(oreground)	R(obot)	R(obot)	U(nknown)

Tabelle 1: Abbildungsmatrix für bildbasierten Kollisionstest

Eine Konfiguration gilt dann als belegt, wenn die Anzahl der Kameras, die ein „Obstacle“-Pixel erkennen, einen festgelegten Schwellwert übersteigt. Ein Beispiel für die Anwendung des Verfahrens zeigt Abbildung 5.

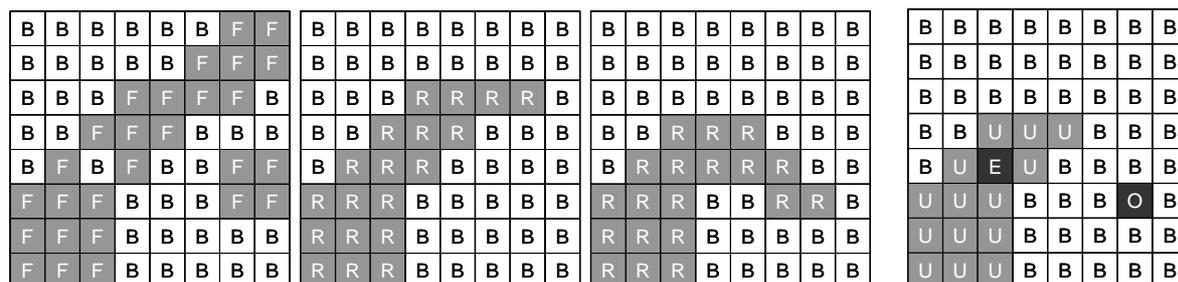


Abbildung 5: Szenensilhouette, aktuelle Robotersilhouette, zu testende Robotersilhouette und das ermittelte Ergebnisbild (von links nach rechts) (aus [Ebert02])

2.4 Bahnplanung

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. Bisher wurden jedoch hauptsächlich Probleme betrachtet, in denen der betrachtete Raum statisch und vollständig bekannt ist. Methoden zur Bahnplanung in statischer, bekannter Umgebung werden zum Beispiel in [Latombe96] und [Gupta98] vorgestellt. Methoden für dynamische Umgebungen, die allerdings ebenfalls vollständig bekannt sind, werden in [Fujimura91] untersucht.

Einige Autoren, zum Beispiel [Novak92], [Lumelsky93], [Feddema94], verwenden Roboter mit künstlichen Sensorhäuten. Andere [Yu99], [Noborio01] verwenden Sensoren, die in der Hand des Roboters montiert sind. Allen diesen Ansätzen ist gemeinsam, dass nur Teile des Arbeitsraumes erfasst werden. Dies ermöglicht eine



Kollisionsvermeidung. Für eine Bahnplanung wäre jedoch eine zeitaufwendige Exploration des Arbeitsraumes notwendig.

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.

Ausgehend von den obigen Überlegungen kann folgendes Anforderungsprofil an eine Bahnplanung für sensorbasierte Planung erstellt werden:

1. Der Planer muss terminieren und eine Fehlerbedingung melden, wenn keine Bahn gefunden wird.
2. Das System muss während der Roboterbewegung die Bahn kontinuierlich überprüfen und eine neue Bahn erzeugen, wenn dies notwendig ist.
3. Der Planer sollte eine Teilbahn liefern, wenn der Weg zum Ziel blockiert ist.
4. Der Planer sollte möglichst wenig Kollisionstest erfordern.
5. Die Bahn sollte nicht direkt an Hindernissen vorbei führen, sondern in der Mitte des freien Raumes verlaufen.
6. Die Bahn sollte so glatt und so kurz wie möglich sein.

Im SIMERO-System wird durch die stationären Kameras der gesamte Arbeitsraum des Roboters 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. Es wurden die Bahnplanungsalgorithmen BestFirst und HD-I [Noborio99], eine Randomized-Roadmap-Planer sowie ein globaler Potentialfeldplaner als Referenz realisiert. Die Planung erfolgt im Konfigurationsraum des Roboters der entsprechend der in [Henrich98] vorgestellten MaxMove-Methode diskretisiert wurde. Während die ersten drei Bedingungen, relativ einfach zu erfüllen sind, erweisen sich die Bedingungen Nr.4 und Nr.5 als 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. Dies kann er aber nur über eine große Anzahl an Kol-



lisionstest erlangen, was im Widerspruch zu Forderung Nr.4 ist. Außerdem steht die Forderung Nr.4 nach einer Bahn in der Mitte des freien Raumes auch im Widerspruch zur Forderung Nr.6 nach einer möglichst kurzen Bahn. In der Praxis wird man sich deshalb mit einem Planer begnügen, der einen Mindestabstand zu Hindernissen einhält. Dieser Mindestabstand kann zum Beispiel durch Vergrößern der Silhouettenbilder einer Szene erreicht werden (siehe 2.1.3).

3 Experimente

Die durchgeführten Experimente bestehen zum einen aus den Komponententest, bei denen die einzelnen Komponenten des Gesamtsystems auf ihre Funktions- und Leistungsfähigkeit untersucht werden. Auf diese soll im folgenden nicht weiter eingegangen werden. Eine andere Art Experiment bezieht sich auf die Leistungsfähigkeit des Gesamtsystems. Ein solches Experiment wird im folgenden vorgestellt.

3.1 Technische Realisierung

Das oben beschriebene Systemkonzept wurde realisiert. Alle Software-Schnittstellen zwischen den einzelnen Komponenten sind so ausgelegt, dass unterschiedliche Realisierungen der einzelnen Komponenten mit geringem Aufwand zusammengeschaltet werden können. Die verwendete Hardware besteht aus einem Stäubli RX130 6-Achs Industrieroboter, einer Adept CS7B Robotersteuerung, einem Standard-PC mit zwei DFG/BW1 Framegrabbern (baugleich zu ELTEC/PCEYE4), vier DMK73/C Grauwertkameras mit Vario-Focus-Objektiven. Die genauen technischen Daten sind bei [TheImagingSource02a,b] zu finden. Der Aufbau der Roboterzelle ist in der Abbildung 6 dargestellt. Die Grauwertkameras befinden sich an den oberen Ecken der Arbeitszelle. Die Software auf Seite der Steuerung ist in der proprietären Sprache V+ geschrieben, auf dem PC wird C++ unter Windows NT verwendet. Steuerung und PC sind durch ein Netzwerk (TCP/IP über Ethernet) verbunden. Das Betriebssystem auf dem PC und die Verbindung zur Robotersteuerung sind nicht echtzeitfähig. Jedoch kann dieser Aspekt bei diesem ersten Prototypen vernachlässigt werden.

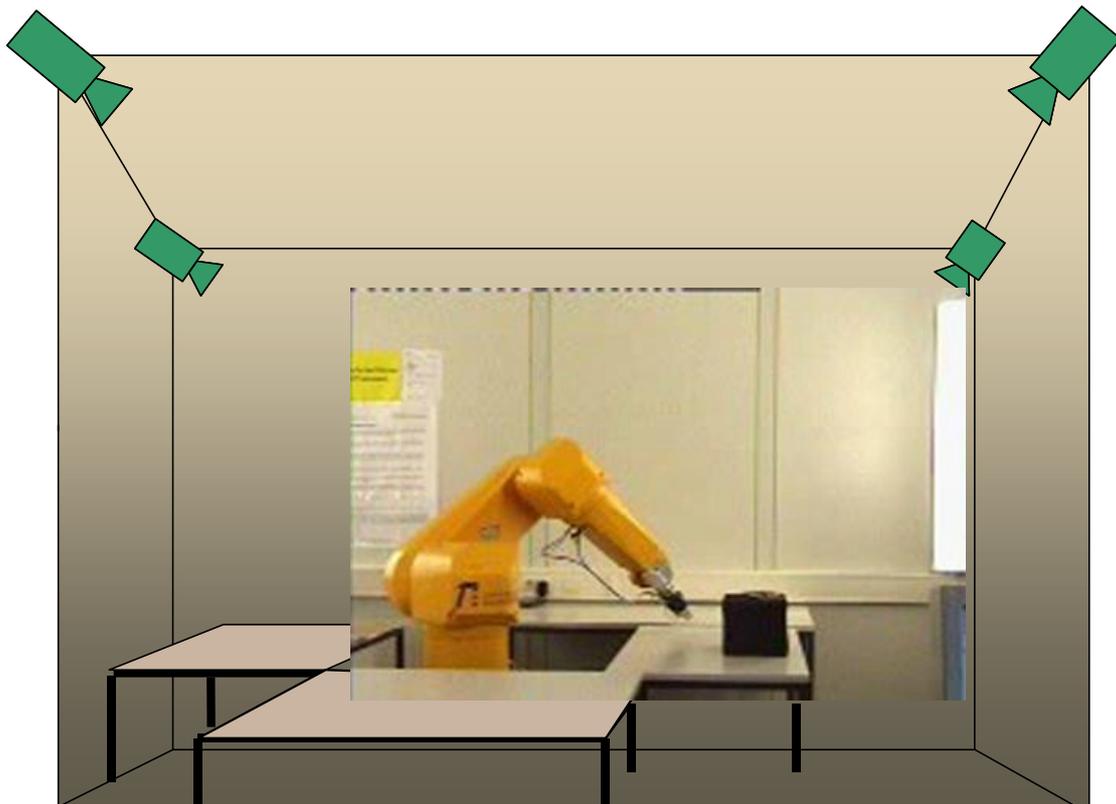


Abbildung 6: Experimenteller Roboterarbeitsraum

3.2 Versuchsdurchführung

Es wurden verschiedene Versuche mit unterschiedlichen Konfigurationsräumen und Hindernissen durchgeführt. In Anlehnung an das Anwendungsszenario hat der Roboter die Aufgabe zwischen zwei Punkten im Raum hin und her zu fahren. Ein Hindernis kommt in die Arbeitszelle in die Bahn des Roboters. In der Abbildung 7 ist dies am Beispiel eines Menschen, dessen Arm dem Roboter im Weg ist, dargestellt. Der Konfigurationsraum des Roboters ist dreidimensional und besteht aus insgesamt 1024 Konfigurationen.

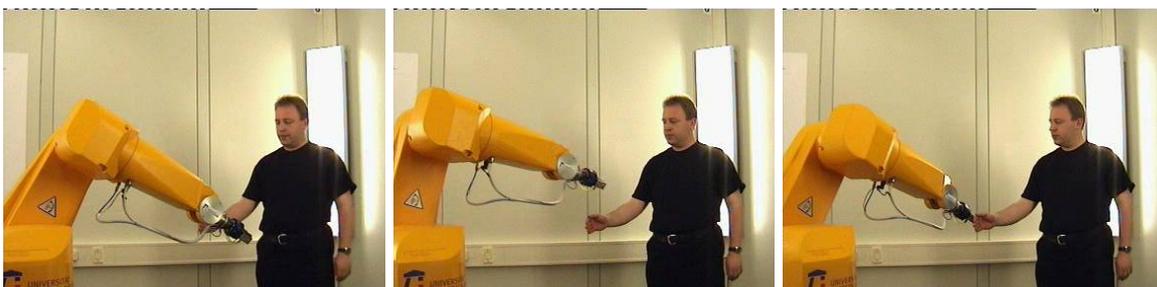


Abbildung 7: Ausweichbewegung des Roboters



3.3 Vorläufiges Ergebnis

Die Versuche zeigen, dass das erstellte Konzept im Prinzip die Anforderungen erfüllen kann, allerdings zeigen sich einige Schwächen

1. Die Bewegung des Roboters ist unterbrochen.
2. Die Bewegung des Roboters ist eckig.
3. Manchmal erreicht der Roboter das Ziel nicht, obwohl es von der Belegung des Raumes her möglich ist.
4. Der Roboter reagiert mit einer spürbaren Verzögerung auf Umweltveränderungen.

Die Ursachen der einzelnen Schwächen werden im folgenden analysiert und Gegenmaßnahmen vorgestellt.

3.3.1 Unterbrochene Bewegung

Die unterbrochene Bewegung hat ihre Ursache in der Robotermodellierung. Das aktuelle Robotermodell ist eine Bilddatenbank und kann daher nur zu bestimmten Konfigurationen Robotersilhouetten erzeugen. Dies erfordert, dass sich der Roboter zum Zeitpunkt der Bildaufnahme in exakt einer dieser Positionen befindet.

Diese Einschränkung kann auf zweierlei Weise angegangen werden. Zum einen, durch eine exakte Zeitsteuerung, bei der der Roboter ohne Halt durch eine Konfiguration fährt und im richtigen Zeitpunkt das Bild aufgenommen wird. Diese Variante ist sehr zeitkritisch und fehleranfällig. Zum anderen kann auch ein anderes Robotermodell verwendet werden, das eine Robotersilhouette für jede beliebige Konfiguration erzeugen kann. Wie im Abschnitt 2.2 beschrieben ist, kann dies zum Beispiel durch eine Berechnung des Aussehens des Roboters aus den Gelenkwinkeln mit Hilfe von approximierenden Hüllkörpern geschehen. Dieser Ansatz wird zur Zeit weiterverfolgt.

3.3.2 Eckige Bewegung

Die eckige Bewegung hat ihre Ursache in der Robotermodellierung und Bahnplanung. Der Konfigurationsraum des Roboters ist diskretisiert und der Roboter bewegt sich immer von einem Gitterpunkt zum nächsten. Eine feinere Diskretisierung verbessert dieses Verhalten, hat aber zur Folge, dass die Bahnplanung aufwendiger wird und falls eine Bilddatenbank als Robotermodell verwendet wird, die Anzahl der zu speichernden Roboterbilder steigt. Eine überschleifende Bewegung des Roboters zum Glätten der Bahn ist möglich, wenn die Kollisionserkennung sicherstellen kann, dass der Roboter auch bei leichter Abweichung von der Sollbahn nicht mit einem



Hindernis kollidiert und wenn das Robotermodell ein Herausrechnen des Roboters an jeder beliebigen Konfiguration zulässt.

3.3.3 Fehlerhafte Rekonstruktion

Wenn mehrere Hindernisse im Arbeitsraum sind und eines davon nahe am Roboter ist, ist der rekonstruierte Konfigurationsraum zum Teil so stark durch Hindernisse belegt, dass das Ziel nicht mehr erreicht werden kann, obwohl dies von der tatsächlichen Belegung des Raumes her noch möglich ist. Grundsätzlich kann diesem Problem immer durch hinzufügen von weiteren Kameras begegnet werden. Durch eine Simulation ist jedoch auch möglich bei einer gegebenen Anzahl von Kameras, einem gegebenen Arbeitsraum und einem Robotermodell, diejenige Positionierung der Kameras zu ermitteln, bei der die Rekonstruktion im Mittel möglichst gut wird.

3.3.4 Beschleunigung des Systems

Nach dem Amdahlschen Gesetz ist es am sinnvollsten den aufwendigsten Teil des Systems zu beschleunigen. Der Hauptaufwand liegt zur Zeit in der Bildverarbeitung. Dies wird durch eine Zeitnahme der in den verschiedenen Modulen verbrauchten Prozessorzeit bestätigt. Eine Beschleunigung des Gesamtsystems sollte also an diesem Punkt ansetzen. Durch die kachelbasierte Verarbeitung, bei der zur Erstellung eines Merkmalspixels nur die Bilddaten der entsprechenden Grauwertbilder verwendet werden, ist die Bildverarbeitung mit Ausnahme der Differenzbildverbesserung hochgradig parallelisierbar. Im Extremfall kann für jede einzelne Kachel eine eigene Verarbeitungseinheit verwendet werden.

4 Zusammenfassung

Die für einen Menschen im Arbeitsraum gefährlichste Bewegung ist die freie Transferbewegung. Daher wird im SIMERO-System diese Transferbewegung abgesichert.

Das System gliedert sich in die vier Hauptkomponenten Bildverarbeitung, Robotermodellierung, Kollisionserkennung und Bahnplanung. Aufgabe der Bildverarbeitung ist es, ein Silhouettenbild der aktuellen Szene zu erstellen. Die Bildverarbeitung gliedert sich ihrerseits in die Module Merkmalsbilderstellung, Referenzbilderstellung und Differenzbilderstellung. Die Aufgabe der Robotermodellierung besteht darin, die Kollisionserkennung mit Wissen über die Geometrie des Roboters zu versorgen. Zur Zeit wird eine Bilddatenbank verwendet. 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. Im SIMERO-System wird eine bildbasierter Kollisionserkennung verwendet. Die Bahnplanung hat die Aufgabe,



ausgehend von einer gegebenen Startkonfiguration eine kollisionsfreie Bahn zu einer gegebenen Zielkonfiguration zu finden. Es werden zur Zeit verschiedene Bahnplaner auf ihre Leistungsfähigkeit unter verschiedenen Randbedingungen untersucht.

Die Versuche zur Abschätzung der Leistungsfähigkeit zeigen, dass der Ansatz prinzipiell geeignet ist, die gestellte Aufgabe zu lösen. Die festgestellten Schwächen werden durch die vorgestellten Ansätze angegangen. Weitere Verbesserungen betreffen die Vergrößerung des Arbeitsraumes und eine Verfeinerung der Diskretisierung des Konfigurationsraumes.

5 Literatur

- [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, Baden-Baden, Germany, August 20 - 22, 2001
- [Ebert02] Ebert, D., Henrich, D. „Safe Human-Robot-Cooperation: Image-based Collision Detection for Industrial Robots“. Accepted for: IEEE Conference on Intelligent Robots and Systems, IROS 2002, Lausanne, September 30th – October 5th, 2002
- [Eckert00] Gerald E. “Automatic Shape Reconstruction of Rigid 3-D Objects from Multiple Calibrated Images”, Eusipco 2000 Proceedings, Tampere, Finland, 2000
- [Feddema94] Feddema, J.T.; Novak, J.L.: “Whole Arm Obstacle Avoidance for Teleoperated Robots”, IEEE Robotics and Automation Proceedings, pp.3303 – 3309, 1994
- [Fujimura91] Fujimura, Kikuo: “Motion Planning in Dynamic Environments”, Springer Verlag, Tokyo, Berlin, New York, 1991
- [Gupta98] Kamal Gupta, Angel del Pobil (ed.): “Practical Motion Planning in Robotics; Current Approaches and Future Directions”, John Wiley & Sons Ltd, Chichester, England, 1998
- [Henrich98] Henrich D; Wurll Ch; Wörn H; “On-line path planning with optimal C-space discretisation”, In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98), Victoria, Canada, Oct. 12-16, 1998
- [Horprasert00] Horprasert T., Harwood D., and Davis L.S. “A Robust Background Subtraction and Shadow Detection” Proc. ACCV'2000 Taipei, Taiwan, January 2000



- [ISO10218] ISO 10218, EN 775: "Manipulating industrial robots – Safety", (ISO 10218 modified), 1992
- [Ivanov00] Ivanov Y., Bobick A., Liu J., "Fast Lighting Independent Background Subtraction", International Journal of Computer Vision 37(2): 199-207, June 2000
- [Latombe96] Latombe J-C : "Robot motion planning"; 4. print.; Boston : Kluwer Acad. Publ., 1996.
- [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
- [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] Hiroshi Noborio and Yasushi Katahira, "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
- [Noborio99] Hiroshi Noborio, Yasuhiro Maeda, 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] Hiroshi Noborio, Yutaka Nishino: "Image-based Path-Planning Algorithm on the Joint Space" IEEE International Conference on Robotics and Automation, pp. 1180-1187, Seoul, 2001
- [Novak92] 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
- [Rosin95] Rosin P.L and 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



[TheImagingSource02a]

<http://www.theimagingsource.com/prod/cam/dmk73/dmk73.htm>

[TheImagingSource02b]

<http://www.theimagingsource.com/prod/grab/dfgbw1/dfgbw1.htm>

[Yu99] Yong Yu, Kamal Gupta „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