DISSERTATION

Szenenanalyse in unstruktuierter Umgebung

Adaptives Verfahren zur Objekterkennung basierend auf der Multi-Sensor-Fusion und aktiven Wahrnehmung

zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat)
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik
der Universität Hamburg

vorgelegt von

Denis Klimentjew

Hamburg, 2015

Genehmigt von der MIN-Fakultät, Department Informatik
der Universität Hamburg auf Antrag von

Prof. Dr. Jianwei Zhang (Erstgutachter)
Department Informatik
Universität Hamburg

Prof. Dr. Joachim Hertzberg (Zweitgutachter)
Institut für Informatik
Universität Osnabrück

Disputation: Hamburg, 06.10.2015

„Das Ganze ist mehr als die Summe seiner Teile“

Aristoteles (384 - 322 v. Chr.)

Abstract

The present thesis tries to change the existing paradigm related to object recognition. Most state-of-the-art publications use a small number of detectors, usually only one. Consequently, the developers work almost exclusively to achieve the improvement of one single detector.

The work presented here takes a different approach. At first the data of multiple, heterogeneous sensors is used. With the help of multi sensor fusion the data is combined and registered in relation to each other. The final result of this step is the 3-D colored point cloud. This point cloud constitutes the input for the used detectors, with the number of detectors varying. The goal of the presented thesis is the combination of the information in such a way that the used detectors can operate on most or all object properties. The detectors results are weighted depending on the quality of the detector and fused to one final result. The advantage of this approach is that this decision is based on more information than only the sum of results from each single detector. Another innovation of the presented work is the inclusion of the service robotics aspect. It integrates two new active components and places a mobile robot platform with grasping capabilities in the center of this thesis. This way, sensors can be driven to a scene, allowing for active sensing, or the point of view on the scene can be changed and/or adapted. The second active component is the robot manipulator which gets the possibility to interact with the objects inside the scene. One of the biggest advantages is the possibility to reduce or completely solve the problem with partial / total occlusion.

The essence of the presented work is an object recognition pipeline based on a robot system with the ability to recognize and analyze the environment. The platform automatically navigates to a given position in the space. The orientation towards the scene can be changed. If the region of interest is reached, the scene can be perceived and the objects inside recognized. If this fails, the perspective of the scene can be changed several times. If this is unsuccessful as well, the robot can manipulate the scene and remove detected or recognized objects from it. The scene is analyzed after each single step. The result is a continuously repeated, closed cycle which includes the perception via sensors, analysis, changing of the sensor position as well as the intervention with the scene.

Should any detected objects fail to be recognized, the sensor information can be used to generate a colored 3-D model from the detected features. This model can be sent to the operator to complete the information and classify the unrecognized objects.

Zusammenfassung

Die vorliegende Dissertation versucht, einen Paradigmenwechsel in Bezug auf die Objekterkennung zu vollziehen. Die meisten vergleichbaren Arbeiten greifen auf eine geringere Anzahl an Detektoren zurück – meistens nur auf einen. Dabei wird, basierend auf der „Güte“des jeweiligen Detektors, die bestmögliche Erkennungsrate erzielt. Somit sind viele Wissenschaftler damit beschäftigt, die Performanz des eingesetzten Detektors zu optimieren. Diese Arbeit geht einen anderen Weg. Dabei werden die Daten mehrerer heterogener Sensoren verwendet. Diese werden unter Zuhilfenahme der Multi-Sensor-Fusion in Bezug zueinander registriert, was grundsätzlich eine einheitliche Auswertung aller zur Verfügung stehenden Daten ermöglicht. Das Ziel ist es, die Daten so zu kombinieren, dass nach möglichst vielen Objekteigenschaften gesucht werden kann.

Auf diese kolorierte 3-D-Punktwolke werden mehrere heterogene Detektoren angewandt. Die Ergebnisse der Detektoren werden anhand ihrer qualitativen Performanz gewichtet und anschließend zu einem Ergebnis fusioniert. Eine weitere Neuerung stellt der Aspekt der Servicerobotik dar. Dieser bringt zwei aktive Komponenten mit sich und stellt eine fahrbare Roboterplattform mit manipulatorischen Fähigkeiten in den Mittelpunkt der vorliegenden Dissertation. Dadurch können die Sensoren einerseits zu einer Szene gefahren werden, was aktive Wahrnehmung ermöglicht, da der Blickwinkel auf die Szene geändert oder angepasst werden kann. Andererseits kann ein Roboter aktiv in eine Szene eingreifen und die beteiligten Objekte manipulieren, wodurch auch das Problem der Verdeckung zumindest entschärft werden kann.

Wird das gesamte Vorhaben der vorliegenden Arbeit zusammengefasst, entsteht ein Erkennungssystem, das auf einer Roboterplattform basiert und in der Lage ist, seine Umgebung wahrzunehmen und zu analysieren. Dabei navigiert der Roboter selbstständig zur bestimmten Position im Raum. Die Szene wird analysiert, die beteiligten Objekte werden erkannt. Schlägt dies fehl, kann die Perspektive einmal oder mehrmals geändert werden. Bringt auch das keinen Erfolg, wird in die Szene aktiv eingegriffen, und erkannte oder detektierte Objekte werden daraus entfernt. Nach jedem Schritt wird die Szene erneut analysiert. Es entsteht ein geschlossener Zyklus, der die Wahrnehmung über Sensoren, die Analyse, die Änderung der Position sowie die Manipulation einer Szene beinhaltet und ständig durchlaufen wird. Sind am Ende dennoch einige Objekte nicht erkannt, kann aus den Daten der Detektion ein koloriertes 3-D-Objektmodell erstellt und zwecks Klassifikation und Ergänzung weiterer Eigenschaften an den Operator geschickt werden.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

List of Algorithms

Kapitel 1

Einführung und Motivation

Dieses Kapitel soll den Leser in die Thematik der vorliegenden Arbeit einführen, sowie die Vorgehensweise verdeutlichen und motivieren. Dabei ist die Frage zu beantworten, warum das gewählte Thema interessant und aktuell ist. Des Weiteren werden die grundlegende Idee der vorliegenden Dissertation - sowie deren Innovation und wissenschaftliche Bedeutung - ausführlich erläutert. Außerdem wird der Stand der Forschung in allen relevanten Bereichen präsentiert, sowie die Parallelen und Widersprüche zu dieser Arbeit werden diskutiert.

Obwohl die vorliegende Dissertation in einem eher technischen Bereich entstand, hat sie sowohl Software- wie auch Hardwarekomponenten zum Gegenstand. Es ist ein Versuch, ein mobiles System und die darauf laufende Software noch enger an einander zu koppeln. Dies steht im Einklang mit unserem Verständnis der heutigen Robotik. Eine Plattform ist aus unserer Sicht ein riesiges Evaluationssystem, in dem das Wissen aus verschiedenen Bereichen, angefangen bei der Elektrotechnik und der Hardware bis hin zur Wissensgewinnung und Verarbeitung, zusammenkommt, integriert, getestet, erweitert und verbessert wird. Erst hier werden viele Probleme sichtbar und nur hier kann ein komplettes System evaluiert und unter realen Bedingungen getestet werden.

1.1. Motivation

Systeme zu entwickeln, die automatisch die Umwelt erkennen, in ihr sicher navigieren und sie über ihre Grenzen hinaus selbstständig entdecken, ist immer noch eine große Herausforderung. Im Bereich der Robotik ist die Wahrnehmung, die Kombination aus sensorischen Informationen und Methoden zum Erkennen und Klassifizieren von Objekten, von besonderer Bedeutung, eine grundlegende und notwendige Komponente jedes Robotersystems. Gegenstand dieser Arbeit ist die Entwicklung eines solchen Systems. Dabei wird auf bekannte Methoden zurückgegriffen, neue Algorithmen werden entwickelt

und evaluiert.

Die dieser Arbeit zugrunde liegende Idee ist die Integration einer Roboterplattform in die aktive Objekterkennung. Ein Roboter ist nicht bloß ein Transportmittel für die Sensoren und Aktuatoren, sondern er ist komplett über die Interaktion mit der Umgebung in diese integriert und an der Szenenanalyse aktiv beteiligt. Die Umwelt wird dabei über die Sensoren wahrgenommen und analysiert. Der Roboter eröffnet somit die Möglichkeit, gezielt in die Szenenanalyse einzugreifen, angefangen bei der möglichen Änderung der Position und damit der Perspektive bis hin zur Manipulation innerhalb der Szene. Die Abbildung 1.1 visualisiert dabei alle für die Objekterkennung relevanten wissenschaftlichen Bereiche.

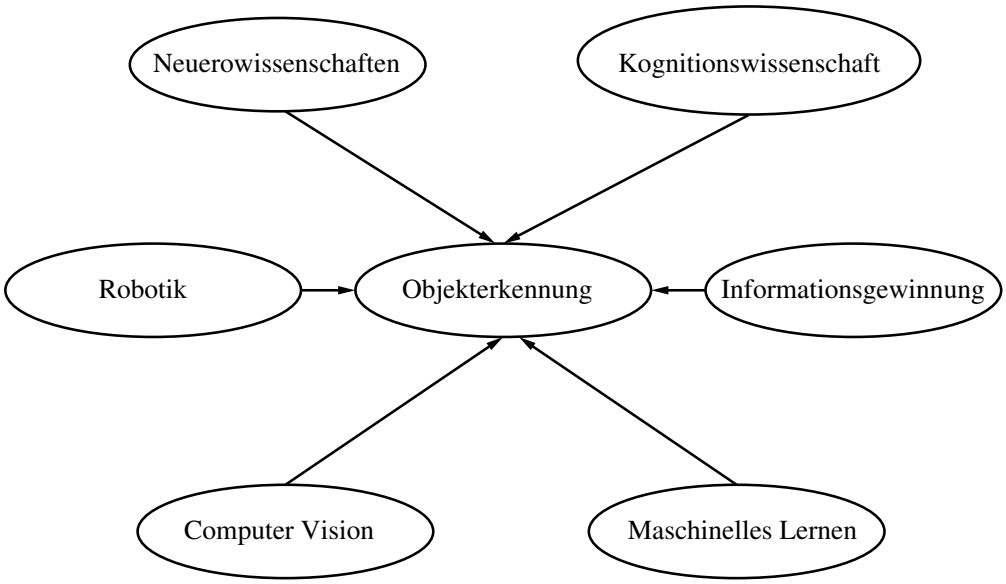


Abbildung 1.1.: Überblick über die, für die Objekterkennung relevanten Forschungsbereiche.

Die Notwendigkeit der Objekterkennung für mobile autonome Systeme ist unumstritten. Dennoch gleich einen Roboter als aktive Komponente des Erkennungsprozesses einsetzen? Viele unterschiedliche Aspekte sprechen dafür, so bringen die rasante Entwicklung und fallende Preise die Roboter als Massenprodukt durchaus in Betracht. Erste Anzeichen sind bereits erkennbar, so sind staubsaugende Roboter in vielen Haushalten bereits vorhanden, sie genießen mittlerweile eine allgemeine Akzeptanz. Auf der anderen Seite wird die Menschheit immer älter. Werden die Ereignisse kombiniert, öffnet die Verwendung und Integration der Roboter in den menschlichen Alltag interessante und vielversprechende Möglichkeiten. Um Sicherheit und Zuverlässigkeit zu garantieren, bedarf es einer verlässlichen Objekterkennung. Daraus ergibt sich folgende Fragestellung,

mit deren Untersuchung sich die vorliegende Dissertation beschäftigt:

Kann die Objekterkennung verbessert und zuverlässiger gemacht werden? In wie weit ist die Objekterkennung in komplexen Szenen mit einer partiellen oder totalen Verdeckung möglich? Wie viel können die Roboter erreichen? Wo sind die Grenzen und wodurch sind diese bedingt?

Die Analyse von komplexen Szenen und die Objekterkennung unter dem Aspekt der partiellen Verdeckung, ist ein seit 20 Jahren immer noch nicht effizient gelöstes Problem [KF07]. Die vorliegende Arbeit wird die beschriebene Problematik der Objekterkennung sicherlich nicht vollständig lösen können, sie trägt aber dazu bei, indem das Problem analysiert und auf unterschiedlichen Ebenen Neuerungen und Verbesserungen integriert und getestet werden. Im klassischen Sinn besteht ein Erkennungssystem nach [JKS95] mindestens aus einem Merkmaldetektor, einer Hypothesenbildung und Verifikation sowie einer Modelldatenbank. Abbildung 1.2 visualisiert die einzelnen Komponenten sowie die Art und Weise, in der die eingehende Information verarbeitet wird.

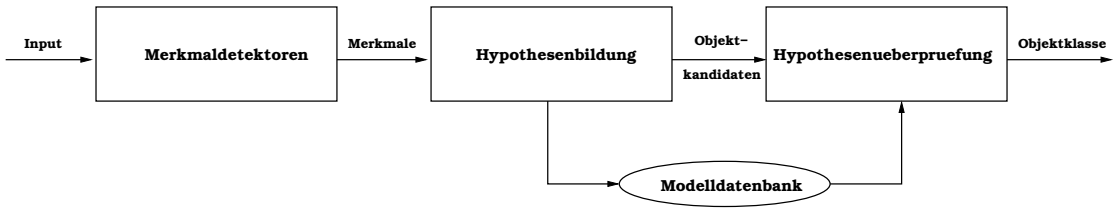


Abbildung 1.2.: Komponenten eines typischen Erkennungssystems.

Dabei werden die Eingangsdaten, Bild oder Punktwolke, nach spezifischen Kriterien abhängig von dem verwendeten Detektor abgetastet. Die extrahierten Merkmale werden mit den bekannten Merkmalen, den in der Datenbank befindlichen Objekten, verglichen. Anhand dieses Vergleichs werden eine oder mehrere Hypothesen gebildet und verifiziert. Ausgegeben wird meistens eine Objektklasse sowie die Wahrscheinlichkeit, mit der das getestete Objekt zu dieser Klasse gehört. Solche Systeme können je nach Bedarf beliebig erweitert werden.

Stehen mehrere unterschiedliche Sensorarten zur Verfügung, kann ein Erkennungssystem durch die Verwendung mehrerer Detektoren verbessert werden. Dies kann zusätzlich durch die untereinander registrierten Sensordaten optimiert werden. Sind die Sensoren zueinander kalibriert, entsteht eine fusionierte Datenstruktur, die für einen beliebigen Punkt der Umgebung alle vorhandenen Informationen bereitstellt, wie zum Beispiel Farb-, Tiefen- und Temperaturinformationen. Diese Struktur erlaubt mehr, als nur die Auswertung einzelner Sensordaten. So können beispielsweise die ermittelten Kanten durch den Abgleich mit der Tiefenkarte in einem 2-D-Bild verifiziert werden.

Des Weiteren können die Werte einzelner Detektoren zu einem gewichteten Ergebnis

zusammengefasst werden. Die Gewichtung ist notwendig, da die Qualität verschiedener Detektoren unterschiedlich ist. So ist zum Beispiel die Wahrscheinlichkeit für die richtige Einordnung eines detektierten Objekts über den SIFT/SURF-Detektor deutlich höher als über die Farbsegmentierung. Kann ein Objekt auch danach nicht eindeutig einer Klasse zugeordnet werden, können durch ein kostenbasiertes Regelsystem weitere Maßnahmen eingeleitet werden. So kann die Perspektive auf eine Szene durch die Bewegung einzelner Roboterteile oder des kompletten Systems verändert werden. Neue Daten werden akquiriert und ausgewertet, zusätzlich können die bereits aus dem gescheiterten Versuch vorhandenen Daten für die Entscheidung herangezogen werden. Schlägt auch das fehl, kann ein Roboter mithilfe seines Manipulators die Anordnung der vorliegenden Szene verändern. Dieser Schritt ist besonders in Verbindung mit einer vorhandenen partiellen oder totalen Verdeckung interessant, dies wird im Rahmen der vorliegenden Dissertation ansatzweise realisiert und getestet.

Aus globaler Sicht entsteht ein aktives, autonomes, roboterbasiertes Erkennungssystem, das viel Potenzial für die Objekterkennung bietet und sogar auf Szenen mit partieller oder globaler Verdeckung angewandt werden kann. Im nächsten Abschnitt werden die Vorgehensweisen sowie einige der möglichen Szenarien vorgestellt, die das Dissertationsvorhaben weitergehend motivieren und verdeutlichen sollen.

1.2. Vorgehensweise und mögliche Szenarien

Ein Roboter navigiert in einer bekannten/unbekannten Umgebung ausgehend von den 3-D-Punktwolken, die er über eigene Sensoren wahrnimmt. Im Fall einer bekannten Umgebung liegt eine statische Karte vor, die generisch ergänzt wird (engl. „dynamical mapping “). Befindet sich der Roboter in einer unbekannten Umgebung, wird diese unter Zuhilfenahme eines Explorationsalgorithmus erforscht und kartiert (SLAM), dadurch kann eine spätere Navigation zu jedem beliebigen Punkt in der Karte gewährleistet werden. Außerdem garantiert der Explorationsalgorithmus eine komplette Erfassung der Umgebung. Das so erfasste Umfeld wird zuerst vorsegmentiert und in folgende vier Kategorien unterteilt:

- Wände,
- Boden,
- Decke,
- parallel zum Boden verlaufende planare Oberflächen.

Die segmentierten Punkte werden in Form von planaren Flächen zusammengefasst, um die Speicher-, Darstellungs- und Recheneffizienz zu steigern. Dabei stellen die parallel zum Boden ausgerichteten planaren Oberflächen (ROIs, engl. „Region Of Interest“)

dar, da auf solchen Oberflächen sich mit höherer Wahrscheinlichkeit Objekte befinden können. Die Positionen solcher Regionen werden in der Karte notiert.

Zur Evaluierung wird die Umgebung auf die Innenräume begrenzt. Die meisten vorgestellten Evaluationsszenarien greifen die Szenarien des RACE-Projektes auf und benutzen die dort verwendeten Objekte. Grundsätzlich ist aber das resultierende System umgebungsunabhängig und kann überall eingesetzt werden.

Soll jetzt nach den bestimmten Objekten gesucht werden, wird der Roboter nacheinander zur den markierten ROIs navigiert. Ein erreichter oder zu untersuchender ROI wird unter Zuhilfenahme unterschiedlicher Sensoren mit möglichst großer Punktdichte abgetastet. Dabei werden die Informationen unterschiedlicher Sensoren in Abhängigkeit von eigenen charakteristischen Merkmalen und Eigenschaften der Szene intelligent zusammengefasst. Das Ziel der oben erwähnten intelligenten Multi-Sensor-Fusion ist, mehr Informationen aus den vorliegenden Sensordaten zu gewinnen als die Summe einzelner Sensorwerte. Es entsteht eine partiell kolorierte Punktwolke. Ähnlich wird auch mit zu analysierenden Szenen verfahren, diese werden mit unterschiedlichen Sensorarten und mit großer Punktdichte wahrgenommen, die Information wird fusioniert und ausgewertet.

In der vorliegenden Arbeit wird die Punktwolke grundsätzlich als ein Bild betrachtet, was die Segmentierung in Vordergrund, Hintergrund und Rauschen erlaubt. Wände, Decke, Boden und die planaren Oberflächen, die bei der Segmentierung herausgebildet worden sind, werden bei der Objekterkennung als Hintergrund verstanden und entfernt. Die Punktwolke wird durch diesen Prozess in mehrere Bereiche unterteilt (engl. „clustering“).

Auf die verbleibenden Voxeldichten (ein Voxel ist die Erweiterung des Pixels um eine dritte Dimension) werden parallel mehrere Algorithmen (Detektoren) angewandt, wie zum Beispiel Farbsegmentierung, Kantendetektion, Form- und Texturdetektoren etc. Dabei werden diese Bereiche weiter unterteilt. Bei diesen Bereichen kann es sich um einzelne Objekte handeln, oder ein Bereich kann untersegmentiert sein und mehr als ein Objekt beinhalten. Ein weiteres Problem stellt die Verdeckung dar. In der vorliegenden Arbeit gehen wir von der aus der 3-D-Bildverarbeitung übernommenen Definition der Verdeckung aus. Danach ist die Verdeckung ein Effekt, bei dem ein Objekt die Sicht auf ein anderes Objekt komplett (total) oder teilweise (partiell) blockiert. Um den beiden Problemen gerecht zu werden, wird ein an dem Roboter montierter 7-DOF-Manipulator verwendet. Nach der 3-D-Rekonstruktion wird versucht, über einen modellbasierten Vergleich die Objekte zu erkennen. Scheitert dies, handelt es sich entweder um Objekte (Untersegmentierung) oder um ein unbekanntes Objekt. Hier existieren grundsätzlich zwei mögliche Ansätze, die vielversprechend sind und im Rahmen dieser Dissertation betrachtet werden. Der erste Ansatz ist die Änderung der Perspektive und damit verbunden die Gewinnung zusätzlicher Informationen. Dabei wird gehofft, dass aus dem veränderten Blickwinkel der Abstand zwischen den beteiligten Objekten so groß wird, dass diese von einander separiert werden können.

Der zweite Ansatz ist die gezielte Manipulation, dabei können die Objekte über die Oberfläche bewegt und/oder gegriffen werden. Durch die Reibung besteht sogar die Möglichkeit, auf die Materialeigenschaften zurück zu schließen. Falls die zu untersuchende Punktwolke mehrere Objekte beinhaltet, wird die Objektanordnung bei der Manipulation gestört. Somit können eventuell durch das erneute Anwenden der Detektoren einzelne Objekte erfolgreich separiert werden. Nach einer weiteren Segmentierung werden die separierten Voxels erneut rekonstruiert. Mit den rekonstruierten Modellen wird versucht, die Objekte zu erkennen. Scheitern alle Versuche, wird davon ausgegangen, dass das Objekt unbekannt ist. Das rekonstruierte Modell kann dann in der Datenbank abgespeichert werden, und/oder es kann eine Nachricht an einen Operator bezüglich der Erkennung und/oder weiterer Kontextinformationen gesendet werden. Das Dissertationsvorhaben ist in der Abbildung 1.3 grafisch dargestellt und wurde in [KRZ12] veröffentlicht.

Die Anwendungsszenarien sind dementsprechend vielseitig und kaum einzugrenzen. Im Weiteren werden nur zwei mögliche Szenarien kurz skizziert, da sie einerseits sehr unterschiedlich sind, andererseits wird die Wichtigkeit und Aktualität der beiden durch zwei europäische Projekte (HANDLE[1] und RACE[2]) bestätigt.

Das erste beschäftigt sich mit der Objektmanipulation. Dabei steht primär die Manipulation eines Objekts in der Hand im Mittelpunkt. Um die notwendigen Manipulationen planen zu können, werden die Position und Orientierung des Objekts im Raum benötigt (6-DOF; engl. „degree of freedom“). Da die beiden Größen sich temporär dynamisch verändern, sollen Position und Orientierung möglichst oft neu bestimmt werden.

Ein weiteres Szenario, das an das RACE-Projekt angelehnt ist und als Hauptszenario angesehen wird, ist der Einsatz des Service-Roboters zur Bedienung von Gästen. Dabei sollen mehrere Gerichte sowie Getränke an Personen serviert werden. Auch hier sind die Informationen über Klassifikation, Position und Orientierung aller beteiligten Objekte im Raum notwendig. Außerdem wird schnell deutlich, dass die Kenntnisse über Objekteigenschaften, wie Form, Beschaffenheit, Schwerpunkt etc., die gestellten Aufgaben erst realisierbar machen. Daraus ergibt sich nicht nur die Segmentierung, sondern auch die Erkennung vorliegender Objekte als äußerst relevant.

Zusammengefasst wird im Rahmen der vorliegenden Dissertation ein System entwickelt, das die Objekte autonom erkennt und damit selbstbestimmte und gespeicherte Kontextinformationen bereitstellt, eine notwendige Basis für die Manipulation und den aktiven Eingriff des Roboters in seine Umwelt. Die Schnelligkeit und Genauigkeit der Umgebungsanalyse bilden notwendige Kriterien, damit der Einsatz des Roboters im menschlichen Umfeld möglich und erfolgreich wird.

[1] http://www.handle-project.eu/
[2] http://www.project-race.eu/

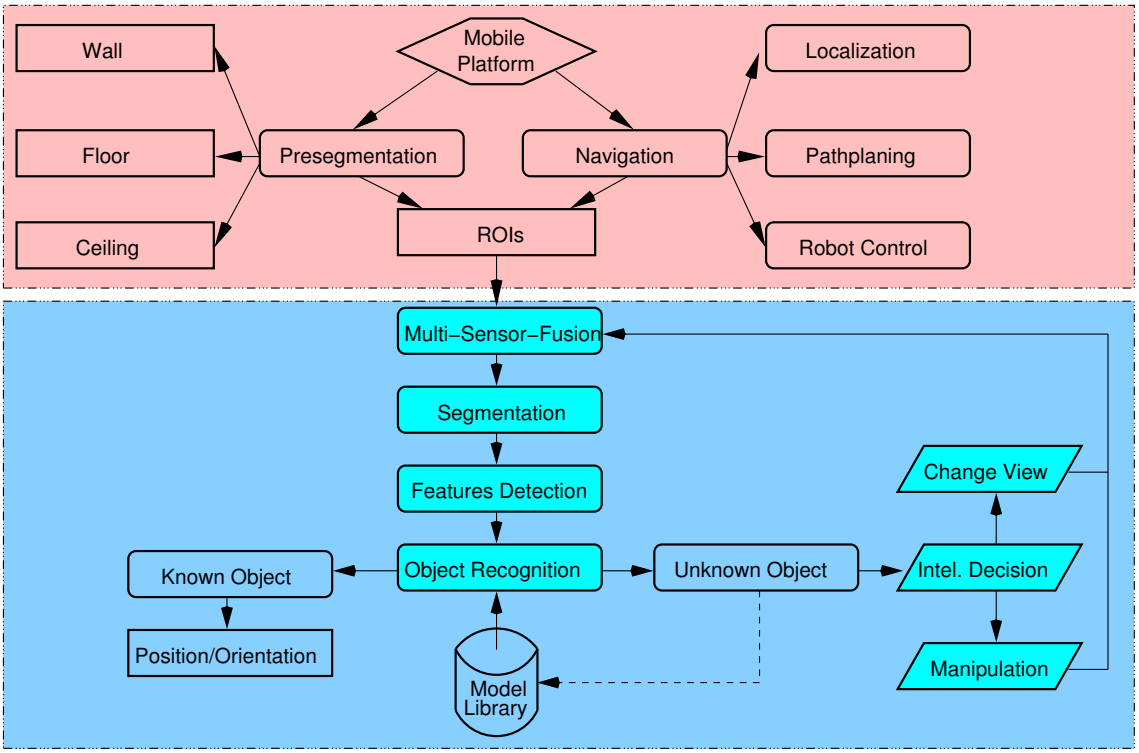


Abbildung 1.3.: Grafische Visualisierung des Dissertationsvorhabens. Das untere, durch eine gestrichelte Linie hervorgehobene und blau unterlegte Rechteck stellt den Kern des Dissertationsvorhabens dar. Das obere Rechteck wird als notwendige und zugrunde liegende Basis betrachtet, deren Inhalt umfänglich erforscht ist und die daher kaum Möglichkeiten zur Gewinnung neuer Erkenntnisse bietet.

1.3. Stand der Technik

Das Thema dieser Arbeit ist komplex, umfasst mehrere Bereiche und ist in verschiedenen Gebieten unterschiedlich stark erforscht worden. Den Kern der vorliegenden Dissertation bilden zwei aufeinander aufbauende Bereiche. Der erste und wichtigste ist die aktive Teilnahme eines Roboters am Erkennungs- und Analyseprozess. Der zweite ist die gezielte und gewichtete Analyse der Ergebnisse unterschiedlicher Detektoren, die auf den fusionierten Daten aufbauend, nach bestimmten Merkmalen suchen. Der Autor verwendet ein utilitaristisches Modell mit vorhandener Unsicherheit. Die beiden Aspekte greifen dabei ineinander. So wird die Szene aus einer Perspektive erfasst, und die fusionierten Daten werden analysiert. Werden nicht alle Objekte erkannt, wird auf der Basis der

gewichteten Analyse unter Zuhilfenahme eines regelbasierten Systems eine Aktion eingeleitet. Dabei stellt der Roboter eine aktive Komponente dar, seine Ressourcen werden nach Gewichtung und Erfolgsversprechen eingesetzt. Die Anwendung ist abgeschlossen, wenn alle Objekte eindeutig erkannt worden sind. Falls alle zur Verfügung stehenden Mittel erschöpft sind, werden das rekonstruierte Modell und weitere gewonnene Daten des unbekannten Objekts in der Datenbank gespeichert. Später können so zum Beispiel durch einen Supervisor die Daten entweder verworfen oder mit weiteren Eigenschaften innerhalb der Datenbank ergänzt werden.

Aufgrund der oben genannten Komplexität wurde entschieden, den Stand der Technik in unterschiedliche relevante Bereiche aufzuteilen und relevante Arbeiten für das jeweilige Gebiet zu präsentieren. Für einige Kapitel, wie zum Beispiel für Kapitel 2, 3 oder „Erfassung der Umgebung“(Anhang D), erschien es sinnvoller, die jeweiligen relevanten Arbeiten direkt dort zu thematisieren. Einerseits da sie direkte Relation mit den jeweiligen Kapitel aufweisen und andererseits für den innovativen Kern dieser Arbeit weniger relevant sind. Am Anfang des nächsten Abschnitts wird daher nur kurz auf die Multi-Sensor-Fusion, die dann später in Kapitel 3 detaillierter behandelt wird, sowie ausführlich auf die Verwendung mehrerer Detektoren und den Entscheidungsprozess auf der Basis der gewonnenen Daten eingegangen. Danach werden die vergleichbaren Arbeiten im Bereich der interaktiven Wahrnehmung und Objekterkennung durch ein Robotersystem präsentiert. Dabei wird die Szenenanalyse als ein weiterer Schritt der Objekterkennung verstanden. Im ersten Schritt werden alle beteiligten Objekte mit 6-DOF (Position und Orientierung) erkannt. Dies ermöglicht im zweiten Schritt nicht nur eine Aussage über vorhandene Objekte, sondern auch die Bildung von Relationen zwischen diesen.

1.3.1. Objekterkennung basierend auf den Daten der Multi-Sensor-Fusion

Unter dem Begriff „Objektwahrnehmung“wird nach Wertheimer [Gol07] die Wahrnehmung von Objekten aus dem täglichen Leben verstanden. Bei der Erkennung beziehungsweise Wahrnehmung geht es um die Zuordnung eines Objekts zu einer (richtigen) Kategorie. Dass es sich bei der Objektwahrnehmung um einen äußerst komplexen Vorgang handelt, bleibt den meisten Menschen verborgen. Für den Menschen sind Erkennung und Wahrnehmung einfache und selbstverständliche Vorgänge.

Wird die Objekterkennung im Kontext der Informatik betrachtet, erfolgt die oben beschriebene Kategorisierung durch einen Abgleich von gespeicherten und wahrgenommenen Merkmalen. Für die Auffindung von Merkmalen wird eine Repräsentation der Umwelt benötigt. Diese resultiert aus der Wahrnehmung der Umgebung über die Sensoren, vgl. Kapitel 2. Dabei wird in den meisten Fällen der standardisierte offene Steuerkreis der Bildverarbeitung (2-/3-D) ein oder mehrmals durchlaufen, vgl. Kapitel 3.

Stehen mehrere Sensoren/Sensorarten zur Verfügung, können die akquirierten Sensordaten durch die (Multi-Sensor-)Fusion erweitert und/oder verbessert werden [LH91] [LH98] [Dau01]. Nach [KRT06] existieren drei mögliche Fusionsschichten, so können die

reinen Sensordaten, gefundene Merkmale sowie unabhängig getroffene Entscheidungen einzelner Detektoren fusioniert werden. In der vorliegenden Arbeit kommen alle diese Schichten zum Einsatz. Somit werden die Sensordaten fusioniert und in Form einer 3-D-Datenstruktur mit vorhandener Farbinformation zur Verfügung gestellt. Trotz der Erstellung einer gemeinsamen Struktur, in der nach Möglichkeit alle Merkmale repräsentiert werden, bleiben die Merkmale von einander unabhängig. Eine mit der vorliegenden Arbeit vergleichbare Vorgehensweise ist in [LSS13] vorgestellt, ohne im Detail auf die verwendeten Algorithmen einzugehen und mit Fokus auf der Berechnung der Position des gefundenen Clusters. In [KMP+11] werden dagegen die einzelnen Merkmale unterschiedlicher Detektoren zu einem mächtigeren Merkmal zusammengefasst.

Die meisten der in der vorliegenden Arbeit verwendeten Detektoren basieren auf der Lokalisation lokaler Merkmale und sind teilweise ineinander integriert. Zusätzlich werden abschließend auch die Entscheidungen einzelner Detektoren fusioniert und ausgewertet. Die in [ZPBB11] vorgestellte Methode kommt ohne die Registrierung aus und weist dementsprechend die erwarteten Problematiken und Unsicherheiten auf. Die Publikation stellt die Verwendung eines 2-D- und eines 3-D-Detektors dar. Trotz der interessanten Idee wird schnell ersichtlich, dass das Vorliegen mehrerer Objekte in einer Szene die Objekterkennung extrem erschwert, wenn nicht sogar unmöglich macht.

Dabei wird die Sensorkonfiguration so gewählt, dass die meisten in Kapitel 3.1 vorgestellten Objekteigenschaften detektiert werden können. Sind zusätzlich die Sensoren zueinander kalibriert, können durch die Registrierung der Sensordaten die Ergebnisse einzelner Detektoren ein weiteres Mal verbessert werden. So können zum Beispiel in einem 2-D-Bild gefundene Kanten durch den Vergleich mit einer 3-D-Punktwolke evaluiert werden. Damit werden in der vorliegenden Dissertation größtenteils alle standardisierten Möglichkeiten der Multi-Sensor-Fusion ausgeschöpft.

Im nachfolgenden Schritt werden nun die Detektoren auf die fusionierten Daten angewandt. Dabei spielt die Annahme, dass dasselbe Modell für die 3-D-Punktwolke als auch für ein 2-D-Bild verwendet werden kann, vgl. Kapitel 3.5, eine Rolle. Der Vorteil dieser Annahme ist einerseits die Erhaltung der Nachbarschaftsbeziehungen und andererseits die Möglichkeit, bekannte Algorithmen und Methoden der 2-D-Bildverarbeitung auch auf die 3-D-Punktwolken anwenden zu können. Die in dieser Arbeit genutzten Detektoren werden in Kapitel 4.2 vorgestellt. Die Ergebnisse einzelner Detektoren werden mit einer Datenbank abgeglichen und gemeinsam ausgewertet. Der nächste Abschnitt geht auf die einzelnen Detektoren sowie die gemeinsame Bildung einer Entscheidung ein.

1.3.2. Mögliche Detektoren

Wie bereits im letzten Absatz des vorhergehenden Kapitels erwähnt, kann auf eine 3-D-Punktwolke mit vorhandener Farbinformation dasselbe Modell angewendet werden wie auf ein 2-D-Bild. Unter der Verwendung der genannten Vorteile starten wir mit der Beschreibung passender 2-D-Detektoren. Anschließend wird dann die mögliche Erweiterung

auf 3-D-Detektoren betrachtet. Hier beschränken wir uns nur auf eine kurze Darstellung der möglichen Detektoren und Merkmalsräume, für die weiterführende Informationen sei der Leser auf unzählige Publikationen und Bücher verwiesen, wie zum Beispiel auf [JKS95].

Nach [AKJ02] existieren drei unterschiedliche Arten von Merkmalen, auf denen die meisten Methoden der 2-D-Bildverarbeitung unter Zuhilfenahme der Datenbank angewendet werden können. Die erste Methodengruppe verwendet die Merkmale, die auf den Farbinformationen basieren. Im Laufe der Zeit entwickelten sich viele verschiedene Farbräume wie zum Beispiel RGB, YUV, HSV usw. Die größten Vorteile dieser Merkmale sind die Lokalität innerhalb des Bildes sowie die weitgehende Unabhängigkeit von Blickwinkel und Auflösung. Da Farbe eine Eigenschaft des Lichts ist, stellen die verändernde Lichtbedingungen neben der Verdeckung die größte Schwierigkeit dieser Methode dar. Natürlich kann die Verwendung der Farbdetektoren zur Segmentierung, Objektdetektion und -erkennung verwendet werden. Die einfachste Methode, verschiedene Regionen mit bestimmter Farbinformation innerhalb des Bildes zu lokalisieren, ist die Anwendung sogenannter Schwellenwerte (engl. „Thresholds“). Aus diesem Grund beschäftigen sich viele Autoren, zum Beispiel in [GS99], mit der Anwendung von unterschiedlichen Schwellenwerten und von deren Kombinationen auf verschiedene Farbräume, mit dem Ziel, die Erkennung stabiler und sicherer zu machen. Grundsätzlich wird für die Objekterkennung ein *Apriori*-Wissen benötigt, aber auch andere Kenntnisse wie zum Beispiel hinsichtlich der Rauschverteilung innerhalb der Eingangsdaten sind von Vorteil. Prinzipiell können alle auf den Farbmerkmalen basierenden Methoden als Detektoren im Rahmen dieser Arbeit eingesetzt werden. Ein kurzer Überblick über die zur Verfügung stehenden Methoden und Algorithmen wird in [Bha11] gegeben. Für diese Arbeit wurden einige signifikante Methoden ausgewählt und integriert. Die Auswahlkriterien sowie die Beschreibung der Methoden werden in Kapitel 4.2 dargestellt.

Die zweite Art der Merkmale stellt die Textur dar. Normalerweise wird ein Eingangsbild analysiert, und ein oder mehrere Werte, wie zum Beispiel die Helligkeit, werden in Form eines Vektors zusammengefasst. Dabei wird immer davon ausgegangen, dass lokale Texturregionen in sich homogen sind. Danach werden die vorhandenen Vektoren mit den gebildeten verglichen und ausgewertet. Die Veröffentlichung [LSP05] gibt einen guten Überblick über die Verwendung möglicher Texturmerkmale. Mittlerweile überwiegen Verfahren, die die Textur mit weiteren Merkmalen kombinieren, also etwa mit der Farbe und/oder Form. Einige Arbeiten beschäftigen sich mit der Erweiterung der bestehenden Methoden auf 3-D-Daten und/oder mit der Entwicklung neuer Algorithmen für diesen Bereich, vgl. [SZ01]. Dennoch bleibt auch hier die Problematik der Verdeckung ungelöst, das zusätzliche Wissen, beispielsweise über die Struktur eingehender Bilder, steigert jedoch die Effizienz vieler Algorithmen enorm. Da einige der in dieser Arbeit eingesetzten Detektoren intern bereits mit Texturmerkmalen arbeiten, wird nicht zusätzlich ein Detektor, der nur mit Texturmerkmalen operiert, verwendet.

Die letzte Gruppe bilden die Methoden, die auf der Auffindung möglicher Formmerk-

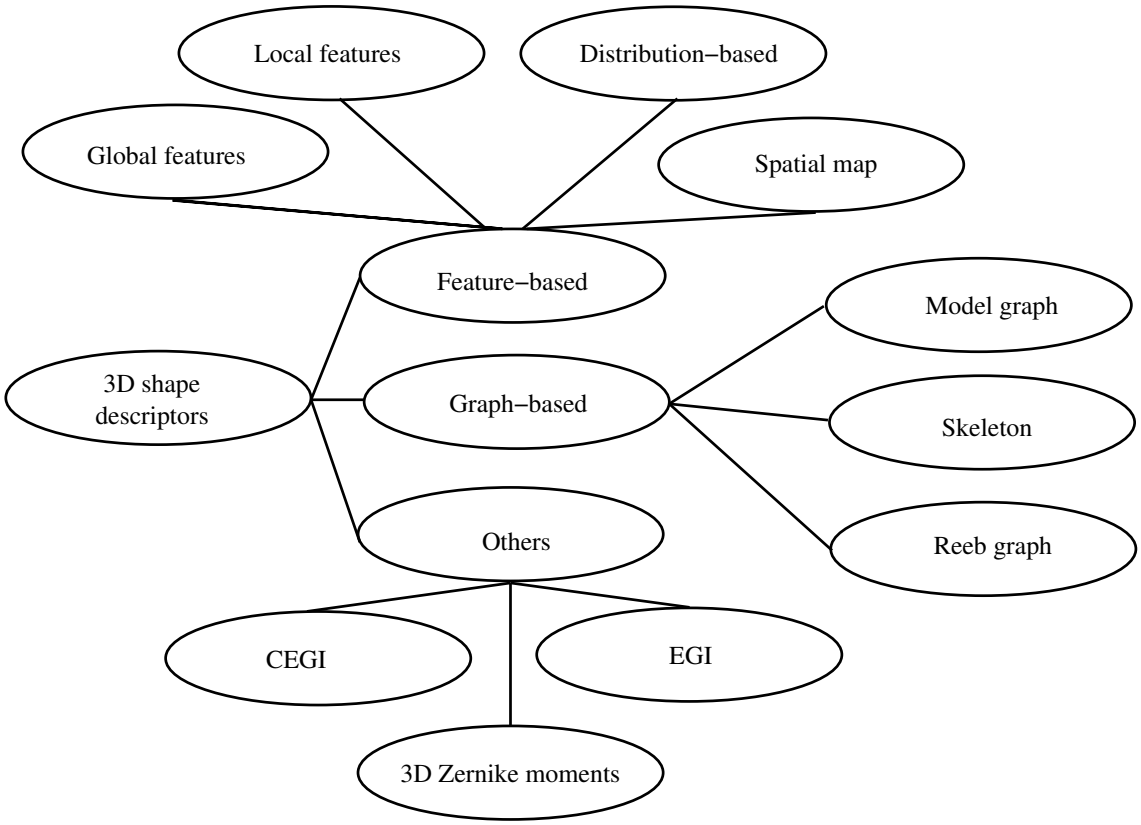


Abbildung 1.4.: Klassifikation der 3-D-Formdetektoren.

male (engl. „shape features“) basieren. Alle diese Methoden erfordern einen Vorverarbeitungsschritt, denn die Formmerkmale sollen zuerst bestimmt werden. Ein Überblick über die möglichen Verfahren und Techniken findet sich in [Lon98]. So bedienen sich die meisten Algorithmen eines Unterschieds in Farb- oder Grauwerten, Helligkeit oder Textur. Eine Übersicht über die Extraktion der Formmerkmale findet sich in [MKJ08]. Sind die Merkmale berechnet, können sie mit der Datenbank verglichen werden. Dabei ist es nicht zwingend notwendig, dass die Merkmale exakt übereinstimmen. Mehrere Verfahren versuchen, durch die Ineinanderführung der Merkmale ein Ähnlichkeitsmass zu bilden und dieses zur genaueren Kategorisierung zu nutzen [BM06]. Der Vorteil vieler dieser Methoden liegt in ihrer Schnelligkeit und Robustheit. Bedingt einerseits durch die Verwendung der 2-D-Informationen und andererseits durch die hoch frequentierte Erforschung sind oft stark verbesserte und effiziente Implementierungsvarianten vorhanden. Daher sind diese Detektoren gute geeignet, wenn es um die Kombination mehrerer Ver-

fahren geht. So liegen zuerst die Anwendung eines schnellen 2-D-Kantendetektors und die anschließende Verifikation der Ergebnisse einer 3-D-Punktwolke relativ nahe. Aber auch die exakte Größe eines Objekts kann nur in einem 3-D-Raum verglichen werden.

Wie bereits erwähnt, existieren viele Kombinationsmöglichkeiten, um die Hypothesen in einem 2-D-Raum zu bilden und diese in 3-D zu verifizieren. Die meisten der oben genannten Verfahren können auf 3-D erweitert werden. Aber auch die Anzahl der Methoden, die nur mit 3-D-Daten arbeiten, steigt kontinuierlich. Nach [ZdFF13] gibt es zwei wichtige Verfahren zum Formvergleich, nämlich die Merkmalskorrespondenz und den Abgleich über die globalen Deskriptoren. Des Weiteren können die 3-D-Formdeskriptoren in drei Kategorien unterteilt werden, die nicht immer eindeutig voneinander getrennt werden können. Abbildung 1.4 visualisiert diese Unterteilung. Wie aus der Abbildung entnommen werden kann, unterteilen sich die Formdeskriptoren in merkmalbasierte und graphenbasierte sowie andere Detektoren.

Die Merkmaldeskriptoren können ihrerseits in globale, lokale, verteilte und räumliche Deskriptoren unterschieden werden. Abgesehen von den lokalen nutzen alle Deskriptoren einen n-dimensionalen Vektor, wobei die Zahl n für alle Formen konstant gehalten wird. Dagegen operieren die lokalen Detektoren auf einzelnen Punkten, somit existiert keine Dimensionsbeschränkung. Sind die Merkmalsvektoren bestimmt, können diese verglichen werden. Dabei wird meistens die Unähnlichkeit bestimmt und darauf basierend eine Entscheidung getroffen. Ein Überblick über die möglichen Wege einer effizienten Berechnung der Merkmale findet sich in [ZC01].

Die graphenbasierten Detektoren sind dabei die komplexesten Deskriptoren. Wie der Name schon sagt, wird ein Objekt in Form eines Graphen oder eines Baums dargestellt. Wird die Repräsentation eines Objekts am Beispiel eines Baums visualisiert, so bilden die einzelnen Objektteile die Knoten. Die Kanten stellen dabei die semantischen Relationen zwischen den einzelnen Objektelementen dar. Da es keine generalisierte Repräsentation gibt, wird dieser vielversprechender Ansatz eher selten verwendet.

Zur Kategorie „andere Deskriptoren“werden das Extended Gaussian Image (EGI) [Hor84], das komplexe EGI (CEGI) [KI93] sowie die 3-D-Zernike-Momente [NK03] verstanden. Da diese keine Verwendung im Rahmen dieser Arbeit fanden, wird im Weiteren nicht näher darauf eingegangen.

Grundsätzlich können alle 3-D-Formdetektoren für die vorliegende Dissertation verwendet werden. Des Weiteren existieren unzählige Möglichkeiten der Anpassung abhängig von den charakteristischen Formmerkmalen. In Kapitel 4.2 wird explizit auf die hier genutzten Detektoren sowie deren Auswahlkriterien, Spezifikationen und Ergebnisse eingegangen. Dahinter steht die Idee einer Verwendung mehrerer Detektoren, die parallel ausgeführt werden. Im nächsten Abschnitt wird auf die Bildung einer gemeinsamen Entscheidung basierend auf den Ergebnissen einzelner Detektoren eingegangen.

1.3.3. Entscheidungsprozess basierend auf den Ergebnissen einzelner Detektoren

Die Fusion kann auf drei unterschiedlichen Ebenen erfolgen, so können Sensorrohdaten (low level), Merkmale (middle level) und die Entscheidungen (high level) fusioniert werden. Alle diese Ebenen werden in der vorliegenden Arbeit eingesetzt, vgl. Kapitel 3 zur Fusion von Rohdaten und zu einigen extrahierten Merkmalen sowie Kapitel 4 zur Entscheidungsfusion.

Zur Steigerung der Robustheit der Objekterkennung werden in dieser Arbeit mehrere Detektoren, die mit unterschiedlichen Merkmalen arbeiten, verwendet. Die Entscheidung eines Detektors i kann als $d_{i,j} \in 0,1$ gesehen werden, wobei $i = 1, 2, ..., n$ und $j = 1, 2, ..., k$. Dabei repräsentiert n die Anzahl der vorhandenen Detektoren und j die Anzahl der verwendeten Klassen. Ist $d_{i,j} = 1$, so wird die Ausgabe des entsprechenden Detektors so interpretiert, dass die eingehende Sensorinformation positiv mit der Datenbank verglichen worden ist. Im Folgenden werden vier mögliche Verfahren zur Entscheidungsfindung nach [MSDC10] präsentiert.

Das erste und wahrscheinlich bekannteste Verfahren ist der Mehrheitsentscheid. Dabei sind sich entweder alle Detektoren einig, oder die Mehrheit der Detektoren stimmt der Zugehörigkeit zu einer bestimmten Klasse zu.

Der gewichtete Mehrheitsentscheid basiert auf der Annahme, dass nicht alle Detektoren die gleiche Performanz bringen. Abhängig von den Charakteristiken werden dann einzelne Detektoren gewichtet. Basierend auf der größten Stimmenanzahl wird dann die Klassifikation vorgenommen. So kann es passieren, dass die Klassifikation auf Basis nur weniger Detektoren erfolgt, obwohl die Mehrheit für eine andere Klasse abstimmt.

Der dritte Ansatz basiert auf der Klassifikation durch eine zuvor aus den Trainingsdaten, generierte Tabelle. Durch die permanente Aktualisierung dieser Tabelle, kann die Entscheidungsfindung angepasst oder verbessert werden. Dabei sollen alle möglichen Kombinationen der Detektorenstimmen gespeichert werden, was zur einer steigenden Ressourcenanforderung führen kann.

Die letzte Methode, die naive bayesische Kombination (Naive-Bayes Combination), geht von einer konditionalen Unabhängigkeit der Detektoren untereinander aus. Da diese für die vorliegende Arbeit nicht relevant ist, wird darauf nicht näher eingegangen. Die Veröffentlichung [Kun04] gibt einen ausführlichen Überblick über die vielfältigen Kombinationsmöglichkeiten von Detektoren.

Wird keine gemeinsame Entscheidung gefunden, wie zum Beispiel im Fall einer vorhandenen partiellen Verdeckung, wird − eine weitere Neuerung dieser Arbeit − die Anwendung nicht sofort abgebrochen. Vielmehr wird durch ein regelbasiertes System zuerst nach dem Kosten-Nutzen-Prinzip über weitere Aktionen entschieden. So können zum Beispiel der Kopf, der Torso und die Plattform bewegt oder es kann sogar mit einem Manipulator direkt in die Szene eingegriffen werden. Danach wird die Szene erneut analysiert.

Der nächste Abschnitt diskutiert die wichtigste Innovation dieser Arbeit: den Roboter als interaktive Komponente der Wahrnehmung und Objekterkennung. Später in Kapitel 4 wird auf die verwendeten Detektoren, die Entscheidungsfindung und die daraus resultierenden Aktionen detailliert eingegangen.

1.3.4. Der Roboter als interaktive Komponente der Wahrnehmung und der Objekterkennung

Die Objekterkennung bildet die zentrale Komponente der Computer Vision, somit ist das Thema seit mehreren Jahrzehnten relevant und wird nach wie vor stark erforscht. Traditionelle Objekterkennungsverfahren sind passiv und arbeiten meistens an einem (2-D/3-D-)Bild, aufgenommen aus nur einer Perspektive. Viele Forscher stellten schon sehr früh fest, dass die aktiven Ansätze für die Objekterkennung ein notwendiges Mittel zur Effizienzsteigerung darstellen [Baj88][Bur88][Bal91][Tso92][Alo93]. Dabei stand vor allem das aktive Sehen im Vordergrund, was dann in den meisten Fällen durch eine Änderung der Perspektive realisiert wurde. Etwas später entstand die Idee, das aktive Sehen auf ein mobiles System zu portieren, wie zum Beispiel in [GL98]. Doch auch hier ging es nur um die Änderung des Blickwinkels auf ein zu untersuchendes Objekt. Aber auch die Verwendung mehrerer Sensoren, die an unterschiedlichen Stellen der Roboterplattform befestigt sind, wie zum Beispiel in [MBBM+14] sind denkbar.

Wie bereits erwähnt, bleibt das Thema ein stark frequentiertes und aktuelles Forschungsgebiet. So präsentiert zum Beispiel die Veröffentlichung [EGS+12] einen roboterbasierten Ansatz zur Szenenanalyse. Dabei wird die Perspektive der Sensoren auf die zu untersuchende Szene durch einen Roboter verändert. Des Weiteren wird nur ein Detektor, nämlich SIFT, eingesetzt. Somit findet weder eine Multi-Sensor-Fusion noch die Verwendung mehrerer unterschiedlicher Detektoren statt.

Ein anderer Aspekt der Verwendung mobiler Roboter wird in [VHH+11] und [EKJ07] vorgestellt. Dabei navigiert ein Roboter in der bekannten Umgebung. Während der Fahrt werden die Sensordaten akquiriert und ausgewertet, darauf basierend werden im nächsten Schritt Objekte detektiert und kartiert. Vorgestellt wird die Detektion, die nur auf einem Detektor basiert, auch eine Klassifizierung von Objekten findet nicht statt.

Die Veröffentlichung [LSN08] präsentiert in Form eines technischen Reports einen ähnlichen Ansatz wie die vorliegende Arbeit. Dabei wird ein Roboter zur aktiven Wahrnehmung und interaktiven Manipulation eingesetzt. Die Methode basiert auf einem Detektor, mehreren Perspektiven und der Möglichkeit zur Manipulation, wobei nur die Bewegungen des Objekts nach links, rechts, nach vorne und hinten erlaubt sind. Der Ansatz kann zu einem Zeitpunkt nur mit einem Objekt umgehen, auch eine partielle Verdeckung stellt ein unlösbares Problem dar. Leider ist aus der Publikation nicht ersichtlich, wie die physikalischen Aktionen eingeleitet werden, beziehungsweise wie ein möglicher Griff aus 2-D-Bildern ohne vorhandenen Tiefeninformation berechnet wird.

Die Autoren der Veröffentlichung [ZS10] planen einen ähnlichen Ansatz ohne die Ver-

wendung der Manipulation. Sie stellen ein mehrschichtiges System vor, dass auf dem Markov-Entscheidungsprozess basiert. Dabei soll der Roboter ihm bereits bekannte Objekte an den bekannten Positionen in der Karte observieren. Geplant ist der Einsatz mehrerer, nicht weiter spezifizierter Detektoren sowie eine mehrfache Änderung der Perspektive durch die mobile Plattform. Die Publikation präsentiert nur einen geplanten, groben Ansatz und geht leider nicht wirklich ins Detail. Derzeit verwenden die Autoren immer noch einen ähnlichen Ansatz mit dem Unterschied, dass mehrere heterogene Robotersysteme zum Einsatz kommen [ZS12].

Die Publikation [CBSF09] erkennt die Objekte aus einem Bild unter Zuhilfenahme lokaler Detektoren. Hier wird der Roboter für die Evaluation des Ansatzes eingesetzt. Ist das Objekt erkannt und dessen Position und Orientierung berechnet, werden durch die Berechnung des Griffs und dessen Ausführung mögliche Fehler bestimmt und zur Evaluation genutzt.

Der Ansatz in [BTM$^+$12] nutzt die aktive Wahrnehmung. Der Unterschied zu den anderen Ansätzen liegt in der Vorgehensweise der Erzeugung verschiedener Perspektiven. Ein Objekt befindet sich bereits zu Anfang der Erkennung in dem Manipulator eines Roboters. Durch dessen Bewegung werden neue Perspektiven erzeugt und Sensordaten gesammelt. Neben den unterschiedlichen Perspektiven stellt die Möglichkeit der gemessenen Änderung der Gelenkwinkel für die Rekonstruktion eines Objektes einen weiteren Vorteil dieser Methode dar. Auch hier wird deutlich, dass die aktive Wahrnehmung und Objekterkennung die Ergebnisse gegenüber den Standardverfahren deutlich verbessern kann.

Alle präsentierten Ansätze nutzen ein oder mehrere Aspekte der aktiven und/oder interaktiven Wahrnehmung. Dennoch konnte kein Ansatz gefunden werden, der die aktive Wahrnehmung, mehrere Detektoren und die Interaktion mit der Szene unter Zuhilfenahme eines mobilen Roboters konzipiert oder realisiert. Trotzdem zeigen die vorgestellten Arbeiten, dass jeder einzelne Aspekt die Szenenanalyse und Objekterkennung verbessern kann. Daher steigt die Wahrscheinlichkeit, durch die Verbindung all dieser Aspekte die bereits standardisierten Verfahren zur Szenenanalyse und Objekterkennung weiter zu verbessern, was die Aktualität, die Notwendigkeit und das Potenzial der vorliegenden Dissertation verdeutlicht. Ein weiterer Aspekt ist die Handhabung einer möglichen partiellen/totalen Verdeckung. Eine Änderung der Perspektive kombiniert mit einer Interaktion durch einen Robotermanipulator eröffnet neue Möglichkeiten für das bis heute nicht ungelöste Problem. Es ist offensichtlich, dass der Ansatz, diese aktiven Komponenten miteinander zu kombinieren, das Problem deutlich minimieren, wenn nicht sogar komplett lösen kann.

1.4. Innovative Aspekte

Das Ziel einer jeden Dissertation ist die Erschaffung neuer oder Verbesserung bereits bestehender wissenschaftlicher Konzepte und/oder Verfahren. Dieser Abschnitt soll den innovativen Beitrag dieser Arbeit herausstellen und verdeutlichen. Das primäre Ziel der Arbeit ist die Verbesserung der Objekterkennung innerhalb der Servicerobotik. Dabei bildet die Objekterkennung die grundlegende Basis für die Interaktion mit der Umgebung. Die Problemanalyse erstreckt sich über mehrere unterschiedliche Ebenen der Erkennung, angefangen bei der Datenakquisition und Datenfusion, über die Merkmalssuche und Auswertung bis hin zum Datenabgleich und der Kategorisierung.

Die Serviceroboter werden, bedingt durch die kontinuierlich fallenden Sensorpreise, mit einer wachsenden Anzahl unterschiedlicher Sensoren ausgestattet. Der Umfang der vorhandenen Informationen über den internen Zustand und die Umgebung steigt dementsprechend. Somit liegt es nahe, mehrere Daten zu fusionieren und damit die vorhandenen Daten zu ergänzen und deren Qualität zu verbessern. Für die Interaktion mit der Umgebung sind die 3-D-Daten unerlässlich. Werden diese 3-D-Punktwolken mit weiteren Eigenschaften, wie zum Beispiel Farbe, Textur etc., kombiniert, entsteht eine Punktwolke, die eine Menge an Informationen, die jeweils exakt zu einem 3-D-Punkt in Relation stehen, bereitstellt. Dafür sollen aber nicht nur die Sensordaten akquiriert, sondern auch zueinander registriert werden.

Basierend auf der Multi-Sensor-Fusion stehen dann mehr Daten zur Verfügung, was den Einsatz mehrerer Detektoren, die auf die Auffindung und Auswertung unterschiedlicher Objekteigenschaften spezialisiert sind, ermöglicht und dadurch die Qualität der Objekterkennung verbessert. Nach diesem Schritt werden viele Ergebnisse einzelner Merkmalsdetektoren bereitgestellt. Im weiteren Schritt werden diese Ergebnisse intelligent zusammengefasst. Da verschiedene Methoden der Merkmalsextraktion unterschiedlich verlässliche Ergebnisse liefern, wird ein Gesamtergebnis unter Zuhilfenahme der gewichteten Abstimmung erreicht. Die gewichtete Abstimmung bietet die Möglichkeit, den Ergebnissen qualitativ hochwertiger Methoden höhere Priorität zu verleihen und damit die Qualität des Gesamtergebnisses zu steigern. Im Fall einer Unsicherheit können weitere Erkennungsschritte zuerst durch ein regelbasiertes System eingeleitet werden.

Durch den Einsatz eines mobilen Systems kann die aktive Wahrnehmung realisiert werden. Des Weiteren wird dadurch eine Option ansatzweise offenbart, die Problematik einer vollen/partiellen Verdeckung aktiv durch einen Robotermanipulator zu reduzieren oder komplett zu lösen.

Im Einzelnen sind die meisten der oben aufgeführten Punkte nicht neu. Deren Kombination und vor allem die aktive Einbeziehung eines Roboters in den Wahrnehmungsprozess bieten aber neue Perspektiven und Möglichkeiten, die bisher nicht zugänglich waren. Das Resultat der Neukombination ist ein autonomes, aktives Wahrnehmungssystem, das die Möglichkeiten und Horizonte der Objekterkennung enorm erweitert und eine verbesserte Grundlage zum Lernen und/oder zur Wahrnehmung auf einer höheren

semantischen Ebene darstellt.

1.5. Gliederung der Arbeit

Nachdem die innovativen Aspekte der Arbeit beleuchtet wurden, wird in diesem Abschnitt der Aufbau der Arbeit dargestellt. Wird die Arbeit unter dem Gesichtspunkt der Software betrachtet, steht, wie bereits beschrieben, die Objekterkennung im Vordergrund. Aus der Hardwareperspektive bildet eine mobile Plattform den Mittelpunkt. Es wird versucht, eine mobile Plattform nach Möglichkeit variabel zu halten. Es geht viel mehr darum, der Objekterkennung die nötige Mobilität zu verleihen und damit neue Wege und Perspektiven für dieselbe zu eröffnen.

Die Wahrnehmung basiert auf den Sensoren, diese liefern die für die Objekterkennung nötigen Eingangsdaten. Kapitel 2 stellt unterschiedliche Sensoren und deren Anwendung dar. Im nächsten Kapitel werden die in dieser Arbeit verwendeten Sensoren dargestellt und ausführlich beschrieben. Des Weiteren wird auf die einzelnen Sensoreigenschaften eingegangen sowie die Möglichkeit, einige Sensoren zueinander zu kalibrieren. Durch diese sogenannte Registrierung wird es ermöglicht, die Daten einzelner Sensoren in Relation zueinander zu setzen. Dabei erlaubt der Einsatz verschiedener Sensoren, den Informationsgehalt und die Wahrscheinlichkeit der richtigen Erkennung zu erhöhen. Dies wird später in Kapitel, „Intelligente Multi-Sensor-Fusion“, ausführlich vertieft. Des Weiteren werden am Ende des nächsten Kapitels die Eigenschaften der beschriebenen Sensoren in einer Tabelle zusammengefasst, diskutiert und ausgewertet.

Kapitel 3 beschreibt die Vorgehensweisen, die Daten unterschiedlicher Sensoren so zusammenzufassen, dass alle möglichen und notwendigen Informationen zur Verfügung gestellt werden. Dabei wird versucht, die Sensoren so zu kalibrieren, dass eine Registrierung zwischen unterschiedlichen Daten verschiedener Sensoren möglich wird.

Damit die Sensorik sowie die Sensorfusion, die in letzten beiden Kapitel beschrieben worden sind, gemeinsam genutzt werden können, wird eine weitere Komponente benötigt. Dabei handelt es sich um die Umgebungserfassung, die der Gegenstand des Kapitels (Anhang D), ist. Die Umgebungserfassung erlaubt dem Roboter, die bekannte als auch unbekannte Umwelt wahrzunehmen, zu erkunden und in ihr sicher zu navigieren. Dies ermöglicht das gezielte Ansteuern einer bestimmten Position sowie die aktive Umgebungserfahrung und dadurch eine effizientere Objekterkennung.

Basierend auf einer mobilen Plattform und der Multi-Sensor-Fusion wird ein Szenario analysiert und interpretiert. Das genaue Vorgehen bei der Analyse und Interpretation ist in Kapitel 4 ausführlich dargestellt.

Kapitel 5 beschreibt die technische Realisierung des Dissertationsvorhabens. Dabei wird auf verschieden Fragen, von der verwendeten Architektur über die Integration in das EU-Projekt RACE bis hin zu Implementationsdetails, eingegangen. Außerdem wird ein Überblick über das System gegeben.

Die in der vorliegenden Arbeit vorgestellten Methoden und Verfahren werden in Kapitel 6 evaluiert. Das letzte reguläre Kapitel 7 fasst die Dissertation zusammen und stellt die möglichen Erweiterungen, Optimierungen und Verfeinerungen dar. Die Dissertation wird mit einem Anhang abgeschlossen.

Kapitel 2

Sensorik

Im Rahmen meiner Lehr- und Forschungstätigkeit am Arbeitsbereich TAMS der Universität Hamburg beteiligte ich mich an vielen unterschiedlichen Projekten, bei denen abhängig von den jeweils gestellten Aufgaben und eingesetzten Robotern verschiedene Sensoren/Sensorarten genutzt wurden. Da die meisten dieser Sensoren für die vorliegende Arbeit relevant sind, bis auf wenige Ausnahmen wie zum Beispiel die Sonarsensoren, und größtenteils sogar aktiv Verwendung finden, werden in diesem Kapitel die nötigen Sensorarten vorgestellt, deren Funktionsweisen beschrieben sowie die Vor- und Nachteile diskutiert.

2.1. Überblick über die wichtigsten Sensoren für die mobile Robotik

Doch zuerst soll der Begriff des Sensors eingeführt werden. Nach R. R. Murphy [Mur00] ist ein Sensor ein Gerät, das die Informationen über eine oder mehrere Eigenschaften der Umwelt messen kann. Damit stellt der Sensor in der Robotik eine notwendige Vorrichtung dar, die Informationen über die Eigenschaften der Umgebung liefert und Interaktionen mit dieser ermöglicht. Des Weiteren wird zwischen passiven und aktiven Sensoren unterschieden. Passive Sensoren verlassen sich auf die in der Umwelt vorhandene Energie und nehmen diese wahr. Ein gutes Beispiel dafür ist eine Kamera, die für einen störungsfreien Betrieb bestimmte Lichtverhältnisse benötigt. Aktive Sensoren senden ein (aktives) Signal aus und erlauben anhand dessen Veränderung eine Schlussfolgerung über die Umwelt. Als Beispiel sind Laserscanner gut geeignet, denn diese senden mehrere gefächerte Laserstrahlenimpulse aus und bestimmen anhand der Laufzeitdifferenz und/oder der Phasenverschiebung zu dem gesendeten und reflektierten Signal die Entfernungen zum Hindernis/Objekt. Zusätzlich wurde von Murphy der Begriff „active sensing“geprägt. Dabei handelt es sich um die Möglichkeiten zur aktiven Wahrnehmung.

So bietet eine Kamera, die fest in einem Raum montiert ist, nur die Möglichkeiten einer passiven Wahrnehmung, wohingegen eine Kamera auf einer Schwenk-Neige-Einheit auch zur aktiven Wahrnehmung eingesetzt werden kann. Des Weiteren ist die Vorstellung von einem logischen Sensor interessant, der als Kombination aus einem physikalischen Sensor mit einer Software zur Erstellung von Wahrnehmung beschrieben werden kann.

Grundsätzlich sollen Roboter sicher in ihrer Umgebung navigieren und mit ihr interagieren. Dementsprechend beschäftigt sich ein großer Teil der Robotik mit dem Aufbau und der Funktionsweise von Sensoren sowie der Verarbeitung anfallender Sensordaten. Basierend auf der von TAMS angebotenen Vorlesung [Zha12], können die Sensoren wie folgt unterschieden werden, vgl. Abbildung 2.1.

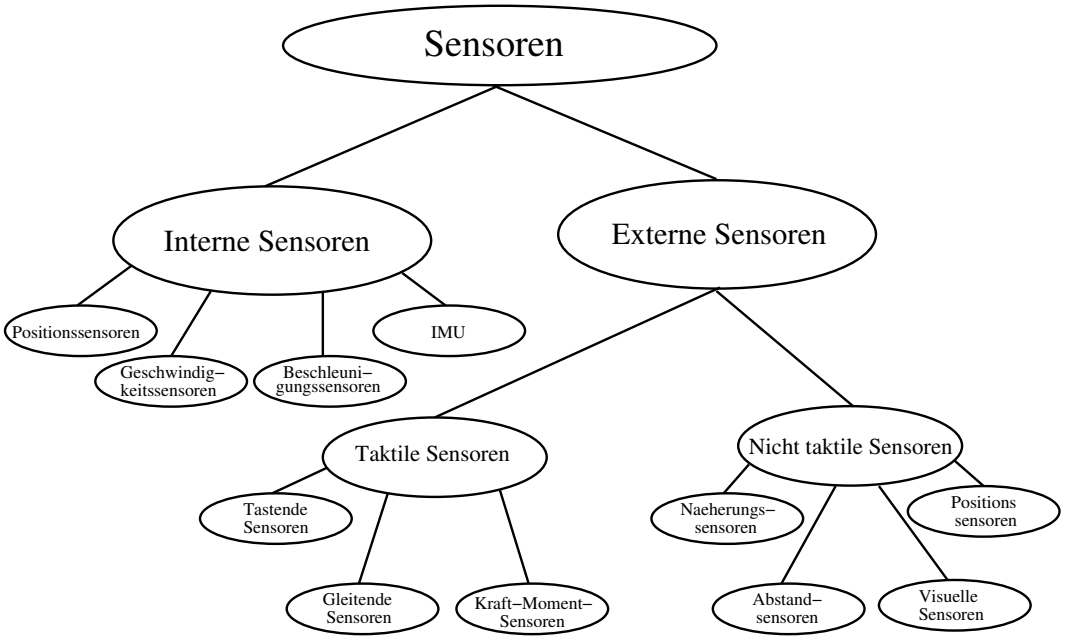


Abbildung 2.1.: Klassifikation von Sensoren.

Eine Beschreibung all dieser Sensoren würde den Rahmen der vorliegenden Arbeit sprengen. Deswegen wurde entschieden, sich nur auf die Sensoren zu konzentrieren, die für den Kern dieser Arbeit relevant sind. Natürlich werden in Teilbereichen, wie zum Beispiel im Zusammenhang mit Navigation, weitere Sensoren eingesetzt, diese werden aber als Stand der Technik betrachtet, und der Leser wird hier auf die weiterführende Literatur verwiesen [HNL12].

Die meisten der hier verwendeten Sensoren sind optischer Natur, daher wird im Rahmen dieser Arbeit auf die typische Aufteilung nach unterschiedlichen Sensorarten verzichtet. Es wird gezielt auf die Sensoren eingegangen, die für die Umgebungswahrnehmung

sowie für die Objektdetektion und Objekterkennung interessant sind. Auch ein am Ende dieses Kapitels präsentierter Vergleich dieser Sensoren wird aus derselben Perspektive durchgeführt.

Die meisten Aussagen sowie die Visualisierungen, soweit nicht weiter angegeben, sind Ergebnisse eigener Forschungsarbeiten und beziehen sich auf bestimmte Sensoren. Die Richtigkeit der Angaben und der präsentierten Sensoreneigenschaften kann nur für die getesteten Sensoren bestätigt werden, sie sollten nicht auf alle Sensoren mit ähnlichem Funktionsprinzip abstrahiert werden.

Im Folgenden werden die für die vorliegende Arbeit wichtigsten Sensoren vorgestellt und ausführlich beschrieben. Am Ende des Kapitels wird ein Vergleich unterschiedlicher Sensoren präsentiert, dessen Ergebnisse nicht nur die Auswahl des am besten geeigneten Sensors ermöglichen, sondern später auch für die Multi-Sensor-Fusion maßgeblich sind. Die Kapitel 2.2 und 2.3 basieren auf der Diplomarbeit des Autors sowie seinen weiterführenden Publikationen [KSZZ08][KSJZ09a].

2.2. Kamera

Eine der ältesten und bis heute populärsten Sensorarten in der Robotik ist die Kamera. Die zugrunde liegende Idee ist schon jahrhundertelang bekannt als die sogenannte *Camera obscura* (lateinisch *camera* für „Kammer“und *obscura* für „dunkel“). Diese kann als eine Art Kasten vorgestellt werden, der mit einer Öffnung, dem optischen Zentrum, versehen ist. Die der Öffnung gegenüberliegende Seite ist halbtransparent. Das Licht, das durch die Öffnung einfällt, erzeugt auf der diametralen, halbtransparenten Seite ein skaliertes, an horizontaler und vertikaler Achse gespiegeltes Abbild, der sich vor der Kamera befindlichen Szene. Zum ersten Mal wird dies von Aristoteles (384–324 v. Chr.) in seinem Werk „*Problemata physica*“beschrieben. Um 980 werden erste dokumentierte Experimente mit der *Camera obscura* von Abu Ali al-Hasan Ibn Al-Haitham (965-1040) durchgeführt. Viele Maler und Wissenschaftler verwendeten bis ins 19. Jahrhundert die *Camera obscura* als Werkzeug für ihre Arbeiten.

Die *Camera obscura* kann durch eine konvexe Linse erweitert werden. Fehlt diese, wird das Prinzip unter dem Begriff „*Lochkamera-Modell*“zusammengefasst. Das Lochkamera-Modell beschreibt die perspektivische Projektion eines dreidimensionalen Raums über das optische Zentrum auf eine zweidimensionale Ebene [Fau95]. Die Abbildung 2.2 stellt das Lochkamera-Modell anschaulich dar. Dabei wird durch die Zentralprojektion eine dreidimensionale Szene auf einer zweidimensionalen Ebene abgebildet.

Somit wird ein Punkt P_w im Raum mit den Koordinaten (x, y, z) auf einen Punkt $P(fx/z, fy/z)$ in der Bildebene projiziert, wobei f die Brennweite der verwendeten Kamera darstellt. Wie schon oben erläutert, findet durch die Abbildung eine Überführung von $\mathbb{R}^3$ nach $\mathbb{R}^2$ statt. Es ist ersichtlich, dass das beschriebene Modell nur in eine Richtung eindeutig ist, da bei der Projektion die Tiefeninformation verloren geht.

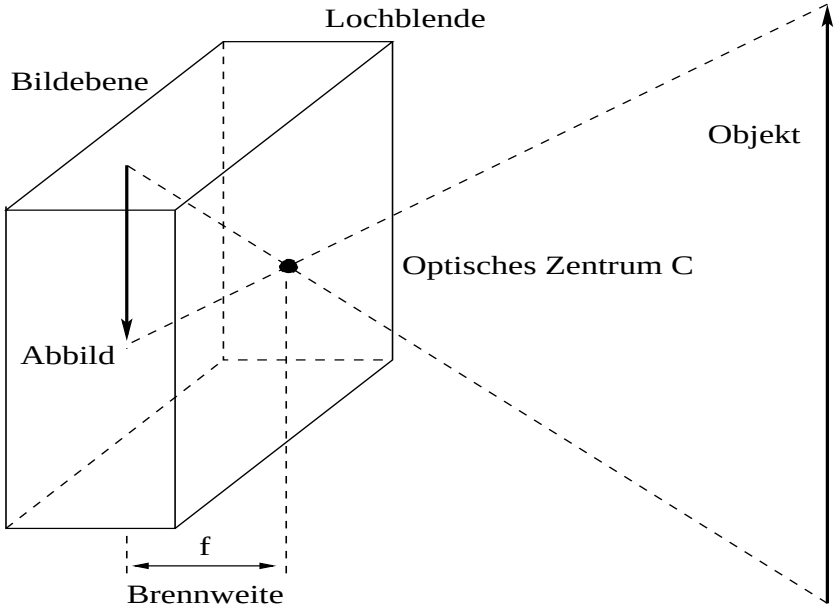


Abbildung 2.2.: Das Lochkamera-Modell. Das einfallende Licht erzeugt auf der gegenüberliegenden, halbtransparenten Seite ein skaliertes, spiegelverkehrtes und auf dem Kopf stehendes Abbild der vor der Kamera befindlichen Szene. Mathematisch ausgedrückt, wird eine dreidimensionale Szene auf einer zweidimensionalen Ebene unter Zuhilfenahme der Zentralprojektion abgebildet.

Das Lochkamera-Modell stellt nur ein abstraktes Modell einer realen Kamera dar und wird in der Wissenschaft zur Beschreibung der grundlegenden mathematischen Zusammenhänge einer realen Kamera verwendet. Um Beugungsfehler und Schärfeverlust zu vermeiden, wird das Loch des Modells als unendlich klein angenommen, was nur theoretisch möglich ist. Um dieser Annahme nahezukommen, werden in der Praxis Objektive verwendet, die aus einer oder mehreren Linsen bestehen. Aber auch die Verwendung von Objektiven hat Nachteile, so werden zum Beispiel nur die Gegenstände scharf abgebildet, die sich in einem bestimmten Abstand von der Kamera befinden. Außerdem verursacht jede Linse und die Geometrie des Sensor-Chips durch die Ungleichmäßigkeiten und Fehler in der Struktur eine Verzerrung.

Durch den Prozess der Kamerakalibrierung wird es möglich, die Beziehung zwischen den Weltkoordinaten und Bildkoordinaten der Kamera mathematisch zu beschreiben [KKS96]. In der Abbildung 2.3 werden die Beziehungen zwischen den Koordinatensystemen der Welt, der Kamera und dem auf der Sensorfläche entstehenden Bildes grafisch dargestellt. Dabei wird ein Objekt im Weltkoordinatensystem über das optische Zentrum C in das Kamerakoordinatensystem transformiert und auf die Bildfläche projiziert.

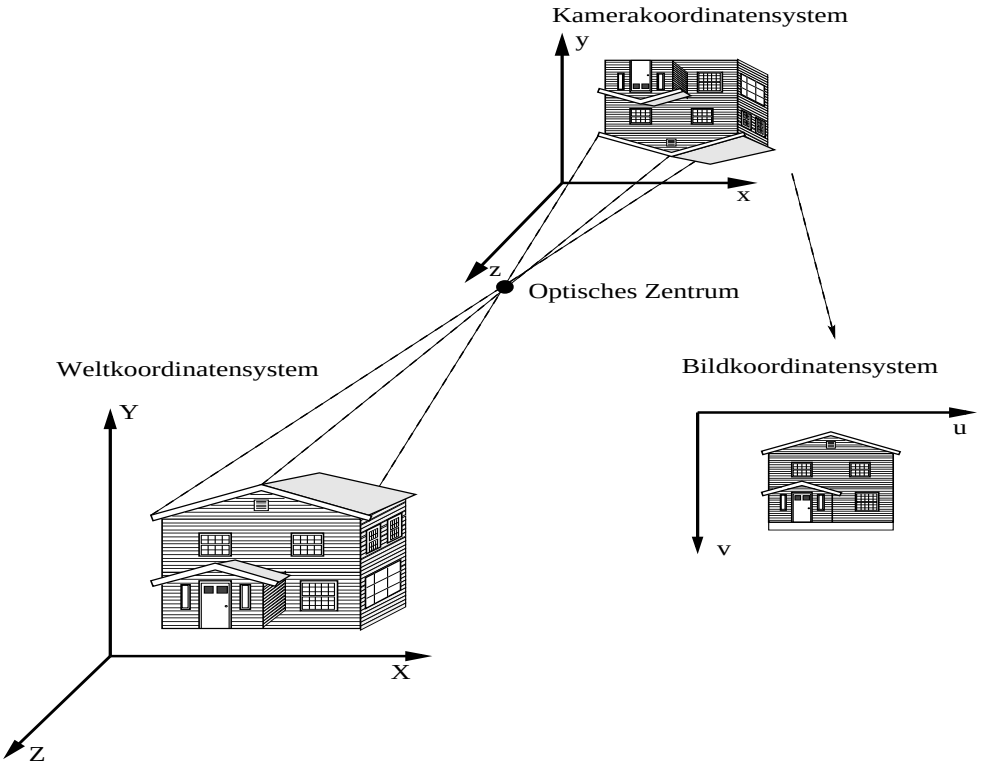


Abbildung 2.3.: Beziehungen zwischen den Koordinatensystemen während der Bildaufnahme. Über das optische Zentrum C wird ein Objekt in das Kamerakoordinatensystem transformiert und auf die Bildfläche projiziert. Das einfallende Licht verursacht auf der Sensorfläche einer Kamera Ladungsänderung und ermöglicht durch anschließende Quantisierung eine Digitalisierung der Bildinformation.

Durch den lichtelektrischen Effekt kann ein zweidimensionales Bild digitalisiert werden. Um von dem gewonnenen Bild auf die Weltkoordinaten zurückschließen zu können, werden kameraspezifische Parameter benötigt. Diese werden über die Kamerakalibrierung bestimmt und bestehen aus den extrinsischen Parametern der Kamera (Rotation und Translation) sowie den intrinsischen (Brennweite, Bildhauptpunkt) und den weiteren Parametern für die Linsenverzeichnungen. Sind diese bekannt, so ist es möglich, die tatsächliche Größe des abgebildeten Objekts sowie die durch den Abbildungsprozess verlorene Tiefeninformation zu rekonstruieren. Eine detaillierte Beschreibung der Kamerakalibrierung sowie der Kalibrierung eines Stereokamerasystems kann im Anhang C dieser Arbeit gefunden werden.

2.3. Stereokamerasystem

Es gibt mehrere Möglichkeiten, um aus einer zweidimensionalen Bildebene die Tiefeninformationen einer dreidimensionalen Szene wieder rekonstruieren zu können: Die Originalgröße eines Objekts im Bild ist bekannt, eine und dieselbe Szene wird aus zwei oder mehr unterschiedlichen Blickwinkeln aufgenommen, oder es werden zwei Kameras eingesetzt. Da ein Einbau von zwei meist identischen Kameras einerseits plausibel und andererseits biologisch inspiriert ist, stellen die Stereokamerasysteme einen seit Jahrzehnten bekannten Sensor für die 3-D-Wahrnehmung der Umgebung dar. Abbildung 2.4 präsentiert zwei in der vorliegenden Arbeit eingesetzte Kameras. Links ist ein selbst entworfenes 3-D-Wahrnehmungssystem zu sehen: Das Stereokamerasystem besteht hier aus zwei Sony XC-999P-Kameras mit VCL-06S12XM-Objektiven und ist auf ca. 15 cm Basislinie montiert auf einem Pioneer 2-DX-Roboter von ActiveMedia Robotics. Rechts ist ein Stereokamerasystem STH MDCS mit 9 cm Basislinie von Videre Design[1] zu sehen.

Dabei wird eine Szene mit beiden Kameras zum gleichen Zeitpunkt (synchronisiert) aufgenommen, anschließend wird nach korrespondierenden Punkten in beiden Bildern gesucht. Ausgehend von dem Abstand der korrespondierenden Punkte, der als Disparität bezeichnet wird, kann die Tiefe dieses Punkts im Raum mittels Triangulation rekonstruiert werden.

Abbildung 2.4.: Zwei in der vorliegenden Arbeit eingesetzte Stereokamerasysteme. Auf der linken Seite ein eigen entworfenes 3-D-Wahrnehmungssystem. Das Stereokamerasystem besteht aus zwei Sony XC-999P-Kameras mit VCL-06S12XM-Objektiven und einer ca. 15 cm Basislinie, montiert auf einem Pioneer 2-DX-Roboter der ActiveMedia Robotics. Auf der rechten Seite ein weiteres Stereokamerasystem, STH-MDCS mit 9 cm Basislinie von Videre Design.

Bevor jedoch die Suche nach korrespondierenden Punkten durchgeführt werden kann, müssen die beiden Kameras kalibriert werden. Damit können die Epipolarlinien parallel ausgerichtet werden, dies wird als Rektifikation der Bilder bezeichnet, was die Korres-

[1] http://www.videredesign.com/

pondenzsuche auf eine zu durchsuchende Bildzeile beschränkt. In der Literatur werden mehrere Möglichkeiten, ein Stereosystem zu kalibrieren, beschrieben, hier werden allerdings nur die Grundlagen und Ziele der Kalibrierung eines Stereosystems skizziert.

Abbildung 2.5.: Gewinnung der Tiefeninformation mit einem Stereokamerasystem. Von links nach rechts: rektifiziertes Image der rechten Kamera, Disparitätsbild erstellt unter Zuhilfenahme des Birchfield-Algorithmus, Rekonstruktion.

Um das Stereosystem kalibrieren zu können, werden die beiden Kameras zuerst einzeln kalibriert, dabei sind nur die intrinsischen Parameter relevant. Mit den intrinsischen Parametern wird die Linsenverzeichnung aus den Bildern herausgerechnet. Anschließend werden die neuen extrinsischen Parameter geschätzt, wobei nicht die Rotation und Translation bezogen auf den Koordinatenursprung bestimmt, sondern eine der Kameras als Referenzpunkt betrachtet wird [GH01]. Bei der in dieser Arbeit verwendeten Kalibrierung wird die Rotation und Translation der rechten Kamera in Relation zu der linken Kamera bestimmt.

Somit werden die Kameras abhängig voneinander ausgerichtet, die Parameter der beiden Kameras können zu einem Stereokameramodell zusammengefasst werden. Mit den geschätzten Parametern kann ein ideales Stereokamerasystem simuliert werden, in dem die Bildebenen koplanar und die optischen Achsen parallel ausgerichtet werden. Damit liegen die beiden Epipole im Unendlichen, die Epipolarlinien sind auf der gleichen Höhe, parallel zur X-Achse, und die zu suchenden Korrespondenzpunkte befinden sich in der gleichen Zeile der beiden Bilder.

Abbildung 2.5 visualisiert die Gewinnung der Tiefeninformation durch ein Stereokamerasystem. Zuerst werden die Bilder rektifiziert, dabei stellt das Bild links das rektifizierte Image der rechten Kamera dar. Die Bilder werden so zueinander transformiert, dass die korrespondierenden Punkte in einer Bildzeile liegen. Danach werden, hier unter Zuhilfenahme des Birchfield-Algorithmus [BT98], die korrespondierenden Punkte bestimmt, und es wird eine Disparitätskarte (Abbildung in der Mitte) berechnet. Aus einem sogenannten 2,5-D-Disparitätsimage kann dann die 3-D-Umgebung rekonstruiert werden (Abbildung 2.5 rechts). Die drei Bilder stammen von dem Stereokamerasystem des hu-

manoiden Roboters Hoap 2[2].

2.4. TOF-Kameras

Im Gegensatz zu den Stereokamerasystemen sind die 3-D-TOF-Kamerasysteme (Laufzeitverfahren, engl. „time of flight“), auch bekannt als PMD (Photomischdetektor, engl. „photonic mixing device“), eine relativ neue Entwicklung. Wie die Bezeichnungen schon andeuten wird die Umgebung mittels eines Lichtimpulses ausgeleuchtet, für jeden Bildvoxel (ein Voxel ist eine Erweiterung des Pixels um eine weitere, also die dritte Dimension) wird die Laufzeit vom Aussenden über die Reflexion an einem Objekt bis zum Empfang des reflektierten Impulses gemessen. Die Entfernung ist direkt proportional zu der gemessenen Laufzeit und kann mit der folgenden Gleichung bestimmt werden:

$$d = \frac{\Delta t}{2} \cdot c, \tag{2.1}$$

wobei d ist Entfernung zum Objekt, c die Lichtgeschwindigkeit in der Luft und Δt die oben beschriebene gemessene Impulslaufzeit.

Das zugrunde liegende Prinzip, das Aussenden mehrerer Lichtimpulse und der Empfang deren Reflexionen, ist dasselbe wie beim Laserscanner. Der große Vorteil dieser Technik liegt einerseits darin, eine ganze Szene in einem Durchgang abzutasten, was eine hohe Bildfrequenz (engl. „frame rate“) ermöglicht wie zum Beispiel bei der PMD CamCube 3.0. Diese TOF-Kamera erreicht eine Bildfrequenz von 40 fps (frames per second) bei einer Maximalauflösung von 204×204 Voxels[3]. Die geringere Auflösung stellt jedoch einen der größten Nachteile dieser Technologie dar. Die CamCube mit den oben genannten Daten ist derzeit die TOF-Kamera auf dem Markt, die zurzeit die höchsten verfügbaren Auflösungen liefert (Stand 06/2011). Der Entfernungsmessbereich ist werksseitig anpassbar, CamCube nimmt dabei, wie die meisten vergleichbaren Produkte, einen Bereich von $0,3\,m$ bis zur $7\,m$ wahr. Es sind aber durchaus größere Distanzen möglich.

[2] http://www.fujitsu.com

[3] http://www.pmdtec.com/fileadmin/pmdtec/downloads/documentation/datenblatt_camcube3.pdf

Der Messbereich ist direkt proportional zur Modulationsfrequenz und kann auf folgende Weise berechnet werden:

$$d_{max} < \frac{c}{2f_m}, \tag{2.2}$$

wobei d_{max} ist die maximale Abtastrate, c ist die Lichtgeschwindigkeit in der Luft und f_m ist die die Modulationsfrequenz. Wenn von der maximalen Distanz der CamCube 3.0 ausgegangen wird, muss die Modulationsfrequenz bei ca. 20 MHz liegen.

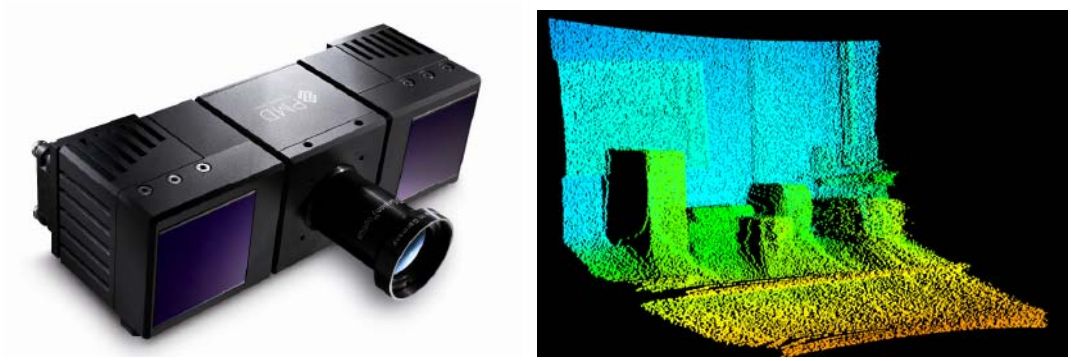


Abbildung 2.6.: PMD CamCube 3.0 (links) und ein entsprechendes Bild mit farbkodierten Tiefeninformationen (rechts)

Abbildung 2.6 zeigt die PMD CamCube 3.0 sowie ein Bild der Tiefenkamera mit farbkodierten Tiefeninformationen, ein sogenanntes 2,5-D-Bild. Damit die Daten der Kamera weiter genutzt werden sowie für eine bessere Integration in die vorhandene Programmierumgebung, wurde die CamCube in ROS[4] (Robot Operating System) integriert. Der eigens implementierte Node (ein Prozess im ROS, der eine Berechnung ausführt und mit den anderen Nodes kommuniziert) nutzt die Treiber von PMD und liefert die 3-D-Daten. Die Visualisierung der Daten in RVIZ (ein Visualisierungswerkzeug in ROS) ist in der Abbildung 2.7 zu sehen. In der oberen linken Ecke findet sich ein Graustufenbild, das nur dem besseren menschlichen Verständnis der Szene dienen soll und durch die Anpassung der Kameradaten (Intensitäten) an den maximalen Wert bestimmt werden kann.

Grundsätzlich besteht die TOF-Kamera aus mindestens vier Komponenten: Erstens aus einer Beleuchtungseinheit, am häufigsten werden Laserdioden oder LEDs verwendet, die die Umgebung ausleuchten sollen. Die Lichtquellen sollten extrem schnell moduliert werden können, damit eine Laufzeitmessung exakt durchgeführt werden kann. Das ausgestrahlte Licht wird im nahen Infrarotbereich und mit einer Impulsdauer im Nanosekundenbereich gesendet. Dieser Umstand hat zwei Vorteile: Einerseits wird die Umgebung dadurch nicht gestört, andererseits kann durch eine Bandpassfilterung nur

[4] http://www.ros.org

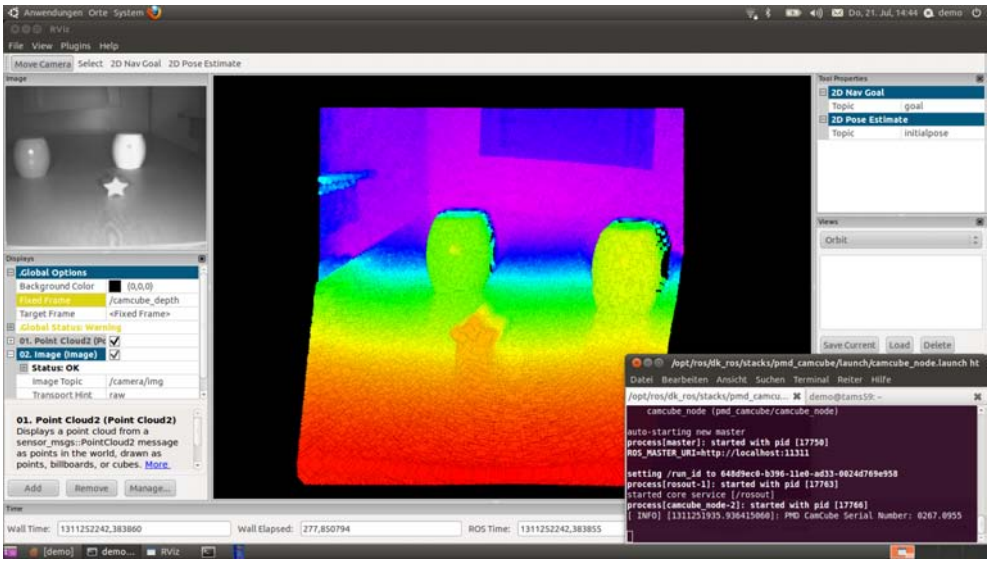


Abbildung 2.7.: Visualisierung der CamCube Daten in ROS. RVIZ, ein ROS-eigenes Visualisierungswerkzeug, stellt die farbkodierte Tiefenkarte, die in alle drei Richtungen per Maus gedreht werden kann, dar. Das Bild in der oberen linken Ecke ist ein künstlich erzeugtes Graustufenbild, das nur dem besseren menschlichen Verständnis der Szene dient und durch die Anpassung der Kameradaten an den maximalen Wert bestimmt werden kann.

die Wellenlänge durchgelassen werden, mit der auch die Beleuchtung arbeitet. Damit kann ein Teil der störenden Umgebungsbeleuchtung eliminiert werden, was die Kamera gegenüber variierenden Lichtverhältnissen etwas unabhängiger macht. Zweitens wird eine Optik benötigt, die, wie bei den anderen Kameras auch, das reflektierte Licht sammelt und auf die nächste notwendige Komponente, den Sensor, abbildet. Der Aufbau des Sensors ist aber gegenüber den Standardkameras deutlich komplizierter. Die einzelnen Pixel des Sensors sollen nicht nur das Licht sammeln, sondern auch seine Laufzeit messen können. Dadurch wird der Sensor vergrößert, was einen weiteren Faktor der niedrigen Auflösung darstellt. Das letzte und nicht weniger komplizierte Bauteil ist die Ansteuerungselektronik. Die Szenenausleuchtung und der Sensor müssen miteinander extrem genau synchronisiert werden. Fehler bei der Synchronisation führen zu ungenauen Distanzen und nicht akkuraten Ergebnissen.

Die beiden letzteren Komponenten verursachen die hohen Preise der heutigen TOF-Kameras. Der Vorteil der TOF-Kameras liegt jedoch in der Genauigkeit ihrer Tiefenkarte, die zum Beispiel für die Kantendetektion relevant ist. Der Nachteil ist demgegenüber die niedrige Auflösung, sodass die Daten in einer größeren Umgebung kaum genutzt werden können.

2.5. Strukturiertes Licht

Die Kombination aus einem Projektor, der ein geometrisch bekanntes Muster auf die Umgebung projiziert, und einer Kamera, die das mit dem Muster beleuchtete Umfeld wahrnimmt, ist ein seit mehreren Dekaden bekanntes Verfahren der Robotik, um die Tiefeninformation eines 2-D-Bildes zu rekonstruieren. Aus der bekannten Geometrie und der gemessenen Größe des projizierten Musters im Kamerabild kann die Tiefe rekonstruiert werden, dieses Verfahren ist unter dem Namen „Strukturiertes Licht“ein fester Bestandteil der Forschung [FSV04]. Dennoch gestaltet sich die Verwendung solcher geometrischer Muster in der Praxis eher schwierig. Die meisten Geräte sind groß und schwer, an ein kompaktes Design war lange Zeit nicht zu denken, und der Einsatz war größtenteils auf die Laborumgebung beschränkt. In den meisten Fällen wird ein sichtbares Muster verwendet. Einerseits ist dies mit geringeren Kosten zu begründen. Andererseits erlauben die sichtbaren Muster eine höhere Auflösung als die für den Menschen unsichtbaren Lichtmuster.

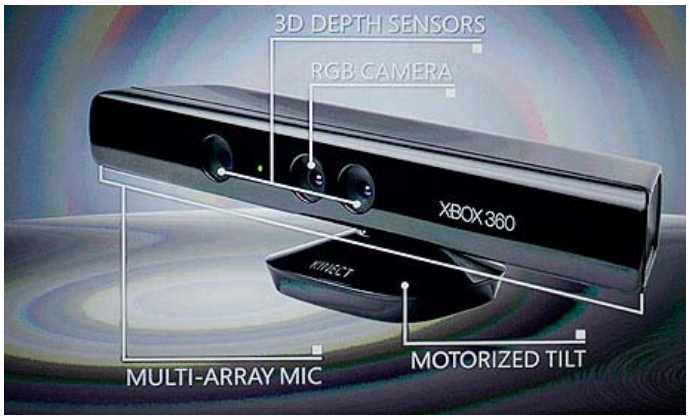


Abbildung 2.8.: Abbildung der Kinect, einer Erweiterung der Microsoft Xbox-360-Spielkonsole. Eine Entwicklung in Zusammenarbeit mit der israelischen Firma PrimeSense.

Im November 2010 brachte Microsoft [5] gemeinsam mit der israelischen Firma PrimeSense[6] eine Erweiterung für die Xbox-360-Spielkonsole[7] unter dem Namen Kinect siehe Abbildung 2.8 auf den Markt. Das Potenzial dieses neuen Sensors wurde ziemlich schnell erkannt, zusammen mit den zuerst inoffiziellen und später offiziellen Treibern revolutionierte die Kinect die internationale Robotik-Forschung. Die hohen Absatzzahlen für diesen Sensor lassen sich durch die vorhandenen Treiber und durch den Preis erklären

[5] http://www.microsoft.com
[6] http://www.primesense.com
[7] http://www.xbox.com

[8]. Eine typische Punktwolke und ein Farbbild, die durch den Kinect-Sensor akquiriert worden sind, zeigt die Abbildung 2.9.

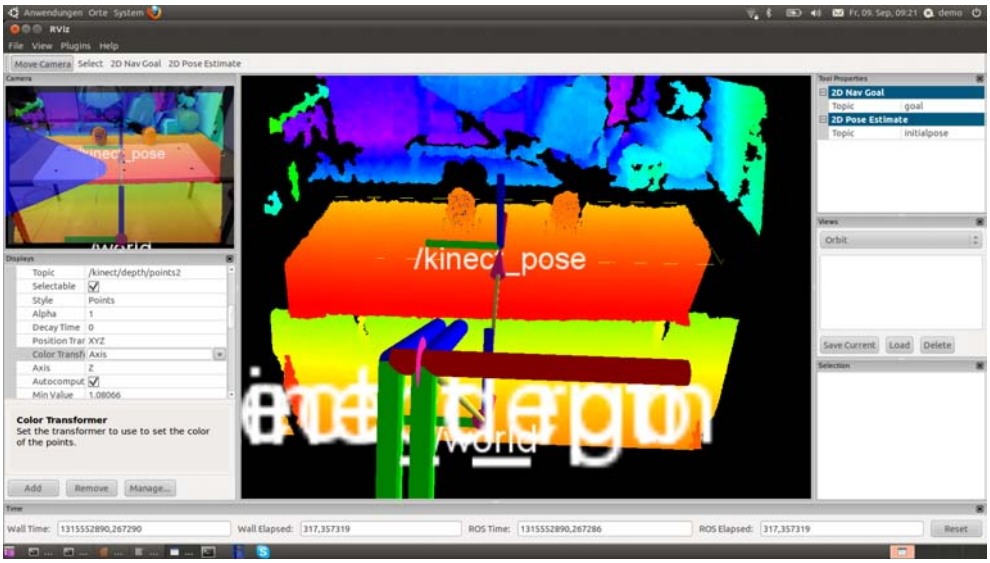


Abbildung 2.9.: Mit der Kinect aufgenommene Punktwolke sowie ein Farbbild, die durch den Kinect-Sensor akquiriert worden sind.

Der Sensor besteht aus einer Farbkamera mit der VGA-Auflösung $[640 \times 480]$ Pixel bei 32 Bit Farbtiefe und einer Geschwindigkeit von 30 fps. Der horizontale Öffnungswinkel beträgt 57°, der vertikale 43°. Die Bestimmung der Tiefeninformation, typisch für ein „Strukturiertes Licht“einsetzendes Verfahren, basiert auf einer Kombination aus einem IR (Akronym für Infrarot) Projektor und IR-Kamera. PrimeSense adaptierte das Signal des Projektors sowie der Kamera in das Infrarotspektrum und machte damit die Lichtmuster unsichtbar für das menschliche Auge.

Die nachfolgende Abbildung 2.10 von PrimeSense visualisiert die wichtigsten Komponenten des Prime-Sensors. PrimeSense präsentiert ein eingebettetes System, das sogenannte SoC (System on Chip), welches mit dem CMOS-Bildsensor verbunden ist. Der darauf laufende Algorithmus ist stark parallelisiert, empfängt das reflektierte nahe Infrarotlicht und berechnet die Tiefeninformation. Das System benutzt ein über die Zeit kontinuierliches Lichtmuster und unterstützt damit das aktive Sehen beziehungsweise die aktive Triangulation.

Die offene Kinect mit den zwei Kameras und dem IR-Projektor ist in der Abbildung 2.11 dargestellt. Die Basislinie, der Abstand zwischen der IR-Kamera und dem IR-Projektor, beträgt ca. 7.5 cm.

[8] http://www.slashgear.com/microsoft-kinect-sells-2-5-million-units-in-25-days-30116846/

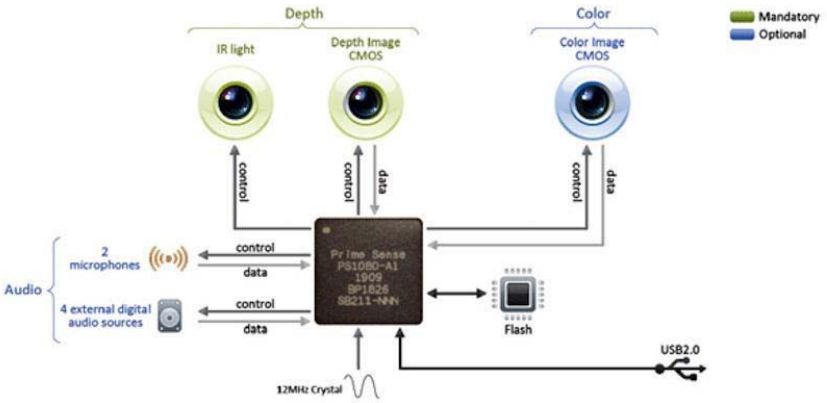


Abbildung 2.10.: Die interne Ansicht des Prime-Sensors. Im Kern befindet sich ein eingebettetes System, ein sogenanntes SoC (System on Chip), das mit dem CMOS-Bildsensor verbunden ist.

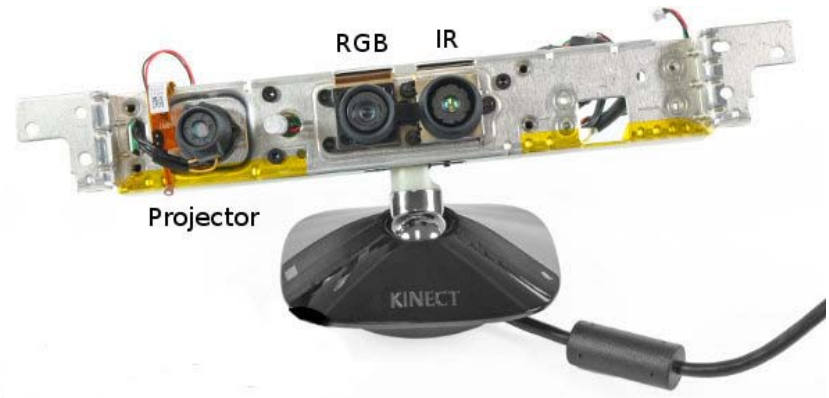


Abbildung 2.11.: Die offene Kinect mit zwei Kameras und einem IR-Projektor.

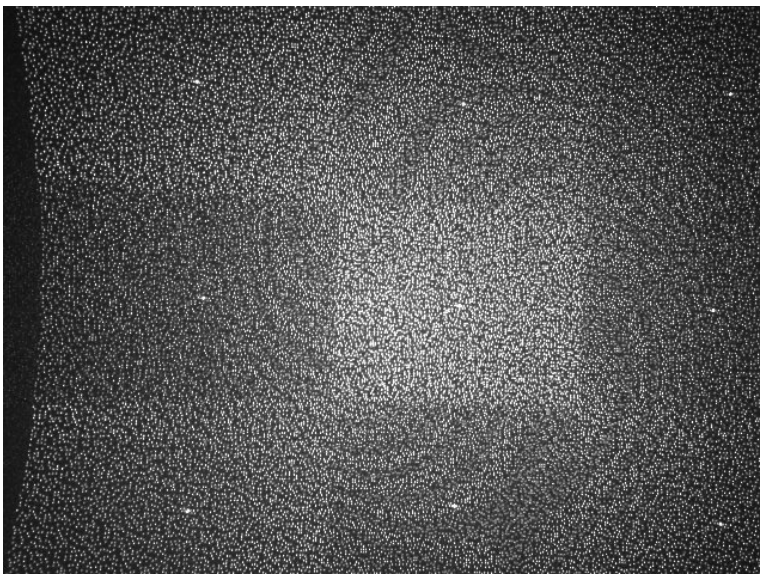


Abbildung 2.12.: Das zur Bestimmung der Tiefeninformation verwendete Muster.

Die exakte Beschreibung des Verfahrens zur Tiefenbestimmung unterliegt einem Patent von PrimeSense, sodass hier weiter Details nicht wiedergegeben werden können. Das verwendete Muster kann jedoch in der Abbildung 2.12 betrachtet werden. Nach inoffiziellen Quellen vergleicht der Sensor die erfassten Daten mit den bei der Herstellung vorkalibrierten Werten und erreicht eine Genauigkeit von $\frac{1}{8}$ Pixel bei einer Auflösung von $[320 \times 240]$ Pixel und einer Tiefenauflösung von 16 Bit sowie einer Geschwindigkeit von 30 fps. Die erreichte Geschwindigkeit von 30 Bildern pro Sekunde ermöglicht nicht nur die 3-D-Rekonstruktion, sondern auch die Verfolgung eines Objekts im 3-D-Raum (engl. „tracking“) oder sogar die 3-D-Kollisionsvermeidung.

Die Genauigkeit der Kinect ist erstaunlich und wird in der Abbildung 2.13 visualisiert. Die Abbildung ist dem ROS-Projekt[9] entnommen. Zusätzlich wird die gegenseitige Beeinflussung zweier Kinect-Sensoren dargestellt.

Des Weiteren verfügt der Sensor über eine Neigeeinheit (engl. „tilt-unit“) mit einer physikalischen Auflösung von $\pm 31°(27°)$ und mehreren Mikrofonen mit 16 Bit Audio bei 16 kHz. Der Arbeitsbereich der Kinect liegt ca. zwischen 0.7 bis 6.0 Metern. Damit stellt die Kinect im Indoor-Bereich einen mächtigen Sensor dar, der Tiefeninformationen und Farbinformationen liefert und dabei extrem schnell ist. Aus diesem Grund ist sie ein optimaler Sensor für den Einsatz in Innenräumen. Mit ihren Eigenschaften ist die Kinect ideal für planare Oberflächen, ihre Nachteile erweisen sich an den Kanten: einerseits durch nur $\frac{1}{4}$ der VGA-Auflösung und das Prinzip des projizierten Musters. Trifft

[9] http://www.ros.org/wiki/openni_kinect/kinect_accuracy

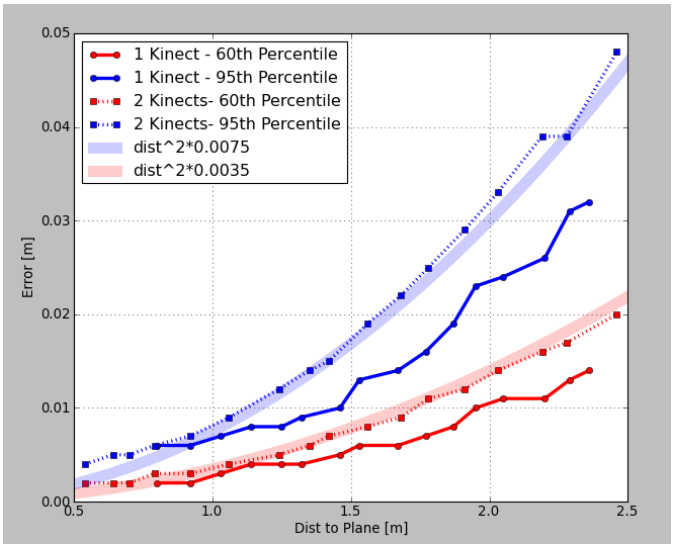


Abbildung 2.13.: Die Genauigkeit der Kinect. Die Abbildung ist dem ROS-Projekt entnommen. Zusätzlich wird die gegenseitige Beeinflussung zweier Kinect-Sensoren dargestellt.

das Muster nicht exakt die Kante, entstehen Ungenauigkeiten, die die Kantendetektion erschweren.

Abbildung 2.14.: Links die ASUS Xtion PRO LIVE im Vordergrund und die Microsoft Kinect im Hintergrund. Das mittlere und rechte Bild zeigen die Montage der ASUS Xtion PRO LIVE auf der, in der vorliegenden Arbeit genutzten, Evaluationsplattform (vgl. Kapitel 6).

Ein weiterer interessanter Sensor, der auf demselben Prime-Sensor basiert, ist die ASUS Xtion Pro Live, dargestellt in der Abbildung 2.14. Durch den Verzicht auf eine Neigeeinheit konnte der Stromverbrauch reduziert werden. Damit begnügt sich der Sensor mit einem USB-Anschluss und ist deutlich kleiner. Da die beiden Kameras stärker

symmetrisch zueinander angeordnet sind, ist der Sensor für Robotikanwendungen sowie im humanoiden Bereich deutlich besser geeignet [KH11].

Der größte Vorteil dieser Kinect-ähnlichen Kameras liegt darin, dass sie Farb- und Tiefenmesssensoren in einem Gehäuse vereinen. Werden beide Sensoren relativ zueinander kalibriert an die Synchronisation muss hier nicht mehr gedacht werden, liefert die Kamera ein durch die Farbinformationen ergänztes Tiefenbild (ein sogenanntes RGB-D-Image). Jedem Farb- wird ein Tiefenwert zugeordnet oder umgekehrt. Damit können gewisse Hypothesen aus einem Raum für einen anderen direkt überprüft, ergänzt und optimiert werden.

2.6. 2-D-Laserscanner

Bei einem 2-D-Laserscanner handelt es sich um einen optischen Sensor, der ein Lichtsignal ausstrahlt und anhand des empfangenen, an dem Objekt reflektierten Lichtstrahls die Entfernung zum Hindernis bestimmen kann. Auch dieses Verfahren, wie alle davor aufgeführten Methoden, funktioniert berührungslos und benötigt keine Reflektoren. Die typischen Laserscanner-Sensoren, die derzeit am häufigsten für unterschiedlichste Robotikanwendungen verwendet werden, sind in der Abbildung 2.15 dargestellt.

Abbildung 2.15.: Typische Laserscanner-Sensoren, die derzeit am Meisten für die Robotikanwendungen verwendet werden. Von links nach rechts: SICK LMS200, Hokuyo UTM-30LX und URG-04LX.

Die Funktionsweise kann am besten anhand der Abbildung 2.16 erklärt werden. Diese Abbildung visualisiert den internen Aufbau sowie die Funktionsweise des Hokuyo URG-04LX. Im Folgenden wird die Arbeitsweise eines 2-D-Laserscanners am Beispiel des Hokuyo URG-04LX Laserscanners, angelehnt an [KOY+05][KYM05], verdeutlicht.

Der Laser sendet einen punktuellen Impuls aus, welcher durch einen rotierenden Spiegel auf einer horizontalen Ebene in alle Richtungen abgelenkt wird. Durch die konstante Geschwindigkeit des Spiegels wird der Winkel zwischen benachbarten Laserstrahlen konstant gehalten, was eine Transformation der errechneten Distanz in das Koordinatensystem der Umgebung ermöglicht. Generell werden zwei grundlegende Methoden, nämlich Lichtlaufzeit- und Phasendifferenzverfahren, unterschieden. Beim Lichtlaufzeitverfahren wird die Distanz anhand der Gleichung 2.1 bestimmt. Dass dieses Verfahren nur für die

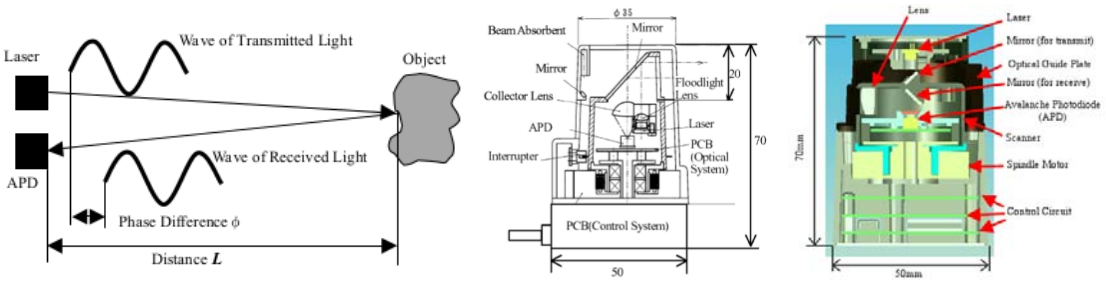


Abbildung 2.16.: Der interne Aufbau und die Funktionsweise eines 2-D-Laserscanners des Hokuyo URG-04LX [KOY+05][KYM05].

längeren Distanzen zum Hindernis funktioniert, ist aus dem Gedankenexperiment zur Berechnung der notwendigen Abtastrate, schnell ersichtlich. Aus diesem Grund werden in den meisten Laserscannern beide Verfahren, Lichtlaufzeit- und Phasendifferenzverfahren, kombiniert. Die linke Seite der Abbildung 2.16 zeigt schematisch das Grundprinzip des Phasendifferenzverfahrens und die Gleichung 2.3 den mathematischen Hintergrund. Dabei werden mehrere Signale unterschiedlicher Frequenz generiert, im Fall von Hokuyo URG-04LX zwei mit jeweils $46,55\,MHz$ und $53,2\,MHz$. So entsteht im Fall von Hokuyo URG-04LX eine Sinuskurve aus mehreren, nämlich 30, nacheinander ausgesendeten Lichtimpulsen. Diese Impulse sind paarweise verschieden durch die Verwendung von Signalen unterschiedlicher Frequenz. Das zur Sinuskurve zusammengesetztes Signal wird ausgesandt und seine Reflexion an einem Hindernis empfangen. Die Fotodiode des Laserscanners empfängt das Signal, und anhand der Phasenverschiebung kann die zurückgelegte Distanz berechnet werden. Der Aufbau bietet eine Menge an möglichen Optimierungen, von der Anpassung der Intensität der Fotodiode bis hin zur Verwendung mehrerer Frequenzen und der Generierung einer Lichtwelle durch deren abwechselnde Verwendung. Da die Messwerte schnellstmöglich an den Benutzer übertragen werden müssen, werden diese pro Scann bestimmt und an den Rechner gesendet. Da diese Prozedur eine gewisse Zeit benötigt, entsteht ein toter Winkel, in dem der Laserscanner keine Information liefert.

$$d = \frac{1}{2} \times \left\{ \phi / \left(\frac{f}{c} \times 2\pi \right) \right\}, \tag{2.3}$$

herbei ist d die Distanz zu dem Hindernis, f die Frequenz, ϕ die Phasendifferenz sowie c die Lichtgeschwindigkeit.

Damit diese Gleichung bei der Verwendung mehrerer unterschiedlicher Frequenzen bestimmt werden kann, soll im Preprocessing-Schritt die gemeinsame Phasendifferenz

ermittelt werden. Im Fall von zwei verwendeten Frequenzen kann folgende mathematische Gleichung (2.4) genutzt werden:

$$\phi = arctan\left(\frac{\sum(s_n \times cos(2\pi/n)}{\sum(s_n \times sin(2\pi/n)}\right), \tag{2.4}$$

hierbei ist ϕ die gesuchte gemeinsame Phasendifferenz, n die Anzahl einzelner Impulse und s_n der Wert eines einzelnen Impulses an der Stelle n.

Der Vorteil der Lichtlaufzeitmessung liegt in der Möglichkeit, alle durch den Strahl erreichbaren Distanzen zu messen. Für kurze Distanzen ist die Messung dagegen ungenau oder gar unmöglich. Dagegen liefern die Phasendifferenzverfahren genauere Ergebnisse für kurze Distanzen, scheitert aber bei den längeren aufgrund der Symmetrie und der daraus folgenden Mehrdeutigkeit der verwendeten Sinus- und Cosinus-Funktionen. Eine Kombination der beiden Verfahren kompensiert die oben genannten Nachteile. Dabei wird für die längeren Distanzen die Lichtlaufzeit genutzt, danach wird das Ergebnis durch das Phasendifferenzverfahren optimiert und verbessert.

Im Vergleich zu den Stereokamerasystemen liefern die Laserscanner genauere Tiefeninformationen, sie sind robuster und weniger abhängig von den Lichtverhältnissen. Dennoch haben auch Laserscanner ihre Nachteile: Dunkle, besonderes schwarze sowie stark reflektierenden Oberflächen können zu Fehler führen. Im Weiteren benötigen die Laserscanner eine Aufwärmphase, bis sie stabile, weniger verrauschte Tiefeninformationen liefern. Der größte Nachteil liegt aber darin, dass die Laserscanner 3-D-Informationen nur für eine 2-D-Linie der Umgebung liefern. Daher beschäftigt sich der nächste Abschnitt mit der möglichen Erweiterung eines 2-D-Laserscanners zu einer 3-D-Umgebungserfassung.

2.7. Erweiterung des Erfassungsbereichs auf 3-D

Es gab eine Menge unterschiedlicher Versuche, den Erfassungsbereich der Laserscanner auf 3-D zu erweitern. Natürlich existieren 3-D-Laserscanner, diese sind aber immer noch groß, schwer und extrem teuer.

Die Grundidee der 3-D-Erweiterung ist die Verwendung eines Mechanismus zur Bewegung des Laserscanners und eine daraus resultierende Erweiterung der Messergebnisse. Zum Beispiel wird in der vorliegenden Arbeit eine Schwenk-Neige-Einheit genutzt (vgl. Kapitel 3.5.1). Grundsätzlich kann das „aktive Sehen“durch einen derartigen Aufbau am besten erreicht werden. Dieses neu entwickelte System kann problemlos durch weitere Sensoren erweitert werden.

Abbildung 2.17 stellt einen Entwurf und einen der ersten Realisierungsversuche eines aktiven Wahrnehmungssystems (Active Perception Stereo Head) dar. Diese Systeme liefern eine große Menge an Daten und sind einfach auf einem Roboter montierbar. Durch eine Kalibrierung können alle anfallenden Daten in ein Koordinatensystem transformiert

Abbildung 2.17.: Entwurf und einer der ersten Realisierungsversuche eines Active Perception Stereo Head.

und damit in Relation zueinander verwendet werden. Der Nachteil solcher Systeme liegt allerdings in der langen Datenakquisitionszeit, damit sind sie für eine dynamische Umwelt nur bedingt geeignet.

2.8. Vergleich unterschiedlicher Sensorarten

Im Folgenden sollen die verschiedenen Sensorarten hinsichtlich ihrer Eignung für die Objektdetektion und Objekterkennung verglichen werden. Dabei wird versucht, möglichst viele unterschiedliche Modalitäten abzudecken. Des Weiteren sollen Vor- und Nachteile des jeweiligen Sensors verdeutlicht werden.

Darüber hinaus sind für diese Arbeit die Problembereiche einzelner Sensoren interessant. Bei den Kameras und beim Stereokamerasystem besteht immer eine Abhängigkeit von den wechselnden Lichtverhältnissen. Ändern sich diese, muss der Sensor beziehungsweise ein Algorithmus an die entsprechenden Bedingungen angepasst werden. Bei der Berechnung der Tiefeninformation mit einem Stereokamerasystem kommt noch zusätzlich die Problematik der homogenen Oberflächen hinzu, da auf diesen keine korrespondierenden Punkte akkurat gefunden werden können.

Laserscanner benötigen für präzise Messergebnisse eine sogenannte „warm-up“Zeit, die bis zu zwei Stunden andauern kann. Während dieser Zeit liefert der Laserscanner eine überdurchschnittliche Anzahl falscher Werte. Des Weiteren entstehen größere Fehler, die schon im Zentimeter-Bereich auftreten können, bei der Reflexion an dunklen oder stark reflektierenden Oberflächen sowie an Objekten, die in einem 45° Winkel zum Sensor platziert sind. Für den weiterführenden qualitativen Vergleich sei der Leser auf [LE07] verwiesen. Den größten Nachteil des simulierten 3-D-Laserscanners stellt natürlich die Datenakquirierungszeit dar, aus diesem Grund kann der Sensor nur bedingt in dynamischen Umgebungen eingesetzt werden.

Sensor	3-D acc.	Acq. time	Res.	RGB	Depth	Grayscale	Thermal
Camera	—	fast	high	×	—	×	—
SKS	low	middle	high	×	×	×	—
LRF	high	fast	low	—	×, only 2-D	—	—
3-D-LRF	high	slow	high	—	×	—	—
PrimeSense	middle	fast	high	×	×	×	—
TOF	high	fast	low	×	×	×	—
IR camera	—	middle	high	—	—	—	×
Sonar	middle	fast	low	—	×	—	—

Tabelle 2.1.: Vergleich unterschiedlicher Sensorarten

In der Tabelle 2.1 steht SKS als Akronym für das Stereokamerasystem, LRF für den Laserscanner (engl. für „laser range finder“), 3-D-LRF für den oben beschriebenen simulierten 3-D-Laserscanner und PrimeSense für alle Sensoren, die auf der von PrimeSense entwickelten Technologie basieren. Die Begriffe wie „high“, „low “, „fast“, „middle“sind selbsterklärend, × steht für vorhandene und − für nicht vorhandene Daten in Bezug auf den jeweiligen Sensor.

Ein aus mehreren Sensoren kombiniertes System erbt natürlich die Nachteile aller eingesetzten Sensoren. Es bietet aber durch die Fusion von redundanter und nicht redundanter Sensorinformation große Möglichkeiten zur Rauschreduktion, zur qualitativen Verbesserung der Daten sowie zur Gewinnung neuer Informationen. Diese entstehen durch die Bildung von Hypothesen anhand der Kombination der vorliegenden Daten. Des Weiteren bietet ein solches System ein großes Spektrum vielfältiger Anpassungsmöglichkeiten an die gestellte Aufgabe, zum Beispiel durch die Auswahl der Sensoren und bestimmter Sensorinformationen durch die am besten geeigneten Algorithmen. Somit stellen solche Wahrnehmungssysteme ein mächtiges, universelles Werkzeug dar, mit dem viele Aufgaben besser gelöst werden können als durch die Verwendung einzelner Sensoren.

Kapitel 3

Intelligente Multi-Sensor-Fusion

Normalerweise werden in der Robotik mehrere unterschiedliche Arten von Sensoren und ein oder zwei Manipulatoren auf einer mobilen Plattform integriert. Damit die Daten einzelner Sensoren sowie Manipulatoren genutzt werden können, sollen alle einzelnen Frames (etablierter Begriff für ein Koordinatensystem in der Robotik) in einen Frame transformiert werden. Dieser Vorgang wird meistens als Registrierung bezeichnet [BM02]. Dabei wird die Plattform kalibriert, und alle Frames werden in ein zentrales Roboterframe transformiert. Des Weiteren wird durch die Navigation gewährleistet, dass zu jedem Zeitpunkt eine gültige Transformation zwischen dem zentralen Roboterframe und der Welt (in den meisten Fällen eine 2-D-/2,5-D-/3-D-Karte) existiert. Die Problematik der Transformation zwischen dem kartesischen Koordinatensystemen und roboterinternen Koordinatensystemen ist nicht trivial. Dennoch ist sie, ähnlich wie die Transformation zwischen direkter und inverser Kinematik, ein notwendiges Instrument.

Als Beispiel kann ein Szenario dienen, in dem der Roboter einem Gast in einem Restaurant einen Kaffee serviert. In dieser Situation lokalisiert sich der Roboter in einer Karte (Welt) und bekommt die Anweisung, den Kaffee von einem Tisch in der Küche aufzunehmen und auf einem Tisch vor dem Gast abzustellen. Dabei darf keine Kollision verursacht werden. Bei der Kollisionsvermeidung/Pfadplanung werden aus den Gelenkwinkeln und der Robotergeometrie die Form bestimmt und mit einer 3-D-Karte verglichen. Für dynamische Hindernisse werden die Daten der Sensoren hinzugezogen. Steht der Roboter vor dem Tisch, sollen die Sensoren in der Welt so ausgerichtet werden, dass der Tisch möglichst komplett erfasst wird. Die Umgebung wird in Sensorframes wahrgenommen und analysiert, um jedoch ein Objekt greifen zu können, wird unter anderem die Transformation in ein Manipulatorframe benötigt. Es ist nicht schwierig, sich vorzustellen, dass bei der Ausführung komplexerer Szenarien eine ständige Transformation zwischen unterschiedlichen dynamischen und statischen Koordinatensystemen erfolgt. Einige der wichtigsten Kriterien hierbei sind die Aktualität und die Genauigkeit

der Transformationen; schlägt ein Schritt fehl, kann die Aufgabe sehr oft nicht mehr erfolgreich abgeschlossen werden.

Dieses Kapitel beschäftigt sich mit der Multi-Sensor-Fusion. Dabei sollen die Möglichkeiten aufgezeigt werden, wie durch die Fusion unterschiedlicher Sensordaten mehr Informationen über invariante Objekteigenschaften gewonnen werden können, als es mit den Daten einzelner Sensoren möglich gewesen wäre. Die Multi-Sensor-Fusion ist somit eine notwendige Voraussetzung für die vorliegende Arbeit. Dabei wird zuerst auf die theoretischen Grundlagen eingegangen. Danach wird der Stand der Technik vorgestellt und anschließend auf die Sensor-Fusion in der mobilen Robotik diskutiert. Des Weiteren wird das Konzept des aktiven Sehens vorgestellt. Das Kapitel wird mit der Präsentation zweier eigens entwickelter Verfahren abgeschlossen: eins für die Kombination eines Stereokamerasystems mit einem 2-D-Laserscanner montiert auf einer Schwenk-Neige-Einheit und ein zweites für die Fusion der Daten eines PrimeSense-Sensors.

3.1. Sensor-Fusion

Die zuverlässige Erkennung von 3-D-Objekten anhand von 2-D-Kamerabildern ist immer noch eine große Herausforderung in der Bildverarbeitung. Grundsätzlich kann ein Objekt über die Form, Größe, Farbe, Bewegung oder Vorhersagen erkannt werden. Dennoch gilt für beliebige 3-D-Szenarien, dass ein Objekt durch einen Sensor nicht von allen Seiten erfasst werden kann und damit nie alle notwendigen Daten für eine exakte 3-D-Rekonstruktion eines 3-D-Modells vorhanden sind. Ein weiteres Problem ist der Verlust der Tiefeninformation bei der Abbildung der 3-D-Umgebung auf einer 2-D-Ebene. Das Blockdiagramm in der Abbildung 3.1 stellt den mittlerweile allgemein akzeptierten Ablauf der Bildverarbeitung grafisch dar.

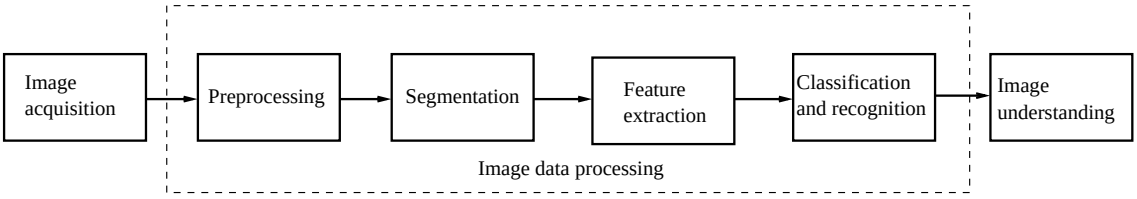


Abbildung 3.1.: Standardisierter offener Steuerkreis der Bildverarbeitung.

Bei der Betrachtung des Diagramms stellt sich die Frage, ob die Tiefeninformation eine primitive Objekteigenschaft ist, wie zum Beispiel die Form oder die Farbe. Dies wird mittlerweile von vielen Forschern angenommen [SM00][MNE00].

Angelehnt an die Natur, kam in der Informatik schon sehr früh der Gedanke auf, Daten mehrerer gleicher oder unterschiedlicher Sensoren gemeinsam auszuwerten, mit dem Ziel, so viele Informationen wie möglich über eine Szene zu erhalten. Dieses Vorgehen er-

laubt es, fehlerbehaftete Sensordaten sowie einen kompletten Ausfall eines oder mehrerer Sensoren zu kompensieren. Besonders in technischen Anwendungen spielt die Integration unterschiedlicher Sensoren eine immer größere Rolle. Durch unterschiedliche Algorithmen und Verfahren werden die Daten verschiedener Sensoren verrechnet. Meistens sind diese Verfahren konzeptuell einfach und können mit geringerem Aufwand implementiert werden. Außerdem benötigen solche Ansätze nur minimales Apriori-Wissen über das zu lösende Problem.

Es existieren vier Arten der Sensor- bzw. Datenfusion [FCR04]:

- **Komplementär:** Sensoren decken verschiedene Bereich ab.
- **Kompetitiv** bzw. **redundant:** Gleichwertige Informationen stehen im Wettbewerb miteinander.
- **Kooperativ:** Daten werden gemeinsam erzeugt, die einzeln unerreichbar sind.
- **Unabhängig:** Daten von nicht in Relation stehenden Sensoren, werden fusioniert.

Ähnlich wie bei den Vorgängen der Wahrnehmung werden mehrere Ebenen der Fusion unterschieden:

- **Frühe Ebene:** Daten werden weitergereicht und es gibt eine gemeinsame Auswertung.
- **Späte Ebene:** Nur einzelne, ausgewählte Ergebnisse werden weitergereicht.
- **Probabilistische Ebene:** Die Wahrscheinlichkeitsverteilung wird weitergereicht.
- **Selbstorganisierte Ebene:** Daten werden automatisch bestmöglich zusammengefasst, um dynamisch auf die Umgebung reagieren zu können.

Auch im Bereich der mobilen Robotik, sei es durch den Einbau von mehr Sensoren oder durch die steigende Rechnerleistung zur Verarbeitung der Sensorinformationen, steigt die Bedeutung der Sensor-Fusion ständig.

3.2. Stand der Technik

Multi-Sensor-Fusion ist ein bekanntes Thema der Informatik. Seit mehreren Jahren werden die Algorithmen in vielen Bereichen, wie zum Beispiel in der Topografie, der Robotik, der Umweltrekonstruktion und bei der Bildung virtueller Welten erfolgreich eingesetzt. Die zur Fusion verwendeten Sensoren werden meistens an ein Szenario angepasst. Im Folgenden werden einige signifikante Beispiele vorgestellt und diese kurz beschrieben.

Heng et. al. präsentieren in ihrer Veröffentlichung [HKS97] eine Methode zur Kollisionsvermeidung, basierend auf der Fusion von Kamera und Sonarsensoren für mobile

Roboter. Die Objekte werden über die Kamerabilder erkannt, danach wird die Kantendetektion angewandt und durch die Daten der Sonarsensoren ergänzt. Die Fusion basiert auf der Position der Objekte im Bild und den Sonardaten. Die Ergebnisse sind besser als bei der Verwendung von nur einem der Sensoren.

Mit dem gleichen Thema, der Kollisionsvermeidung, befassten sich auch Jacobs et al. in [JFS+10]. Der so optimierte Pfad wird als initiale Lösung zur Kollisionsvermeidung genutzt. Die hindernisfreie Region vor dem Roboter wird trianguliert und für die Berechnung des nächsten Zielpunktes genutzt. Diese Methode ist auf Schnelligkeit ausgelegt, damit sie in RoboCup-Szenarien genutzt werden kann.

Die Fusion mehrerer Sensoren für Simultaneous Localization and Mapping (SLAM) ist weit verbreitet. Zum Beispiel nutzen Tokekar et. al. [TBFP09] die Fusion mehrerer Laserscanner nicht nur für SLAM, sondern auch für die Objektrekonstruktion.

Nach der Entwicklung der 3-D-Laserscanner, wobei es sich häufig um zwei in einem Gehäuse vereinigte, senkrecht zueinander montierte, bewegliche 2-D-Laserscanner handelt, wird in der Geodäsie und der Archäologie oft die Fusion von Punktwolken mit einem Bild genutzt. Dies soll dem Menschen helfen, sich besser innerhalb der fusionierten Daten zu orientieren. Die Fusion basiert meistens, wie in dem Artikel von Schneider et. al. [uHGM07], auf den akkuraten geometrischen Modellen beteiligter Sensoren.

Es wird deutlich, dass in den meisten Fällen versucht wird, die heterogenen Sensorinformationen zu fusionieren. Das Interesse daran, für ein Pixel nicht nur Textur- und Farb-, sondern auch Tiefeninformationen zu gewinnen, ist sehr groß. Auch Nicolas Burrus präsentiert auf seiner Internetseite[1] ein Verfahren zur Fusion von Farb- und Tiefeninformation mithilfe eines PrimeSense-Sensors. Speziell für die Objekterkennung ist die Multi-Sensor-Fusion besonders interessant, da dadurch alle fünf in Kapitel 3.1 beschriebenen Eigenschaft eines Objekts genutzt werden können. Bereits in den 1990er-Jahren wurde die Multi-Sensor-Fusion erforscht und in [BH96] veröffentlicht.

3.3. Sensorfusion in der mobilen Robotik

Sensor-Fusion wird zunehmend als eine wichtige perzeptuelle Tätigkeit in der mobilen Robotik angesehen. Robin R. Murphy untersuchte grundlegende Aspekte der Sensor-Fusion. In ihrer Veröffentlichung von 1996 stellt sie die biologischen und kognitiven Verfahren der intelligenten Sensor-Fusion gegenüber [Mur96].

Der biologische Ansatz beschäftigt sich mit der Frage, wie die Sensor-Fusion funktioniert. Schon 1978 beschrieb L. E. Marks „die Einigkeit der Sinne„, basierend auf von ihm durchgeführten psychologischen Studien [Mar78]. Er stellt fest, dass erst die Kombination der Sinne eine angemessene Wahrnehmung eines Objekts oder einer Situation ermöglicht. Seine Theorie der Korrespondenz von Sensoren basiert auf fünf Doktrinen,

[1] http://nicolas.burrus.name/index.php/Research/KinectCalibration

von denen jede experimentell bestätigt worden ist. Da diese für die vorliegende Arbeit nicht relevant sind, werden sie nicht ausführlich erläutert.

Weiterführend zu Marks präsentieren Stein und Meredith in ihrer Veröffentlichung ein neurologisches Modell der Sensor-Fusion basierend auf Studien zum Gehirn von Katzen [SM93]. Dieses Modell beschreibt, die besonders für die Robotik relevanten Aspekte der Sensor-Fusion.

Sensor-Fusion

- verbindet die Wahrnehmung mit der Aktion,
- bezieht die Kontextinformation mit ein,
- kombiniert Sensoren auf unterschiedliche Art und Weise für verschiedene Wahrnehmungen.

Die Autoren beschreiben in ihrer Studie eine Art Agentensystem, wobei jeder Sensor einen Agenten darstellt. Jeder Sensor hat eine eigene neuronale Repräsentation, die vom eigenen Wahrnehmungsbereich und der Sensitivität abhängt. Die Sensor-Fusion hat die Aufgabe, sich der Umgebung und dem Zustand der Agenten anzupassen.

Während der biologische Ansatz sich mit der „Wie“-Frage beschäftigt, geht der kognitive Ansatz der Frage nach, warum die Sensor-Fusion ein zentraler Teil der Wahrnehmung ist. Pick und Saltzmann generalisieren die Trennung von Marks. Nach Meinung der Autoren hängt die Wahrnehmung von der aktuellen Aufgabe ab [PS78]. Besonders sichtbar wird dies am Beispiel der Navigation in einem Raum im Kontext der Suche nach einem Schlüssel.

Bower argumentiert, dass die Informationen von unterschiedlichen Sensoren üblicherweise übereinstimmend ist [Bow74]. Es werden verschiedenen Daten auf eine Stelle unserer Umgebung projiziert, zum Beispiel durch das Hören und Sehen eines Objekts. Diese Übereinstimmung ist eine Eigenschaft unserer Umwelt und macht die Sensor-Fusion überhaupt erst sinnvoll. Durch die Ungenauigkeit der Sensoren oder gezielte experimentelle Manipulationen kann diese Übereinstimmung zerstört werden. Bower unterscheidet vier „Stufen der Einigkeit“(engl.: „level of unity“) des Wahrgenommenen.

- **Vollständige Einheit:** Es existiert ein Kompromiss, der durch das System ermittelt werden kann.
- **Unterdrückung fehlerhafter Sensoren:** Fehlerhaften Sensoren können erkannt und deren Information kann gezielt unterdrückt werden.
- **Rekalibrierung fehlerhafter Sensoren:** Fehlerhaften Sensoren können erkannt und rekalibriert werden.

- **Keine Einigkeit:** Es kann kein Kompromiss erreicht und damit keine Integration vollzogen werden.

Für alle vier Stufen können Beispiele im Verhalten von Menschen oder Tieren gefunden werden.

Lee beschreibt in seiner Arbeit [Lee78] zwei Typen der orientierten Aktivität bei Tieren, die Ermittlung der für eine Aufgabe relevanten perzeptuellen Information und die Ausführung derselben. Das Konzept des Autors für die mobile Robotik unterteilt die Sensor-Fusion in zwei Phasen. In der ersten erforschenden Phase (oder Anlaufphase) wird die Fusion vorbereitet. Dabei spielen die Orientierung der Sensoren sowie die relevanten Eingangsdaten eine Rolle. In dieser Phase werden die notwendigen Sensoren in Abhängigkeit von der Aufgabe ausgewählt, und ihre Position, Orientierung und Sensitivität werden kalibriert. In der zweiten Phase findet die eigentliche Wahrnehmung statt. Die beiden Phasen kommunizieren miteinander, um die bestmöglichen Ergebnisse zu erzielen.

Am Ende dieses Unterkapitels bleibt nur festzustellen, dass keine Theorie der Sensor-Fusion existiert, die hinreichend ist, um die menschliche Wahrnehmung nachzubilden. Erst die Zusammenfügung mehrerer unterschiedlicher Verfahren und Lernmethoden kann die Forschung voranbringen, damit die menschliche Wahrnehmung besser verstanden werden kann, vgl. Appendix B. Eine, nach Meinung des Autors, vielversprechende Vorgehensweise zur Nachahmung der menschlichen Wahrnehmung stellt das bereits erwähnte Konzept des „aktiven Sehens“dar, auf das im nächsten Abschnitt genauer eingegangen wird.

3.4. Aktives Sehen

In der vorliegenden Arbeit wird die Umgebung, eingeschränkt auf die ROIs, aktiv wahrgenommen (vgl. Konzept des aktiven Sehens [Mer99]). Der Ausgangspunkt des aktiven Sehens ist die Tatsache, dass die Sensordaten eine hohe Redundanz aufweisen. Die Verarbeitung der anfallenden Daten ist sehr aufwendig, wie am Beispiel der menschlichen Wahrnehmung am besten demonstriert werden kann. Der visuelle Kortex, der für die Verarbeitung von visuellen Informationen verantwortlich ist, beansprucht einen großen Teil unseres Gehirns, und die gemessene Aktivität bei der Auswertung der Bildinformationen ist sehr hoch. Daher schlägt Frau Mertsching in ihrer Publikation vor, die Aufmerksamkeitsmechanismen in aktiven Sehsystemen bevorzugt zu betrachten. Diese Mechanismen erlauben eine kontextsensitive Figur-Hintergrund-Trennung und eine Realisierung von Realzeit-Verhalten durch eine Serialisierung des Datenstroms. Dabei wird unter dem Begriff der „Aufmerksamkeit“eine Fokussierung auf einen relativ kleinen Bereich der wahrgenommenen Umwelt verstanden. Dabei wird die Umgebung erfasst, interessante Bereiche werden selektiert und erneut, meist mit einer deutlich größeren Informationsdichte, wahrgenommen. Die Verwendung von einer höheren Auflösung in

den ausgewählten Bereichen kann dabei als foveales Sehfeld betrachtet werden. Die Bereiche, die mit einer niedrigen Dichte wahrgenommen werden, werden als Peripherie des Blickfeldes gekennzeichnet. Eine der wesentlichen Voraussetzungen für das aktive Sehen ist eine Blicksteuerung, die meistens, wie in Kapitel 2 beschrieben, durch eine Schwenk-Neige-Einheit (PTU, engl. für „pan-tilt unit“) realisiert wird.

In der vorliegenden Arbeit werden, wie bereits beschrieben, planare, parallel zum Boden verlaufende Oberflächen als ROIs betrachtet. Nachdem diese selektiert wurden, werden die Sensoren entsprechend positioniert und die Oberflächen mit einer höheren Punktdichte wahrgenommen. Damit an diesen Stellen ein maximaler Informationsgehalt gewonnen werden kann, sollen die Sensordaten unterschiedlicher Sensorarten intelligent fusioniert werden. Das Ziel ist es, die Tiefeninformationen mit den Farbinformationen zu vereinen, was in Form einer kolorierten Punktwolke (RGB-D, RGB steht hierbei für die Farb- und D für die Tiefeninformation) realisiert werden kann. Wird die Oberfläche über einen Zeitraum wahrgenommen, stehen genügend Informationen zur Verfügung, damit die Objekte sowie deren Position und Orientierung über Form, Größe, Farbe und Bewegung erkannt werden können.

Im nachfolgenden Abschnitt werden unterschiedliche Möglichkeiten der Verwendung der intelligenten Multi-Sensor-Fusion genauer betrachtet und diskutiert.

3.5. Verwendete Verfahren

In diesem Abschnitt werden einige selbst entwickelte Verfahren der Multi-Sensor-Fusion beschrieben und ausführlich diskutiert. Den Anfang macht eine Kombination aus einem Laserscanner, einer Schwenk-Neige-Einheit sowie einem Stereokamerasystem. Dieser eigens entwickelte Algorithmus ist Bestandteil eines komplexeren Systems und wurde im Rahmen einer IEEE-Konferenz bereits veröffentlicht [KZ11]. Ein weiterer Abschnitt beschreibt die Kalibrierung und die Multi-Sensor-Fusion von PrimeSense-Sensoren, die in der vorliegenden Arbeit vorwiegend Verwendung findet.

3.5.1. Kombination eines Laserscanner mit einer Schwenk-Neige-Einheit und einem Stereokamerasystem

Wie bereits im vorangegangenen Abschnitt dargestellt, ist es das Ziel jeder Sensor-Fusion, mehr Informationen zu gewinnen, als es aus den einzelnen Sensordaten möglich ist. Lange Zeit lieferten nur Stereokamerasysteme brauchbare und vor allem erschwingliche Informationen über die 3-D-Umgebung. Später kamen Laserscanner hinzu, die akkurater und schneller arbeiten, jedoch nur 2-D-Informationen liefern. Da die Preise für 3-D-Laserscanner immer noch sehr hoch und die Geräte groß und schwer sind, findet diese Art von Sensoren kaum Verwendung in der Robotik. Daher wurden schon sehr früh Versuche unternommen, beide Sensorarten, Stereokamerasysteme sowie 2-D-Laserscanner zu fusionieren.

Es entstanden mehrere Systeme, die auf der Kombination aus einem Laserscanner und einer Schwenk-Neige-Einheit (PTU, engl. für „pan-tilt unit“) basieren und damit 3-D-fähig sind. Zwei Beispiele dafür sind die Plattformen, die am Arbeitsbereich TAMS der Universität Hamburg entworfen und realisiert wurden. Diese sind in den Abbildungen D.9 sowie 3.2 grafisch dargestellt.

Die Plattform, die in der Abbildung 3.2 links dargestellt ist, basiert auf einer PTU der Firma Active Media Perception, einem Hokuyo-URG-04LX-Laserscanner sowie einem Stereokamerasystem. Das Stereokamerasystem besteht hierbei aus zwei Sony XC-999P mit PAL-Auflösung und ist über einen Framegrabber angeschlossen. Die Reichweite des Laserscanners wird vom Hersteller mit maximal 5,6 m angegeben und beträgt tatsächlich, gemessen an der Genauigkeit der gelieferten Tiefenwerte, ca. 4 m.

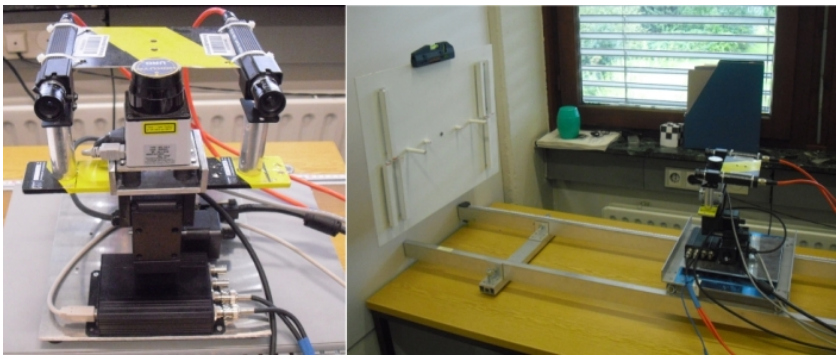
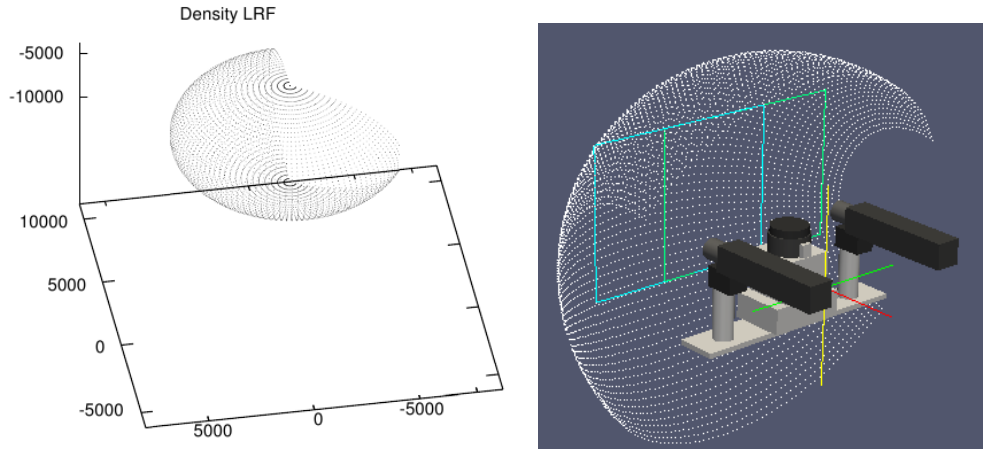


Abbildung 3.2.: Plattform zur Sensor-Fusion des Arbeitsbereichs TAMS, basierend auf einer PTU, einem Hokuyo URG-04LX und einem Stereokamerasystem (links), und der Kalibrierungsaufbau (rechts).

Der Nachteil solcher Systeme ist die lange Datenakquirierungszeit, was die Anwendung in dynamischen Szenen praktisch unmöglich macht. Die Genauigkeit hängt ab von der Genauigkeit einzelner Komponenten sowie der Registrierung einzelner Sensorframes zueinander. Die Punktdichte hängt von der Art der Montage auf der PTU ab: An der Drehachse des Laserscanners ist die Dichte am höchsten, zur Mitte der Achse hin nimmt die Punktdichte kontinuierlich ab. Des Weiteren ist die Punktdichte umgekehrt proportional zur Größe der Schritte der PTU bei der Umgebungserfassung. Die Punktdichte der vorgestellten Plattform sowie der Fusion mit dem Stereokamerasystem ist in der Abbildung 3.3 dargestellt. Da die Schrittgröße variabel verändert werden kann, ist es möglich, das System dynamisch an die zu scannende Umgebung anzupassen.

Für die weiteren Verarbeitungsschritte wird eine registrierte Punktwolke benötigt. Registriert bedeutet in diesem Zusammenhang, dass einzelne 3-D-Punkte in ein und dasselbe Koordinatensystem projiziert werden. Dafür wird eine homogene Transformation benötigt, die abgesehen vom Winkel der PTU über die gesamte Zeit konstant bleibt,

soweit der Abstand zwischen einzelnen Laserstrahlen unverändert gehalten wird. Wird zusätzlich die Farbinformation benötigt, eine sogenannte RGB-D-Punktwolke, muss eine weitere Transformation bestimmt werden, die die einzelnen 2-D-Farbkoordinaten des Bildes auf die 3-D-Koordinaten der Punktwolke abbildet. Um beide notwendigen Transformationen bestimmen zu können, werden mehrere korrespondierende Paare benötigt, ähnlich wie bei den Prozessen der standardisierten Bildverarbeitung. Solche Prozesse sind unter dem Begriff extrinsische Sensorkalibrierung bekannt.



(a) (b)

Abbildung 3.3.: Punktdichte des simulierten 3-D-Laserscanners, basierend auf dem 2-D-Laserscanner und der PTU sowie der Kombination mit einem Stereokamerasystem.

Die Vorteile einer Datenfusion aus einem simulierten 3-D-Laserscanner sowie einem Stereokamerasystem werden deutlich, wenn die Problembereiche einzelner Sensoren betrachtet werden. Bei der **Kamera** sind dies

- homogene Oberflächen,
- starke Reflexionen,
- variierende Lichtbedingungen.

Bei einem **Laserscanner** sind es

- Fehlwerte (während und nach der Aufwärmphase),
- schwarze Oberflächen,

- Objektorientierung gegenüber dem auftreffenden Laserstrahl,
- dünne Objekte (der Laserstrahl ist breiter als das Objekt),
- starke Reflexionen.

Es ist ersichtlich, dass die Fusion der beiden Sensoren ein besseres Ergebnis als die Daten der einzelnen Sensoren ergibt. Dennoch bleibt die Verdeckung selbstverständlich ein Problembereich beider Sensoren.

Eine der essenziellen Annahmen dieser Arbeit ist die Verwendung des selben Modells für die 3-D-Punktwolke wie auch für ein 2-D-Bild. Eine 2-D-Abbildung I kann als Summe folgender einzelner Komponenten gesehen werden:

$$F_{i,f}(\overrightarrow{x}) = \alpha \cdot BG_{i,f}(\overrightarrow{x}) + N_{i,f}(\overrightarrow{x}) + T_{i,f}(\overrightarrow{x}) \tag{3.1}$$

oder vereinfacht

$$F_i = T_i + BG_i + N_i \tag{3.2}$$

hierbei stellt F eine Abbildung, T_i den Vordergrund, BG_i den Hintergrund und N_i das Sensorrauschen an der Framenummer i dar. Der Vorteil dieser Annahme ist einerseits die Erhaltung der Nachbarschaftsbeziehungen und andererseits die Möglichkeit, bekannte Algorithmen und Methoden der 2-D-Bildverarbeitung auch auf die 3-D-Punktwolken anwenden zu können. Dieses Konzept wurde in eigens durchgeführten industriellen Projekten bereits erfolgreich eingesetzt und mehrfach veröffentlicht [KFBZ10] [KFBZ11]. Wie bereits beschrieben, werden nur ROIs mit der vorgestellten Plattform erfasst. Dieses wird durch die Erfassung und Vorsegmentierung der Umgebung durch einen rotierenden Laserscanner ermöglicht. Eine weitere Voraussetzung bei der Nutzung eines simulierten 3-D-Laserscanner ist, dass die zu scannende Szene über die Scanzeit statisch bleibt.

Beide Sensorarten weisen unterschiedliche Charakteristiken sowie Wahrnehmungsbereiche auf. Abbildung 3.4 stellt exemplarisch die Wahrnehmungs- sowie Überlappungsbereiche einer Kamera und eines Laserscanners dar. Die gesuchte Transformation zwischen den beiden Sensoren basiert auf deren extrinsischen Parametern und ist von der Beschaffenheit einer Szene sowie von der Entfernung zu den Objekten unabhängig. Zuerst werden die Kameras einzeln und dann als Stereokamerasystem kalibriert. Das Verfahren wurde vom Autor während seiner Diplomarbeit implementiert und es basiert auf der Kalibrierungsmethode nach Z. Zhang [Zha98] [Zha00] und der Rektifikation nach der Methode von Fusiello, Trucco und Verri [FTV00]. Dieses Verfahren wurde im Rahmen einer IEEE-Konferenz veröffentlicht [KSJZ09b]. Der Kalibrierungsprozess transformiert die Bilder einer Kamera in das Koordinatensystem der anderen und berechnet die Disparität. Damit müssen für die Kalibrierung des Stereokamerasystems und des Laserscanners die Bilder von nur einer Kamera in das Koordinatensystem des simulierten 3-D-Laserscanners transformiert werden, wie in der Abbildung 3.4 dargestellt.

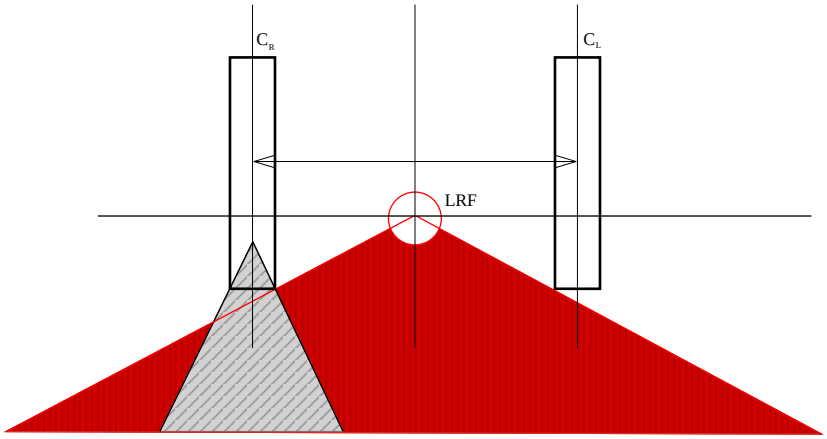
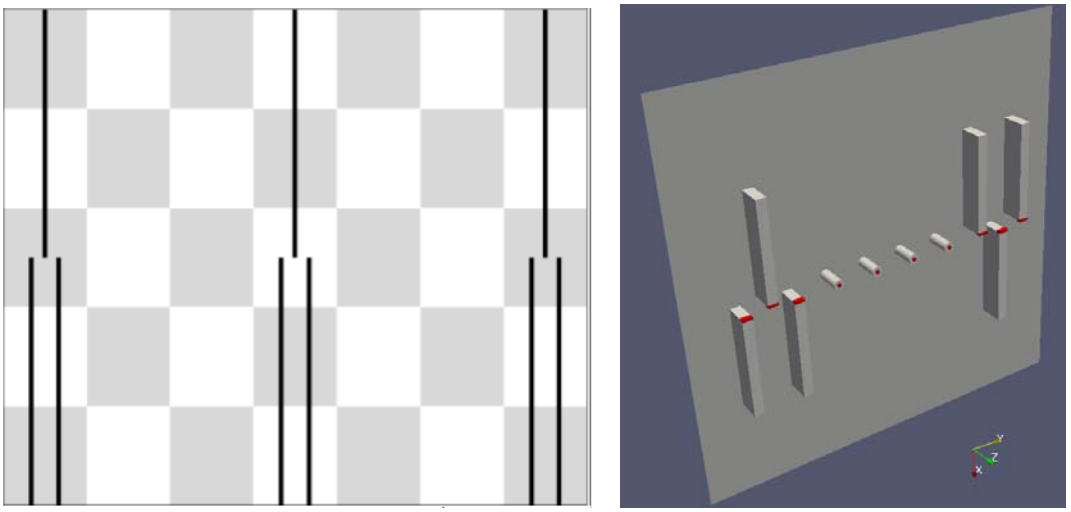


Abbildung 3.4.: Grafische Darstellung des Wahrnehmungs- und Überlappungsbereichs einer Kamera und eines Laserscanners.



(a) (b)

Abbildung 3.5.: Geplanter Kalibrierungskörper (links) und das Modell des resultierenden 3-D-Kalibrierungskörpers, das durch einige Experimente angepasst worden ist (rechts).

Die ursprüngliche Idee war eine Kombination aus den typischen Eigenschaften der Kalibrierungskörper einer Kamera (ein Schachbrettmuster) und einem Laserscanner. Anfänglich wurde ein 3-D-Körper erweitert durch ein 2-D-Schachbrettmuster genutzt,

siehe Abbildung 3.5(a). Da, wie bereits erwähnt, ein Laserscanner in den schwarzen Bereichen zu Fehlern neigt, wurde der Kalibrierungskörper neu gestaltet. Der verwendete Kalibrierungskörper basiert auf einer planaren Fläche, zwei einander gegenüberliegenden, sogenannten Y-Strukturen und einigen Stiften von unterschiedlicher Länge. Die Y-Strukturen stellen eine ideale Form für den Laserscanner dar. Deren Oberflächen sowie die der Stifte werden für die Auffindung von korrespondierenden Punkten in den Laserscannerdaten genutzt. Für das Stereokamerasystem wird auf die Farbinformation zurückgegriffen. Dafür wurden die Y-Strukturen durch rote Striche und die Stifte durch die Punkte derselben Farbe markiert. Das Modell des resultierenden Kalibrierungskörpers ist in der Abbildung 3.5(b) dargestellt. Die Abbildung 3.2 rechts zeigt die seitliche Ansicht des Kalibrierungskörpers sowie den verwendeten Kalibrierungsaufbau. Der Pseudocode unseres Verfahrens ist in dem Algorithmus 1 zusammengefasst. Weiterführende Informationen sowie die Implementierungsdetails können unserer Publikation [KHZ10] entnommen werden.

Algorithm 1 The sensor fusion algorithm

1: **procedure** FUSION(LRF, SCS)
2: The laser scanner is moved by the pan-tilt unit and scans the environment in coarse steps.
3: Through changes in the depth information, our system localizes the region of interest (ROI) and the changeover from one to two teeth (or vice versa) in the Y-structure and rescans it in finer steps to find the desired position.
4: Detection the characteristic pattern for the correct position of the laser scanner in relation to the calibration body and moves the pan-tilt unit to that position.
5: Stop the platform and acquire a camera image.
6: Applying the color threshold value to the camera image.
7: Calculating of corresponded points.
8: Rectification of both lines to each other (laser scanner and an image).
9: Calculation of transformation matrix between a laser range finder and a camera (is independent of scene structure).
10: Calculation of the overlapping area with help of known geometrical parameters and computed fundamental matrix (the transformation matrix can be used in the overlapping area only, otherwise the transformation would cause an error and produce wrong correspondences).
11: Adopt the matrix to the point cloud and an image.
12: **return** partially colored point cloud
13: **end procedure**

Nach der Berechnung der Parameter, des Überlappungsbereichs und der Bestimmung der Transformation kann die adaptive Multi-Sensor-Fusion vollzogen werden.

Es wurden einige bekannte Kalibrierungsmethoden getestet. Die besten und detaillier-

testen Ergebnisse wurden aber mit dem oben genannten Verfahren erreicht. Bei vielen bekannten Algorithmen ist es generell sehr schwer, eine genaue Aussage sowie eine gewisse Wiederholbarkeit der Ergebnisse zu erreichen, zum Beispiel bei der Verwendung der Verfahren, die auf den ICP-Algorithmen (engl. für „iterative closest point“) basieren. Die Problematik liegt in der unterschiedlichen Größe der Punktwolken sowie der niedrigen Qualität der Ergebnisse des Stereokamerasystems. Partiell liefern die Algorithmen eine brauchbare Transformation, dennoch sind die Ergebnisse oft ungenau. Die Wiederholbarkeit ist nicht gegeben, die Bestimmung einer aussagekräftigen Transformationsmatrix kann nicht garantiert werden. Die Erweiterung des vorgestellten Algorithmus durch die Fusion lokaler Merkmale, wie Kanten und Ecken, ist nach Meinung des Autors vielversprechend. Genauso verhält es sich mit der Fusion von zuvor extrahierten statischen Oberflächen oder Bewegungssegmentierungen. Der Autor ist davon überzeugt, dass die Kalibrierung unter Zuhilfenahme des 3-D-Kalibrierungskörpers die akkuratesten Ergebnisse liefert. Durch den Vergleich mit den bekannten Methoden der 2-D-Bildverarbeitung wird diese Annahme zusätzlich bestätigt.

Eine weitere Verbesserung kann durch die Interprätation der redundanten Daten in den Überlappungsbereichen erreicht werden. Dabei wurden die Kameradaten für zum Aufspüren problematischer Bereiche beider Sensoren genutzt. Diese Bereiche können mit etablierten Methoden der 2-D-Bildverarbeitung, wie Medianfilter oder Histogramme, angewandt auf den HSV-Farbraum, lokalisiert werden. Zum Beispiel können die homogenen Bereiche durch die Mean-Shift-Segmentierung oder Similarity-Measure-Algorithmen detektiert werden. Die folgende Tabelle zeigt die Priorisierung eines der Sensoren für die Tiefeninformation abhängig von den festgestellten Problembereichen.

Problems	LRF	SCS	Avail. inform.
Homogeneous surfaces	×	–	d_{LRF}
Invalid values (LRF)	–	×	d_{SCS}+c+t
Tiny objects	–	×	d_{LRF}+d_{SCS}+c+t
Black surfaces	–	×	d_{SCS}+c+t
Lighting conditions	×	–	d_{LRF}
Strong reflexions	–	–	–
–	×	×	d_{LRF}+d_{SCS}+c+t

Dabei steht LRF als Akronym für den Laserscanner (engl. für „laser range finder“), SCS für das Stereokamerasystem, d (engl. für „depth“) für die Tiefeninformation, c für Farb- und t für Texturinformation. Die Tiefeninformation von dem mit einem × markierten Sensor ist priorisiert in der vorliegenden Situation. Andernfalls werden genauere Informationen eines simulierten 3-D-Laserscanners bevorzugt. Die implementierte Architektur basiert auf den hier erläuterten Annahmen, die aus den Erkenntnissen der oberen Tabelle resultiert. Abbildung 3.6 zeigt das vereinfachte Flussdiagramm des partiell kolorierten 3-D-Rekonstruktionssystems.

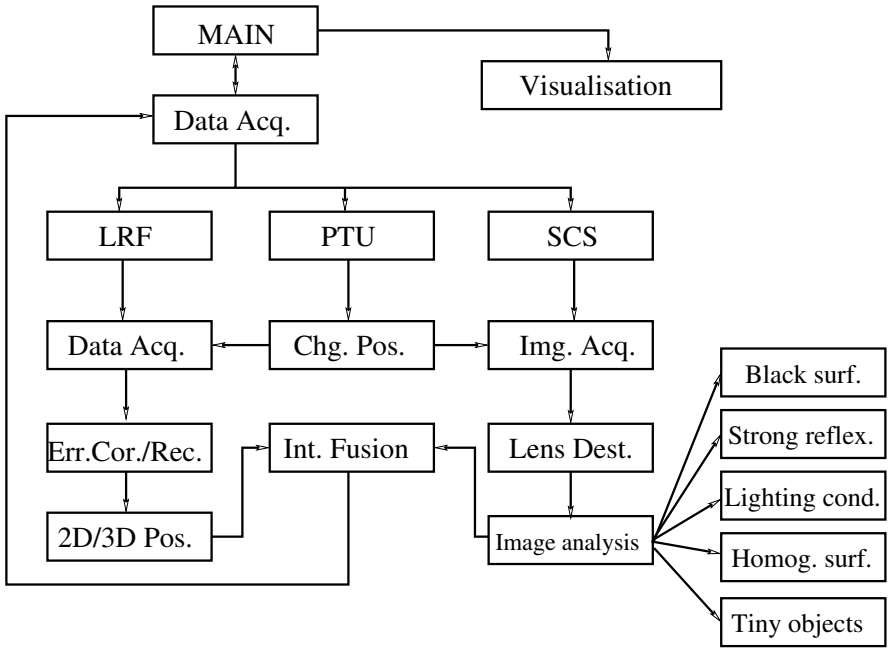


Abbildung 3.6.: Vereinfachtes Flussdiagramm eines partiell kolorierten 3-D-Rekonstruktionssystems.

Je mehr Sensoreninformationen zur Verfügung stehen, desto akkurater und zuverlässiger sind die Ergebnisse des Algorithmus. Konsequenterweise ist eine der besten Strategien für die Bewegung des Roboters, eine dem ROI gegenüber vernünftige Position und Orientierung zu finden. Damit können alle möglichen Sensorinformationen mit minimalen Fehlern akquiriert werden. Das Problem ist vergleichbar mit dem Next-Best-View-Problem der Bildverarbeitung.

Eine andere Möglichkeit besteht in der Integration weiterer Sensoren. Zum Beispiel ist die Hauptplattform des Arbeitsbereichs TAMS, der Serviceroboter TASER mit einem weiteren 2-D-Laserscanner ausgestattet. Damit die 3-D-Information über die Umgebung gesammelt werden kann, wird hier der 6-DOF-Roboterarm als Schwenk-Neige-Einheit genutzt. Dabei existieren zwei Möglichkeiten, diese Daten zu integrieren: zum einen durch die beschriebene Kalibrierung und zum anderen durch die Registrierung, dies wird ermöglicht durch die bekannte Geometrie und Wiederholgenauigkeit des Roboterarms. In der vorliegenden Arbeit sind alle Sensorwerte auf das zentrale Koordinatensystem des Roboters registriert. Daher wurden die bei der Kalibrierung gewonnenen Daten des Stereokamerasystems auf das Koordinatensystem des Laserscanners projiziert. Die Laserstrahlen werden ihrerseits in Relation zu dem Punkt im Fuß der Schwenk-Neige-Einheit abgebildet, wie in der Gleichung 3.3 dargestellt.

$$\begin{bmatrix} c(\varphi)c(\theta) & -s(\theta)c(\varphi) & s(\varphi) & -d_z s(\frac{\varphi}{2}) \\ s(\theta) & c(\theta) & 0 & 0 \\ -c(\theta)s(\varphi) & s(\theta)s(\varphi) & c(\varphi) & -d_z s(\frac{\varphi}{2})c(\frac{\varphi}{2}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

hierbei stehen c und s für Kosinus beziehungsweise Sinus, θ ist der Defliktionswinkel zwischen den Laserstrahlen (ein Mehrfaches des Abstands zwischen zwei Laserstrahlen) und φ der Winkel der Schwenk-Neige-Einheit, d_z ist die vertikale Komponente des Vektors zum Koordinatenursprung.

Mit zwei weiteren simplen Translationen kann dieses Frame in das Koordinatensystem des Roboters überführt werden.

Abbildung 3.7.: Eine typische Alltagsszene in einem Büro. Eine Tischplatte mit zufällig darauf platzierten Objekten in unterschiedlicher Tiefe.

Für die Evaluation wird die oben beschriebene Plattform eingesetzt. Die in der Abbildung 3.7 dargestellte Szene ist typisch für eine Büroumgebung. Die Ergebnisse der adaptiven multimodalen Sensor-Fusion zwischen einem Laserscanner und einem Stereokamerasystem, die auf einer Schwenk-Neige-Einheit montiert sind, zeigt Abbildung 3.8. Das Ergebnis ist eine partiell kolorierte Punktwolke. Die Perspektive wurde zwecks besserer Sicht verändert. Der Erfassungsbereich des Laserscanners ist durch den Aufbau beschränkt. Der Algorithmus ist in C/C++ implementiert, die Visualisierung wurde mithilfe der OpenGL realisiert.

Es ist denkbar, auf die Ergebnisse der Fusion ein Rekonstruktionsverfahren für die Oberflächenrekonstruktion, hier den Ball-Pivoting-Algorithmus, anzuwenden. Das Ergebnis der Rekonstruktion wird in der Abbildung 3.9 präsentiert. Das Bild zeigt nur den fusionierten Bereich an, alle Voxel außerhalb des fusionierten Bereichs wurden schwarz eingefärbt.

Die implementierte Anwendung kombiniert frühe Sensor-Fusion und den Interpretati-

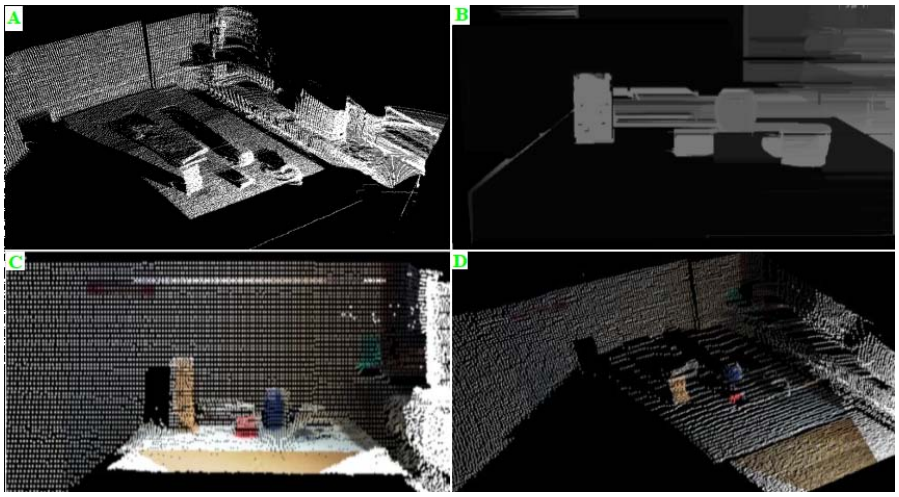


Abbildung 3.8.: a) 3-D-Bild eines Laserscanners, das durch eine Schwenk-Neige-Einheit bewegt wird. b) Disparitätsbild des Stereokamerasystems. Die Abbildungen c) und d) zeigen die frühe Sensor-Fusion der Farb- und Tiefeninformation aus zwei unterschiedlichen Perspektiven.

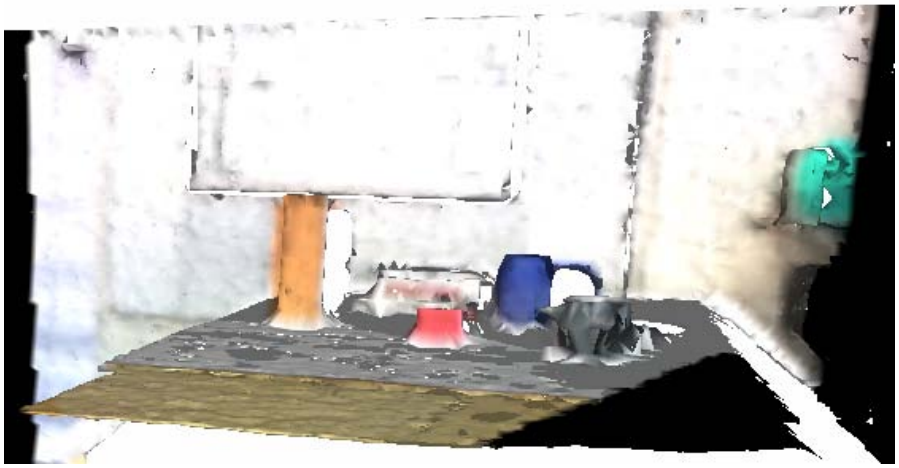


Abbildung 3.9.: Die Ergebnisse der 3-D-Rekonstruktion mit dem Ball-Pivoting-Algorithmus inklusive des Interpretationsschrittes. Nur der fusionierte Bereich ist sichtbar.

onsschritt. Die Bilder machen deutlich, dass hiermit bessere Ergebnisse erreicht werden können, als nur durch die Anwendung der Sensor-Fusion. Durch die Analyse der Bilder

können bessere Entscheidungen in Bezug auf die Sensordaten und deren Fusion getroffen werden. Die Darstellung der Kaffeetasse aus der Abbildung 3.7 beispielsweise ist in den Laserscannerbildern kaum zu identifizieren. Darüber hinaus fehlt diese nach der früheren Fusion komplett, siehe Abbildungen 3.8 a) sowie c) und d). Die Interpretationskomponente ermöglicht bessere Resultate, so ist die Tasse in der Abbildung 3.9 klar identifizierbar und kann deutlich besser erkannt werden. Ein weiteres Beispiel aus derselben Szene ist die homogene Struktur der Tischplatte. Das Stereokamerasystem findet keine korrespondierenden Punkte und damit keine Tiefeninformation, siehe Abbildung 3.8 b). Nach dem Interpretationsschritt und der Fusion ist die Tischplatte klar erkennbar und kann in weiteren Schritten segmentiert werden.

Natürlich akkumuliert das konstruierte 3-D-Wahrnehmungssystem die Fehler einzelner individueller Komponenten und erschwert die Auswertung der Ergebnisse. Die Probleme einzelner Sensoren wurden innerhalb dieses Kapitels betrachtet, deswegen wird hier nicht weiter darauf eingegangen. Eine andere Schwierigkeit hängt direkt mit den physikalischen Eigenschaften der Laserscannerstrahlen zusammen. Die Größe der Laserstrahlen wächst mit der Entfernung. Der Laserscanner liefert entweder die kürzeste Distanz zu einem Objekt oder den Mittelwert der Distanzen, abhängig von der Einstellung der verbauten Fotodiode in dem verwendeten Laserscanner. Gleichzeitig nimmt die Oberfläche, die durch ein Pixel beziehungsweise Voxel dargestellt wird, an Größe zu. Dabei wird aber ein Pixel weiterhin nur mit einem Wert pro Farbkanal abgebildet. Dieser Fehler beeinflusst die Ergebnisse der Multi-Sensor-Fusion abhängig von der Punktdichte, also ist der Einfluss in der Mitte schwächer, nach außen nimmt er zu. Eine weitere Fehlerquelle ist der Parallax-Effekt, der mit der größer werdenden Distanz zwischen zwei Sensoren zunimmt, aber mit größer werdender Entfernung abnimmt.

Während der Experimente wurde mit einer maximalen Fehlerrate von fünf Pixel in horizontaler und drei Pixel in vertikaler Richtung gemessen. Dabei wurden mehrere Tischszenen mit unterschiedlichen Objekten sowie Positionen untersucht. In der Abbildung 3.10 werden die Ausbreitung der Laserstrahlen, die Pixelgröße in Relation zur Entfernung sowie die maximale Fehlerrate der beiden Größen in *mm* grafisch zusammengefasst.

Das Ziel der multimodalen Sensor-Fusion ist nicht nur die verbesserte 3-D-Rekonstruktion, sondern viel mehr die Nutzung der gewonnenen Informationen für die Robotik, wie etwa für die Objekterkennung. In diesem Sinne wurden als Beispiel zwei stark frequentierte Methoden der 2-D-Bildverarbeitung, nämlich Farbsegmentierung und Kantendetektion, auf die Ergebnisse angewandt. Die Resultate können in der Abbildung 3.11 betrachtet werden.

Dabei nutzen wir den JSEG [WYP05] für die Farbsegmentierung und Sobel-Algorithmen [SF68] für die Kantendetektion als häufig anzutreffende Vertreter beider Merkmalsextraktionklassen. Die Ergebnisse sind durchaus mit den Ergebnissen für die originalen 2-D-Bilder vergleichbar. Werden die beiden Algorithmen eingesetzt und durch die Merkmale des 3-D-Raums erweitert, zum Beispiel durch die Nutzung von euklidi-

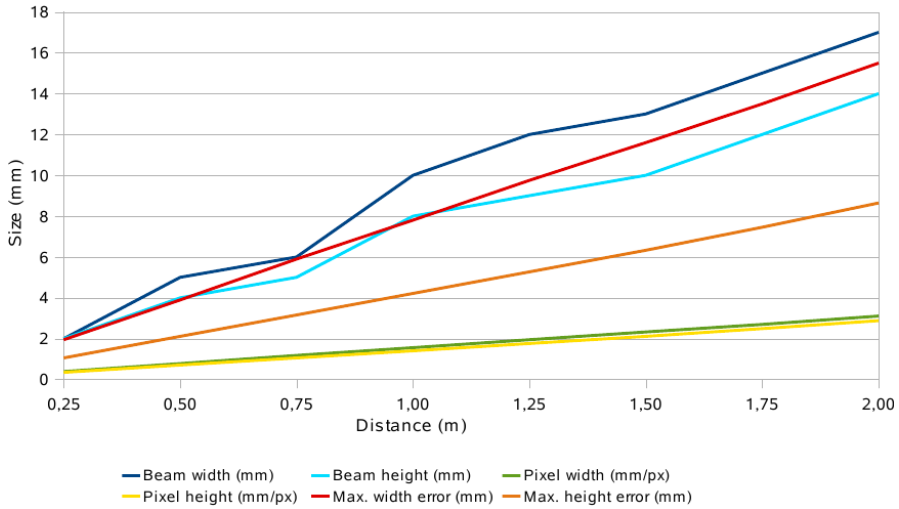


Abbildung 3.10.: Grafische Zusammenfassung der Ausbreitung der Laserstrahlen, der Pixelgröße in Relation zur Entfernung sowie der maximalen Fehler der beiden Größen in *mm*.

Abbildung 3.11.: Anwendung zweier stark frequentierter Methoden der 2-D-Bildverarbeitung auf einer partiell kolorierten 3-D-Punktwolke. Farbsegmentierung (links) und die Kantendetektion (rechts).

schen Distanzen, können sogar Griffe für die segmentierten Objekte berechnet werden. Abbildung 3.12 illustriert die Berechnung der möglichen Griffe des Roboterarms für die vereinfachte Szene und ein vereinfachtes Modell der blauen Tonne, die auch in Abbildung 3.7 zu sehen ist. Die Farbinformation wurde entfernt, damit eine bessere Performanz der Anwendung erreicht werden konnte. Zur weiterführenden Information über die Kalkulation der Griffe sowie die Funktionsweise des Simulators sei der Laser auf [BZ06] verwiesen. Abbildung 3.13 zeigt die Anwendung des kalkulierten Griffs auf ein reales Szenario.

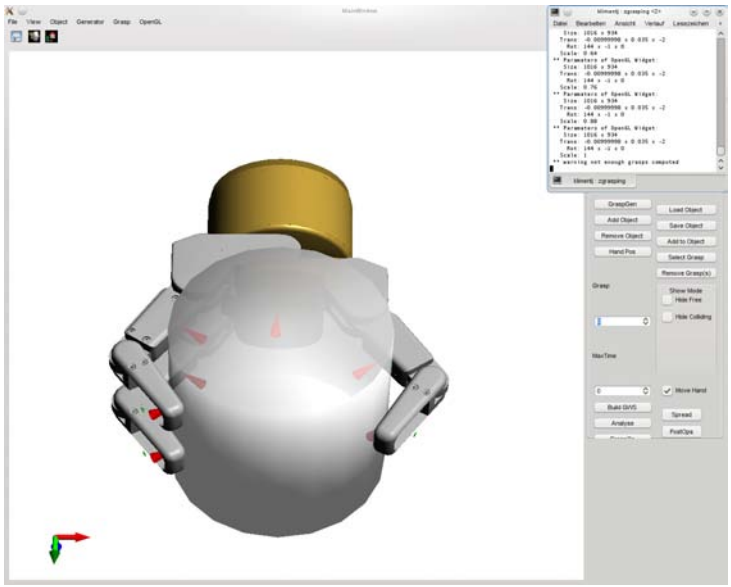


Abbildung 3.12.: Die Berechnung des Griffs und dessen Simulation für das vereinfachte blaue Tonnenmodell aus der Originalabbildung. Die Farbinformation wurde entfernt, damit eine bessere Performanz der Anwendung erreicht werden konnte.

Abbildung 3.13.: Die Anwendung eines zuvor kalkulierten Griffs auf ein reales Szenario.

Die zeitliche Performanz der gesamten resultierten Anwendung ist linear und direkt proportional zur Auflösung der Schwenk-Neige-Einheit und der Anzahl der 3-D-Punkte.

Eine Kombination aus präzisen Distanzen und Farbinformationen eröffnet der Robotik neue Möglichkeiten. Nicht nur die Objekterkennung, sondern auch die Interaktion eines Serviceroboters mit seiner Umgebung kann davon außerordentlich profitieren. Zusätzlich erhöht die Fusion mit dem Laserscanner, der einerseits sehr schnell Daten liefert und andererseits weniger Abhängigkeit von den Lichtverhältnissen als eine Kamera aufweist, enorm die Sicherheit beim Einsatz in menschlicher Umgebung.

3.5.2. Kalibrierung der PrimeSense-Sensoren

In Kapitel 2 wurden bereits die Gründe für den Einsatz von PrimeSense-Sensoren ausgiebig erläutert. Dieser Abschnitt behandelt die Fusion der von den PrimeSense-Sensoren gelieferten Information. Der PrimeSense-Sensor besteht aus einer Farbkamera sowie einem Infrarotprojektor und einer Infrarotkamera. Die Kalibrierung profitiert von dem Gehäuse, das alle Komponenten zusammenhält. Damit bleiben die meisten Parameter, wie die Basislinie (Abstand beider Kameras zueinander) und die Orientierung, während der Nutzungsdauer weitgehend konstant. Zwar werden von Microsoft und ASUS unterschiedliche Kameras verbaut, die physikalischen Eigenschaften sind aber größtenteils die gleichen, Xtion hat jedoch etwas größere Öffnungswinkel. Dieser Umstand und die Tatsache, dass die Datenausgabe in demselben Format produziert wird, weckt die Idee, ein universelles Kalibrierungsverfahren zu entwickeln. Da Farb- (RGB) sowie Tiefeninformationen (D, engl. für „depth“) geliefert werden, entstand sehr schnell das Konzept einer RGB-D-Kamera.

Die originalen Ausgangsdaten sind weder zueinander synchronisiert noch kalibriert. Da der Sensor mit einer maximalen Frequenz von $30\,Hz$ arbeitet, kann die Synchronisation weitgehend vorausgesetzt werden. Bei der Kalibrierung von Farb- und Tiefeninformation kam, wie schon bei der Fusion von Stereokamerasystem und Laserscanner, ein Kalibrierungskörper zum Einsatz. Diesmal wurde ein 2-D-Kalibrierungskörper entwickelt, der mehrere kreisförmige Öffnungen aufweist und in einer einheitlichen Farbe, die sich stark vom Hintergrund unterscheidet, lackiert wurde. Der obere Teil der Abbildung 3.14 visualisiert den genutzten Kalibrierungskörper sowie die gelieferten Daten beider Sensoren.

Der Kalibrierungskörper erlaubt die Auffindung von korrespondierenden Punkten, von Kreiszentren in Farb- und von Öffnungen in Tiefenbildern. Dabei wird wie folgt vorgegangen: Der Kalibrierungskörper wird vor einem einheitlich gefärbten Hintergrund in einem Abstand zwischen $0,60\,Meter$ und $1,20\,Meter$ platziert. Die Daten einzelner Sensoren werden akquiriert und zuerst separat betrachtet. Das Farbbild wird im ersten Schritt im HSV-Farbraum transformiert und die Farbhistogramme werden gebildet. Die Histogrammbildung ermöglicht die Sortierung der Pixel in Weiß, Schwarz und Grau. Die restlichen Punkte werden gemeinsam angeordnet. Mit diesen Daten werden Vorder- und Hintergrund separiert. Danach kann der ROI, die planare Kalibrierungsfläche (Vordergrund), gefunden werden. Sind die Kreise identifiziert, können durch den Vergleich mit den Nachbarn, das sogenannte Growing, erst die Umrandung der Kreise und im

Algorithm 2 The calibration algorithm for Kinect-like sensors

procedure CALIBRATION(RGB-D)
    Synchronized data acquisition of RGB and depth information.
    RGB:
    Transform to HSV color space.
    Generate a histogram.
    Sorting to white, black, gray, and other pixels.
    Fore- / background separation.
    Reduce to ROI (calibration body in foreground).
    Finding a rgb-holes.
    Sorting a rgb-holes.
    Depth:
    Transform the distances to standard values (m)
    Generate the binary image (if depth between $0.65\,m - 1.1\,m$ then 1 else 0).
    Reduce to ROI through remove the lines (if number of 0 values $> 99\,\%$).
    Finding a depth-holes (growing).
    Sorting a depth-holes.
    Both:
    Check and remove of false-positives.
    Calculation of hole centers.
    Calculation of and finding the minimal distance between RGB- and depth hole centers.
    Generate the corresponding pairs.
    Calculation of homogeneous transformation matrices (with help of RANSAC, 7/8 point, and LMEDS algorithms).
    Return the **homogeneous transformation matrices**.
end procedure

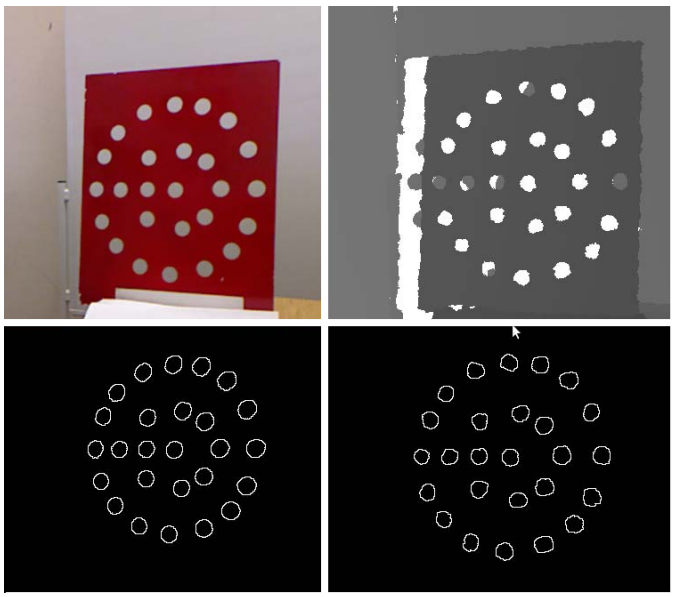


Abbildung 3.14.: Oben sichtbar die Ausgangsdaten der RGB- und Tiefenkamera des eingesetzten Kalibrierungskörpers. Unten sichtbar die daraus segmentierten Merkmale.

nächsten Schritt deren jeweiliges Zentrum bestimmt werden. Abschließend werden die Zentren sortiert. Bei der Tiefeninformation werden erst die Distanzen in die Standardmetrik (Meter) überführt und binarisiert. Liegt der Wert für den jeweiligen Pixel in dem angegebenen Intervall, wird sein Wert auf 1 gesetzt, sonst auf 0. Der ROI wird durch die Entfernung der Linien mit überwiegend mehr als 99 % an 0-Werten herausgebildet. Nach der Vordergrundsegmentierung können durch einen ähnlichen Growing-Ansatz die Umrisse der Öffnungen sowie deren Zentren bestimmt werden. Mit der Sortierung der Zentren wird dieser Schritt abgeschlossen. Im nächsten gemeinsamen Schritt werden die Ergebnisse optimiert, indem zuerst die Falsch-Positive-Merkmale entfernt und die kürzesten Distanzen zwischen den Kreisen und den Öffnungen berechnet werden. Ausgegeben werden zueinander korrespondierende Punktpaare. Der Pseudocode des eigens implementierten Algorithmus ist in dem Algorithmus 2 zusammengefasst.

Sind die korrespondierenden Punkte bestimmt und in ausreichender Anzahl vorhanden, kann die Transformationsmatrix berechnet werden. Im Rahmen der vorgestellten Kalibrierungsmethode wurden vier unterschiedliche Algorithmen getestet und evaluiert, nämlich RANSAC, der 7/8-Punkte-Algorithmus sowie LMEDS (least median of squares). Die beste Fundamentalmatrix wurde mit dem RANSAC-Algorithmus berechnet. Wird die Fundamentalmatrix angewendet, kann eine neue Datenstruktur akquiriert werden, die zueinander registrierte Farb- und Tiefeninformation beinhaltet.

Eine theoretische Evaluation solcher Algorithmen ist enorm aufwendig. Auf der einen Seite wächst der Informationsgehalt eines Pixels proportional zu der Entfernung; auf der anderen Seite übt die veränderte Perspektive (Basislinie) beider Sensoren einen negativen Einfluss (Parallaxeffekt) aus. Während der praktischen Experimente wurde die Qualität des Verfahrens nachgewiesen, es zeigte sich ein großes Potenzial. Der resultierende Algorithmus ist robust, schnell und einfach zu benutzen. Der Benutzer platziert den Kalibrierungskörper in einem bestimmten Abstand vor der Kamera, der Vordergrund soll sich dabei möglichst stark vom Hintergrund unterscheiden, und startet die Kalibrierung. Die Kalibrierung läuft komplett automatisch ab und kann nach Bedarf beliebig wiederholt werden. Die Transformationsmatrix wird automatisch berechnet und aktualisiert. Der Sensor ist sofort einsatzbereit und liefert eine kolorierte Punktwolke, die weiter verarbeitet werden kann. Weiterführende Informationen sowie die Implementierungsdetails können der Publikation des Autors [KRZ12] entnommen werden.

Im folgenden Kapitel 4 werden basierend auf den fusionierten Daten die Szenenanalyse sowie die Objektsegmentierung ausführlich dargestellt und beschrieben.

Kapitel 4

Szenenanalyse und Interpretation

Werden die Ziele der Arbeit noch einmal in Erinnerung gerufen, wird relativ schnell klar, dass die Arbeit eine Verbindung zwischen den Sensoren und der Erkennung auf der semantischen Ebene herstellt. Die Daten der Sensoren werden aufbereitet und zur einander registriert. Auf die resultierenden Daten werden mehrere Erkennungsalgorithmen angewandt. Die Ergebnisse werden durch parallel ablaufende Prozesse, wie das aktive Sehen, und durch einen möglichen Eingriff des Roboters in die Szene aufgewertet. Natürlich würde die Szenenanalyse auf der semantischen Ebene die Objekterkennung deutlich verbessern, so können zum Beispiel auch die Relationen zwischen den Objekten betrachtet werden. Die Vorteile können wieder einmal anhand eines Restaurantszenarios demonstriert werden, so wird zum Beispiel eine Tasse auf einem Tisch erkannt. Ein Objekt in der Nähe kann aber nicht eindeutig zugeordnet werden, und die Wahrscheinlichkeit für eine Salz- oder Pfeffermühle oder einen Behälter mit Zucker sind ungefähr gleich groß. Auf der semantischen Ebene würde die Analyse die Wahrscheinlichkeit einer Zuckerdose deutlich hervorheben und damit die Objekterkennung erneut verbessern. Eventuell reicht sogar eine deutlich geringere Wahrscheinlichkeit, um auf der semantischen Ebene die Erkennung zu spezifizieren.

Viele Forscher haben die möglichen Vorteile der zusätzlichen Analyse auf der semantischen Ebene erkannt. Die Analyse auf der semantischen Ebene basiert auf der guten und zuverlässigen Objekterkennung (Hypothesenbildung). Deswegen werden im Folgenden einige Arbeiten aus diesem Bereich, die ähnlich argumentieren wie diese Arbeit, vorgestellt und in Relation zur der vorliegenden Dissertation gesetzt.

Obwohl das Endziel, die Erstellung 3-D semantischer Karten, bei allen im Folgenden vorgestellten Publikationen gleich ist, unterscheiden sich die Wege dorthin doch stark. Viele der Autoren sind eher an den „quasi“ bereits vorhandenen Karten interessiert, weniger aber daran, wie diese Art der Repräsentation erreicht werden kann. Deswegen wird dem Weg den rohen Sensordaten bis hin zur Objekterkennung, die den Schwerpunkt der

vorliegenden Dissertation darstellt, etwas weniger Aufmerksamkeit geschenkt. Dennoch gibt es einige interessante Ansätze und Ergebnisse.

In [BGM+11] und [ZPBB11] wird der Fokus auf die Erkennung von Handgriffen gelegt. Es werden immer Küchenszenarien betrachtet, und zuerst die planare Segmentierung durchgeführt. Nach der planaren Aufteilung werden die Türen und anschließend die Handgriffe gesucht. Für die Segmentierung als auch für die Detektion werden RANSAC und Schattierungen im 2-D-Bild verwendet. Die Registrierung der Sensordaten findet nicht statt. Dafür wird aber die Möglichkeit der Öffnung der erkannten Tür analysiert, auch der Perspektivenwechsel ist in die Erkennung integriert.

In [GWAH13] liegt der Brennpunkt bei der Kategorisierung der Möbel in einer Büroumgebung mit dem Ziel, eine semantische Karte zu erstellen. Zum Einsatz kommen die planaren Segmentierungs- sowie „Region Growing"-Verfahren.

Die Autoren der Publikation [MBBM+14] nutzen mehrere Kameras für die Realisierung der aktiven Wahrnehmung. Auch hier werden zuerst die planaren Flächen segmentiert. Anschließend findet eine Kategorisierung über Part-Graph-Methoden mit einem Hashtable statt. Für alle drei Publikation, durch die Verwendung eines einzigen, Detektors gilt: Scheitert das eingesetzte Verfahren, kann die Unternehmung keinen Erfolg mehr liefern.

Die Publikationen [KBA+14] und [KDH14] verwenden mehrere Verfahren zur Objekterkennung in Kombination mit Reasoning. Die Objekterkennung basiert auf dem Framework von Aldoma [AMT+12], das auf die Algorithmen der Point Cloud Library (PCL) zurückgreift. Der Veröffentlichung liegt ein volumetrischer Detektor zugrunde, eine Verbesserung der Erkennung wird klar nachgewiesen.

Die Autoren der Publikation [WLS14] gehen ein ganz anderen Weg und stellen den „Semantic Hierarchy Graph"vor. Die Arbeit basiert auf den verbesserten Conditional Random Fields (CRF). Ein Bild wird in Segmente unterteilt, die nur einer Klasse zugeordnet werden. Aufgrund dieser Klassifikation werden semantische Bäume aufgestellt. Die Evaluation findet in häuslicher Umgebung statt, dabei werden eher größere Objekte wie Kühlschrank, Sofa und Bett in Bildern gefunden und klassifiziert. Passend zu den Aufgaben werden zwei Relationstypen definiert: Is-part-of und Is-type-of. Die Nutzung der RGB-D-Daten stabilisiert einerseits das Verfahren, andererseits bietet eine 3-D semantische Karte mit Farbinformationen eine deutlich bessere Grundlage für das weitere Vorgehen.

Alle vorgestellten Publikationen benutzen einige Elemente, die auch in der vorliegenden Arbeit eingesetzt werden. Der hier verfolgte Weg, bei dem die Sensorinformationen registriert, kombiniert und damit vervollständigt sowie mehrere Detektoren für unterschiedliche Objekteigenschaften eingesetzt werden, konnte aber nicht festgestellt werden. Dennoch ist die Bedeutung der semantischen Karten unumstritten. Des Weiteren fehlt natürlich auch die standardisierte Rückführung von den semantischen Karten zur der Objekterkennung, die definitiv in den nächsten Jahren viele Forscher beschäftigen wird.

Dennoch können im Rahmen der vorliegenden Arbeit nur einige wenige Schritte in die-

se Richtung gemacht werden. Wie bereits an mehreren Stellen dieser Arbeit beschrieben, werden die Sensordaten fusioniert. Des Weiteren werden unterschiedliche Detektoren angewandt, außerdem wird das aktive Sehen und das Eingreifen in den Erkennungsprozess integriert. Dadurch können mehr Daten genutzt und ausgewertet werden, was die Qualität der Hypothesenbildung enorm steigert. Außerdem werden in der Datenbank, die über die Objekt ID mit der Erkennung verbunden ist, viele Objekteigenschaften hinterlegt. Unter anderem sind dort die Informationen über Form, Farbe, Gewicht, über den Reibungskoeffizienten zwischen dem Objekt und dem Tisch etc. hinterlegt. Somit gibt es Möglichkeiten über diese Eigenschaften die Objekterkennung zusätzlich zu verbessern.

Den Schwerpunkt der vorliegenden Dissertation bildet die Anwendung von verschiedenen Methoden auf unterschiedlichen Ebenen der Objekterkennung. Damit steht die Kombination und Anwendung mehrerer differierender Verfahren im Mittelpunkt, die eine relativ neu Entwicklung in der Robotik darstellt.

Dieses Kapitel beschreibt die Segmentierung und die darauf basierende Anwendung der Detektoren sowie die Herausbildung, Interpretation, und Reaktion auf die resultierenden Ergebnisse. Dabei steht nicht nur die Analyse der Szenen, die nur einzelne voneinander abgegrenzte Objekte beinhalten, sondern auch die Szenenanalyse mit mehreren Objekten, teilweise auch mit partieller oder totaler Verdeckung im Vordergrund. Die Definition des Begriffs „Verdeckung“wurde im Abschnitt 1.2 gegeben. Die Übersicht und Analyse der vergleichbaren Arbeiten wurde in Kapitel 1 dargestellt.

Die einzelnen Detektoren werden auf die segmentierten Cluster angewandt. Somit stellt die Segmentierung einen wichtigen und notwendigen Vorverarbeitungsschritt dar. Scheitert die Segmentierung, so sind auch die Analyse und Interpretation der vorliegenden Szenen nicht erfolgreich. Daher wird im folgenden Abschnitt die Segmentierung vorgestellt und detailliert beschrieben.

Die dargestellte Vorgehensweise sowie alle Algorithmen und Methoden wurden im Rahmen der Publikationen in [KRZ12] und [KRZ13] veröffentlicht.

4.1. Tiefen- und Farbsegmentierung von Objekten

Die erste Segmentierung läuft innerhalb der Punktwolke mithilfe der euklidischen Distanz und einem fest vorgegebenen Schwellenwert, zum Beispiel 2 cm, ab. Damit jedem Cluster die Farbinformation möglichst exakt zugeordnet werden kann, sind folgende Schritte notwendig. Dabei werden die homogenen Gleichungen sowie deren Herleitung aus dem Kapitel 2 genutzt. Der erste Schritt findet unter Annahme von idealen Bedingungen im sogenannten Lochkameramodell statt. Betrachtet wird die Projektion eines Objektpunktes an Position $(x, y, z)^T$ im Kamerakoordinatensystem auf eine Position $(u, v)^T$ im Bildframe.

Zuerst erfolgt die ideale Projektion des Objektpunktes $(x, y, z)^T$ im Kamerakoordinatensystem auf den Punkt $(x', y')^T$ in der Bildebene mit Brennweite $f = 1$. Damit $x' = \frac{x}{z}$

und $y' = \frac{y}{z}$. Damit wird die vom Objektpunkt $(x, y, z)^T$ emittierte Strahlung an der Position $(x', y')^T$ erfasst.

Dieselbe Ebene wird für die Modellierung der optischen Verzerrung genutzt. Ein Punkt an der Position $(x', y')^T$ wird damit an die Position $(x'', y'')^T$ verschoben.

$$x'' = x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x' y' + p_2(r2 + 2x'^2) \tag{4.1}$$

und

$$y'' = y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y'^2) + 2p_2 x' y', \tag{4.2}$$

wobei $r^2 = x'^2 + y'^2$; (k_1, k_2, k_3) Koeffizienten der radialen und (p_1, p_2) der tangentialen Verzeichnung sind.

Zum Schluss werden dann die Punkte des verzerrten Bildes mittels intrinsischer Parameter auf der Position $(u, v)^T$ des Kamerasensors (2-D) abgebildet.

$$u = f_x x'' + c_x \tag{4.3}$$

und

$$v = f_y y'' + c_y, \tag{4.4}$$

wobei (c_x, c_y) Koordinaten des Hauptpunktes sind und (f_x, f_y) die Bildweite in Pixel darstellen.

Somit kann ein ROI innerhalb des RGB-Bildes gefunden werden, der die Farbinformationen, unter Annahme einer genauen intrinsischen Kalibrierung der Kamera beinhaltet, die mit zuvor ermittelten Tiefeninformationen übereinstimmen. Der Vorteil liegt darin, dass die Farb- und Textursegmentierung nur auf die gefundenen Cluster angewendet werden kann, was eine bessere Performanz und Ressourcenschonung garantiert.

Abbildung 4.1 visualisiert die Ergebnisse des Clustering sowie der Tiefen- und Farbsegmentierung: Das linke Bild ist eine original Aufnahme der Kamera. In der Mitte ist die Tiefensegmentierung mit darauf projizierten Farben zu sehen. Das rechte Bild zeigt den zur Tiefensegmentierung korrespondierenden RGB-Imageausschnitt. Die einzelne farbigen Kreise visualisieren gefundene Farbinformationen an den entsprechenden Stellen.

Mit bloßem Auge ist erkennbar, dass aus den beiden Abbildungen mehr Informationen gewonnen werden können, als nur bei der Betrachtung eines der beiden Bilder. Zusätzlich bietet die Farbsegmentierung weitere Möglichkeiten zur Beurteilung der Qualität der Segmentierung. Einer der elementaren Ansätze wäre zum Beispiel die Annahme, dass beim Vorhandensein mehrerer besonders komplementärer Farben eine Untersegmentierung vorliegt, usw.

Die Problematik der auf der euklidischen Distanz basierenden Segmentierung liegt im Unterschreiten des eingestellten Schwellenwerts. Wird dieser zu groß gewählt, werden die Objekte untersegmentiert, wie die Abbildung 4.1 verdeutlicht. Natürlich ist der

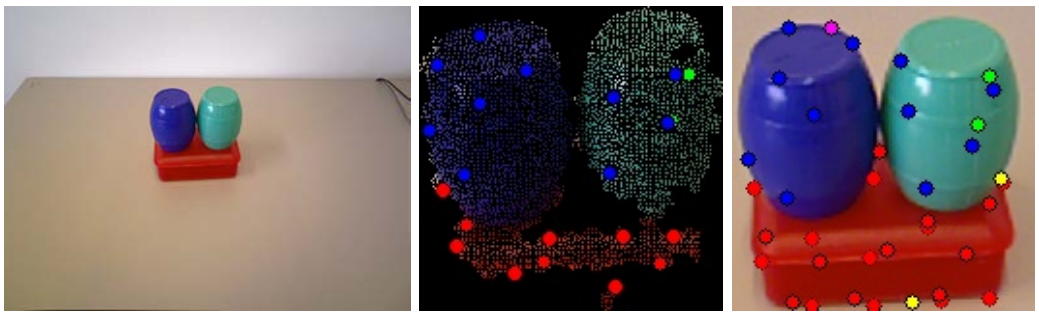


Abbildung 4.1.: Visualisierung der vorläufigen Ergebnisse: Links abgebildet ein Originalbild der Kamera. Das mittlere Bild zeigt die Tiefensegmentierung mit darauf projizierten Farben. Rechts dargestellt ist der zur Tiefensegmentierung korrespondierende RGB-Imageausschnitt. Die einzelnen farbigen Kreise visualisieren gefundene Farbinformation an den entsprechenden Stellen.

Schwellenwert von der Distanz zum Sensor abhängig und darf nicht zu klein gewählt werden. Einen extremen Fall stellt die Verdeckung dar. Daher bieten Methoden, die mit mehreren unterschiedlichen Objekteigenschaften arbeiten, größere Möglichkeiten für die sichere Objekterkennung. Somit bleibt der Bedarf, weitere Objektmerkmale im Analyseprozess zu verwenden, weiterhin bestehen und es können weitere Merkmale aus dem 2-/3-D-Raum hinzugenommen werden.

4.2. Detektoren zur Objekterkennung

Sind die Bereiche segmentiert, kann mit der Objektdetektion begonnen werden. Obwohl die vorliegende Arbeit sich auch mit der Objektdetektion bei partieller Verdeckung beschäftigt, soll zuerst die Erkennung einzelner Objekte sicher gewährleistet werden. Dafür betrachten wir zunächst einzelne Detektoren, die sich einerseits als zuverlässig erwiesen haben und andererseits möglichst alle wichtigen Bereiche, die bereits in Kapitel 3.1 vorgestellt worden sind, für die Objektdetektion abdecken. In der Praxis hat sich die Kombination von Detektoren als ein äußert effektives Mittel zur Effizienz- und Performanzsteigerung erwiesen. Einige Algorithmen lagen bereits implementiert vor und wurden in die bestehende Architektur integriert und für die Umgebungsverhältnisse parametrisiert, andere wurden implementiert, verbessert und/oder weiterentwickelt. Anschließend werden die Möglichkeiten aufgezeigt, um die Ergebnisse unterschiedlicher Detektoren intelligent zusammenzufassen und auswerten zu können.

Zu Beginn werden die Detektoren betrachtet, die im zweidimensionalen Raum operieren. Danach werden die 3-D-Detektoren sowie einige Erweiterungen von 2-D- auf 3-D-Detektoren betrachtet.

4.2.1. Farbinformationen

Die Farbsegmentierung kann für mehrere Zwecke eingesetzt werden. Einerseits bietet die Farbsegmentierung einige Anhaltspunkte zur „Güte“des Clustering: Sind zu viele unterschiedliche Farben vorhanden, steigt die Wahrscheinlichkeit der Untersegmentierung, wie im mittleren Image der Abbildung 4.1 dargestellt. Dennoch sind einige Objekte im menschlichen Umfeld speziell bunt gehalten, sodass der Bedarf an weiteren Merkmalen besteht. Außerdem ist die Farbe eine Eigenschaft des Lichts und somit von den Lichtverhältnissen abhängig.

Abbildung 4.2.: Zwei Beispiele der Farbsegmentierung mit einem der bekanntesten Farbsegmentierungsalgorithmen JSEG. Links dargestellt eine einfache Szene mit klaren Farbübergängen und optimal erzielten Segmentierungsergebnissen. Rechts abgebildet eine komplexere Szene mit vielen ineinandergreifenden Texturen und kaum weiter verwendbaren Ergebnissen.

Grundsätzlich kann die Farbsegmentierung durch die Anwendung mehrerer Schwellenwerte, jeweils zwei für eine Farbe (ein maximaler und ein minimaler Wert), in den Farbkanälen erfolgen. Durch sich verändernde Lichtbedingungen und unterschiedliche Farbfacetten liefern solche Algorithmen kaum reproduzierbare Ergebnisse. Hier wird die Funktion der Farbsegmentierung anhand eines der besten Farbsegmentierungsalgorithmen namens JSEG [LTP09] verdeutlicht. Die Hauptidee hinter dem Algorithmus ist die Aufsplittung in zwei Schritte. Im ersten Schritt werden die Farbinformationen klassifiziert, was die Unterteilung in Regionen ermöglicht. Dabei bleibt der Algorithmus erstmals nur im Farbraum aktiv. Danach wird jedem Pixel ein Muster der passenden Farbe einer Klasse zugeordnet. Danach werden die Grenzen der Bereiche durch die Anwendung zweier Schwellenwerte bestimmt. Anschließend werden die einzelnen Regionen betrachtet und durch den Vergleich der Nachbarpixel (Region Growing) spezifiziert und klassifiziert. Damit kann der Algorithmus nicht nur auf die einzelnen Bilder, sondern

auch zuverlässig auf Bildsequenzen angewendet werden.

Die Abbildung 4.2 visualisiert die Farbsegmentierung mit einem der bekanntesten Farbsegmentierungsalgorithmus JSEG. Dabei liefert das Algorithmus für das linke Bild mit wenigen Farben, einem hohen Kontrast und klaren Farbübergengen - das Originalbild ist in der Abbildung 4.1 zu sehen - ein beeindruckendes Ergebnis. Hingegen sind die Ergebnisse für das rechte Bild mit seiner komplexeren Struktur und vielen heterogenen Merkmalen - wie in der Abbildung 4.4 zu sehen, die Ergebnisse kaum weiter verwendbar.

Infolgedessen kann die Farbsegmentierung nur als ein zusätzlicher Indikator verwendet werden. Genauso wie bei der Überprüfung der „Güte“des Clustering kann anhand der Ergebnisse eine Tendenz für Unter- oder Übersegmentierung erkannt werden. Für die praktische Realisierung werden die Ergebnisse der Farbsegmentierung weniger gewichtet im Vergleich zu den Ergebnissen anderer verwendeter Detektoren.

4.2.2. Erweitertes SIFT/SURF

Ein gutes Beispiel für robuste und zuverlässige Merkmale stellen dabei die SIFT-Merkmale („Scale Invariant Feature Transformation“, engl. für skalierungsinvariante Merkmalstransformation) [Low04] beziehungsweise die SURF-Merkmale („Speeded Up Robust Features“, engl. für beschleunigte, robuste Merkmale) [BTG06] dar. Beide Merkmalsdetektoren werden auf Graustufenbilder angewendet mit dem Unterschied, dass SURF-Detektoren nicht den Gauß- sondern den Mittelwert-Filter einsetzen. Durch die Verwendung von Integralbildern erreichen SURF-Detektoren einen konstanten Zeitaufwand und sind grundsätzlich schneller als SIFT-Detektoren.

Die Hauptidee des Algorithmus ist die mehrmalige Anwendung des entsprechenden Filters mit unterschiedlicher Parametrisierung, wie zum Beispiel im Falle des Gauß-Filters mit unterschiedlichen λ-Werten. Bleiben gewisse Merkmale über alle Durchläufe konstant erhalten, werden diese in einer Menge, dem sogenannten Feature-Bag, zusammengefasst. Soll ein Objekt klassifiziert werden, benötigt der Algorithmus ein Bild, wobei nur das entsprechende Objekt das Bild komplett ausfüllen muss. Es können mehrere Bilder nacheinander verglichen werden. Für die Klassifikation eines Objekts werden nicht die Bilder, sondern dazugehörige Merkmalsmengen verglichen. Dies ermöglicht die deutliche Beschleunigung des Algorithmus. Die typische Visualisierung der Merkmale und des Merkmalsvergleichs kann in der Abbildung 4.3 betrachtet werden.

Die auf SIFT bzw. SURF basierenden Methoden sind schnell, robust und liefern gute Ergebnisse. Sind genügend Merkmale vorhanden, können Objekte sicher und wiederholbar erkannt werden. Im Falle partieller Verdeckung können die Methoden meistens die Objekte trotzdem erkennen, sofern genügend Merkmale in den unverdeckten Bildbereichen vorhanden sind. Die Algorithmen sind zudem in der Lage, trotz der Skalierungsunterschiede und perspektivischen Verzerrungen die Objektbegrenzung (engl. für „Bounding Box“) zu bestimmen. Diese Abgrenzung kann von weiteren Algorithmen genutzt werden, etwa zur Berechnung des Schwerpunkts.

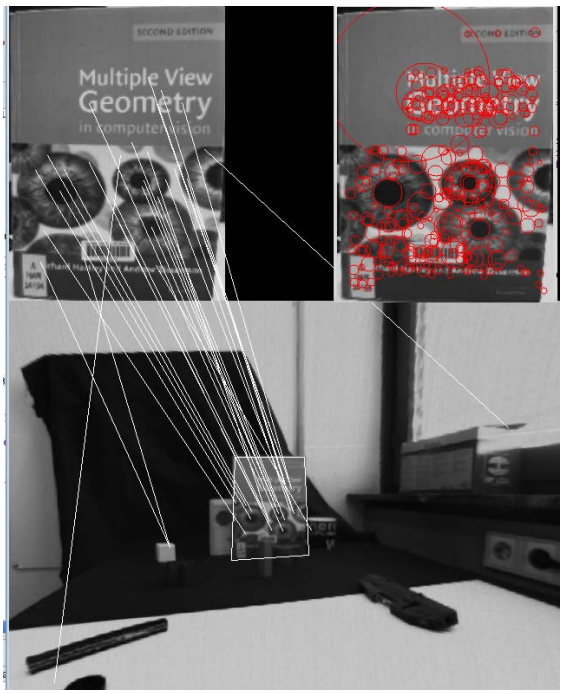


Abbildung 4.3.: Visualisierung der SIFT-Merkmalen und des Merkmalsvergleiches. Oben rechts dargestellt sind die gefundenen Merkmale, visualisiert durch rote Kreise. Die Bilder links oben und mittig unten zeigen die korrespondierenden Merkmale beider Aufnahmen. Trotz einiger weniger Ausreißer ist das Objekt deutlich erkennbar.

Die Abbildung 4.4 visualisiert die Objektdetektion mit einem SIFT-Detektor. Aus dem Bild ist ersichtlich, dass partielle Verdeckung die Objekterkennung nicht beeinflusst, wenn genügend Merkmale auf den Objekten gefunden werden können. Drei Objekte werden sicher erkannt, die dahinter stehende Flasche, mangels der Anzahl der Merkmale, aber nicht mehr. Durch die Transformation der Begrenzung des ursprünglichen Objekts kann die Umrandung des gefundenen Objekts im neuen Bild trotz der partiellen Verdeckung korrekt eingezeichnet werden.

Wie alle Detektoren hat natürlich auch der SIFT/SURF-Detektor seine Nachteile. So wird die gleiche Textur überall als gleiches Objekt erkannt, auch wenn es sich um ein reales Objekt und seine Abbildung handelt oder um ein gleich aussehendes Objekt in unterschiedlichen Skalierungen. Um diesen Nachteil zumindest geringfügig zu verringern, soll bei den Originalbildern auf einen möglichst exakten Ausschnitt eines Objekts aus seiner Umgebung geachtet werden. Des Weiteren ist SIFT/SURF ein bildbasiertes Verfahren und damit von den Lichtbedingungen abhängig.

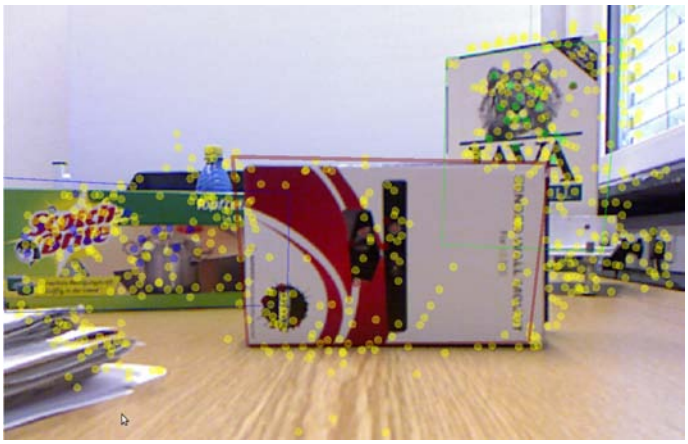


Abbildung 4.4.: Visualisierung der Objektdetektion mithilfe eines SIFT-Detektors. Das Bild zeigt deutlich, dass trotz der partiellen Verdeckung alle drei Objekte erkannt werden, sofern genügend Merkmale in nicht verdeckten Bildausschnitten gefunden werden können.

Sind die signifikanten Merkmale bestimmt, liefert der Algorithmus nicht nur diese, sondern auch die Objektbegrenzung sowie Position und Orientierung des Objekts im Bild (2-D). Wird ein Objekt trotz der vorhandenen Verdeckung erkannt, kann die 2-D-Objektumrandung kaum für weitere Analysen genutzt werden. Ein Greifen des gefundenen Objekts ist nicht möglich, auch die 2-D-Transformation kann nicht aussagekräftig genutzt werden. Die bestimmten signifikanten Merkmale können aber durchaus verwendet werden. Hier kann erneut das fusionierte 3-D-Bild angewendet werden. In der vorliegenden Arbeit werden, ermöglicht durch die Multi-Sensor-Fusion, die signifikanten Merkmale durch die Tiefeninformationen ergänzt. Somit steht eine Menge an 3-D-Punkten eines erkannten Objekts zur Verfügung. Damit kann einerseits versucht werden, alle Punkte in einer bestimmten Tiefe aus dem untersegmentierten Cluster zu separieren. Andererseits kann bei der vorhandenen Tiefeninformation des Datenbankbildes auch die 3-D-Transformation direkt bestimmt werden.

Die oben beschriebene Erweiterung hat mehrere positive Aspekte. Einerseits wird die Performanz des Algorithmus dadurch nicht beeinflusst. Des Weiteren werden die Merkmale auf 3-D erweitert und bieten mehr Möglichkeiten für die Szenenanalyse und die Handhabung der gefundenen Objekte sowie der partiellen Verdeckung.

Durch die oben aufgeführten Charakteristiken der Algorithmen stellen deren Ergebnisse einen wichtigen Bestandteil der Objekterkennung dar. Auf der einen Seite können diese aufgrund ihrer Sicherheit, Wiederholbarkeit und Robustheit extrem hoch gewichtet werden, auf der anderen Seite sind diese Methoden schnell und können somit als Erstes (sofern nicht parallel genutzt) angewandt werden. Des Weiteren kann der Algorithmus

durch mehrere Bilder eines Objekts erweitert werden, dadurch kann die Wahrscheinlichkeit der richtigen Erkennung deutlich gesteigert werden. Auch das Greifen eines gefundenen Objekts und die damit verbundene Auflösung der partiellen Verdeckung sind realisierbar.

4.2.3. Größe

Wird nur mit 2-D-Bildern gearbeitet, ist die Skalierung eines Objekts innerhalb einer Szene ohne externes Wissen nicht bekannt. Damit kann die Größe eines erkannten Objekts nie eindeutig, sondern nur aus der Bildinformation heraus bestimmt werden.

Auch hier kann die vorhandene 3-D-Information vorteilhaft ausgenutzt werden. So kann die Objektgröße, natürlich nur aus der gegebenen Perspektive, direkt aus dem 3-D-Cluster kalkuliert werden. Da die Information nicht komplett ist, kann das Objekt größer beziehungsweise länger ausfallen, aber nicht umgekehrt. In der vorliegenden Arbeit wird eine 3-D-Umrandung des Clusters mithilfe der PCA (Principal Component Analysis) bestimmt und zum Abgleich mit den Datenbankinformationen genutzt. Zwar reicht es nicht für eine eindeutige Erkennung, dennoch kann die aus dem 3-D-Cluster bestimmte Größe als Ausschlusskriterium verwendet werden. Dadurch kann der Vergleich mit deutlich kleineren Objekten aus der Datenbank von vornherein ausgeschlossen werden. Somit wird nicht nur die Performanz gesteigert, sondern, wie im Falle des ICP^2, auch die Erkennungsgenauigkeit. Unter bestimmten Blinkwinkeln kann sogar eine 3-D-Umrandung erkannt werden, die die Auswahl noch weiter eingrenzt.

4.2.4. Erweitertes ICP (ICP^2)

Diese Arbeit nutzt ROS als Framework sowie einige integrierte Methoden. Für die Objekterkennung wird dabei ein sogenannter „Iterative Distance Fitter“eingesetzt, auf den im Abschnitt 4.2.5 näher eingegangen wird. Das Vorgehen basiert auf dem ICP-Algorithmus, der in Kapitel D.3 ausführlich beschrieben wird. Dabei werden die Punktwolken nur in zwei Dimensionen verglichen, was hauptsächlich zwei Nachteile mit sich bringt. Erstens werden nur rotationssymmetrische Objekte und zweitens nur solche mit einer bestimmten Orientierung, symmetrisch zur nach oben zeigenden Z-Achse, erkannt. Zusätzlich soll das Modell aus der Datenbank diesen Kriterien exakt entsprechen.

Daraus ergibt sich die Aufgabe, diese Nachteile zu kompensieren. Dafür wird ein ICP-Algorithmus genutzt, der in drei Dimensionen arbeitet. Dabei wird der RANSAC-Algorithmus verwendet, der in Abschnitt D.9 vorgestellt wird. Der Algorithmus versucht, einen zufällig ausgewählten Voxelsatz aus der Punktwolke in das Modell zu transformieren. Diese Methode ist iterativ, erzielt die bestmögliche Transformation und eine Ähnlichkeitswahrscheinlichkeit basierend auf den Distanzen zwischen den Punkten. Das Ähnlichkeitsmaß basiert auf der Minimierung des quadratischen Fehlers, der in der Gleichung D.1 vorgestellt wird. Zwar sind die oben genannten Nachteile damit kompensiert,

es bleibt aber immer noch ein grundlegendes Problem aller ICP-Algorithmen bestehen. Wird ein kleineres Objekt in ein größeres transformiert und sind die beiden einander ähnlich in Bezug auf ihr Kurvenverhalten, liefert der Algorithmus extrem hohe Wahrscheinlichkeiten für deren Gleichheit.

Betrachten wir den euklidischen Abstand beziehungsweise die euklidische Norm für einen 3-D-Vektor, die für das Ähnlichkeitsmaß als Grundlage genutzt wird und in der Gleichung 4.5 definiert ist.

$$d(\vec{v}, \vec{\omega}) = \|\vec{v} - \vec{\omega}\| = \sqrt{(v_1 - \omega_1)^2 + (v_2 - \omega_2)^2 + (v_3 - \omega_3)^2} \tag{4.5}$$

Somit steigt die Wahrscheinlichkeit für einen positiven Vergleich proportional zur Größe des Objekts. Damit die Methode sichere und solide Ergebnisse liefert, wurde beschlossen, den Algorithmus zu erweitern und den Vergleich in beide Richtungen durchzuführen, sowie beide Ergebnisse zusammenzufassen. Die Gleichung 4.5 besagt, je kleiner das Ähnlichkeitsmaß desto größer die Wahrscheinlichkeit für die Gleichheit. Liegen die Ergebnisse für die beiden Vergleiche vor, können diese mit der gleichen Wahrscheinlichkeit zusammengefasst werden. Dafür werden die einzelnen Werte zuerst auf Minimum normiert und anschließend gleich verteilt zusammengefasst. Die Berechnung der resultierenden Wahrscheinlichkeit wird in der Gleichung 4.6 präsentiert.

$$\frac{1}{2} \cdot \left(\frac{MAX_{c2m} - Cloud2Model}{MAX_{c2m} - MIN_{c2m}} \cdot 100 + \frac{MAX_{m2c} - Model2Cloud}{MAX_{m2c} - MIN_{m2c}} \cdot 100 \right) \tag{4.6}$$

Die Abbildung 4.5 visualisiert erste Ergebnisse des Algorithmus. Dabei wird in der Gazebo-Simulation eine Kaffeetasse durch die Sensoren des simulierten PR2-Roboters erfasst und vom Algorithmus zur Veranschaulichung zunächst mit nur vier Datenbankmodellen verglichen. Wegen der zweifachen Anwendung des ICP-Algorithmus, dem beidseitigen Vergleich, wird der Algorithmus im Weiteren als ICP^2 bezeichnet.

Der ROS-eigene Algorithmus, genauso wie der Vergleich einer dreidimensionalen ICP-Methode zwischen dem Database-Modell und der Eingangspunktwolke, liefert 100 % Wahrscheinlichkeit für das größte Objekt in der Datenbank, das auf einem hölzernen Griff montierte Tablett. Wird dagegen die Eingangspunktwolke mit dem Datenbankmodell verglichen, wird die Kaffeetasse zur 100 % als Plastiktonne erkannt. Der verwendete ICP^2-Algorithmus sowie dessen fusionierte Ergebnisse, präsentiert im Diagramm 4.6, liefern 100 % Wahrscheinlichkeit für die Kaffeetasse, alle weiteren Wahrscheinlichkeiten liegen deutlich niedriger.

Wie bereits erwähnt, liegt der größte Nachteil der ROS-eigenen Objekterkennungsmethode in dem Vergleich nur von zwei Richtungen (x, y). Daraus resultieren zwei Bedingungen, die erfüllt sein müssen: Erstens geht die Anwendung davon aus, dass ein Objekt rotationssymmetrisch ist, außerdem soll es aufrecht auf dem Tisch platziert werden. Um diesen Nachteil weiter zu erforschen, wurden unter denselben Bedingungen, wie in Abbildung 4.5 präsentiert, weitere Tests durchgeführt. Dabei wird die Orientierung der Tasse

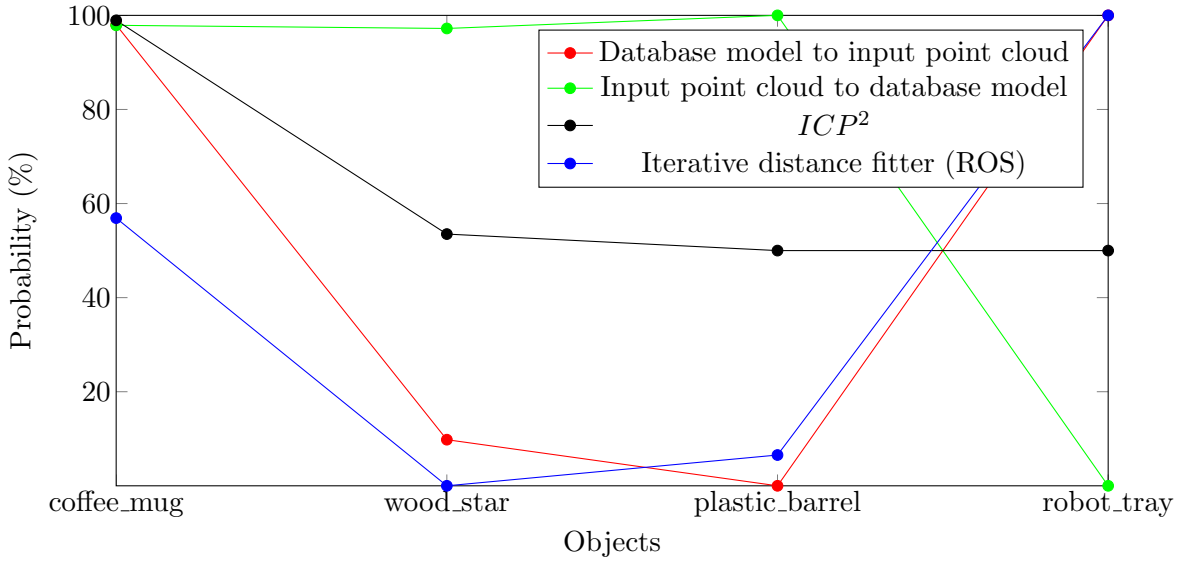


Abbildung 4.5.: Vergleich zwischen der Standard-ROS-Objekterkennung (Iterative Distance Fitter) und ICP in jeweils nur eine Richtung sowie mit einem eigens entwickelten ICP^2-Ansatz.

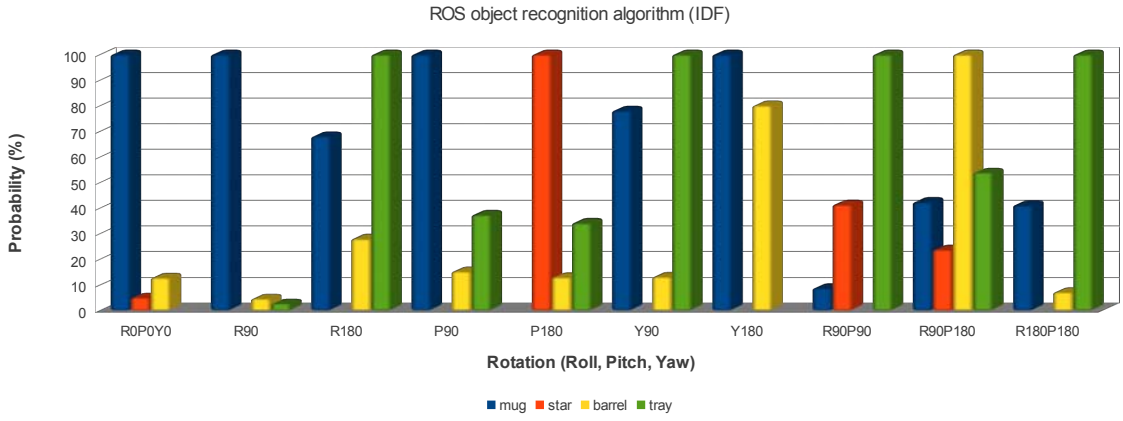


Abbildung 4.6.: Ergebnisse der ROS-eigenen Objekterkennungsmethode. Durch die Änderungen der Orientierung wird die Tasse nur in 40 % der Fälle richtig erkannt.

auf dem Tisch in drei Richtungen variiert (Roll, Pitch, Yaw), danach wird versucht, mit ROS-eigenen sowie der ICP^2 Methode die Tasse zu erkennen. Währenddessen werden

zehn unterschiedliche Orientierungen jeweils zehnmal getestet und zusammengefasst.

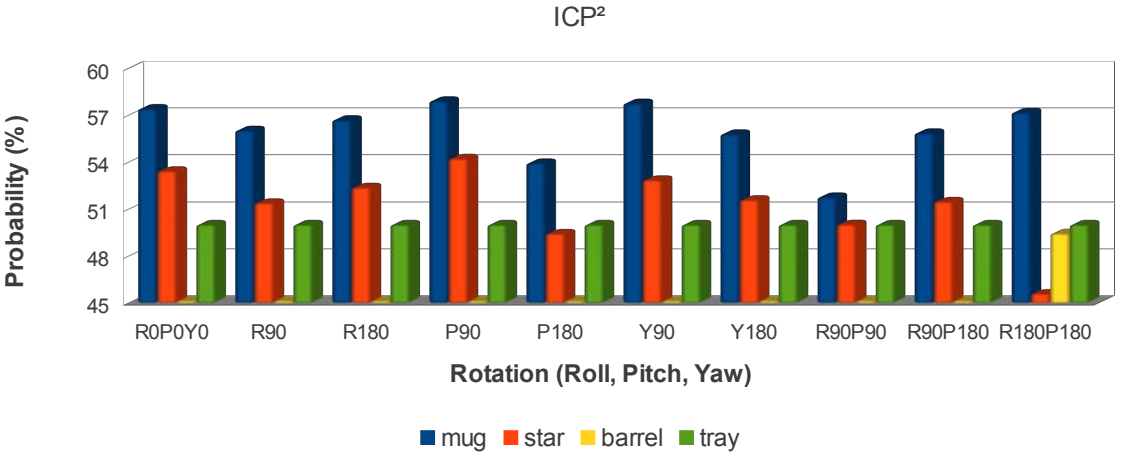


Abbildung 4.7.: Ergebnisse des ICP^2-Algorithmus. Die Tasse wird unter allen getesteten Orientierungen sicher und zuverlässig erkannt.

Die Resultate der ROS-Objekterkennungsmethode sind nur bedingt visualisierbar. Die Methode findet zuerst die bestmögliche Transformation der eingehenden Punktwolke zu einem Datenbankmodell und liefert anschließend die Summe der quadratischen Fehler zwischen den Eingangspunkten und den näheren Punkten des Modells. Je kleiner der Wert, desto größer ist die Wahrscheinlichkeit eines positiven Vergleichs. Für die bessere Darstellung werden die Ergebnisse jedes einzelnen Tests auf ein Minimum normiert. Abbildung 4.6 verdeutlicht die Problematik des ROS-Standard-Algorithmus, in 60 % Prozent der Fälle liefert die Methode falsche Ergebnisse.

Die Ergebnisse des ICP^2-Algorithmus dagegen bestätigen davor gemachte Überlegungen und Annahmen. Die Tasse wird sicher und zuverlässig erkannt. Bei insgesamt 100 Testdurchläufen ist es nur einmal vorgekommen, dass die Tasse zwar die höchste Wahrscheinlichkeit aufwies, die aber nur geringfügig größer war als die Wahrscheinlichkeit des Tabletts. Abbildung 4.7 visualisiert diese Ergebnisse.

Der ICP^2-Algorithmus ist robust und stabil. Natürlich verdoppelt die Methode den Rechenaufwand. Wird aber die alltägliche Situation betrachtet, steht der Roboter während der Objekterkennung vor dem Tisch und akquiriert sowie analysiert die Sensordaten. Alle weiteren Soft-/Hardwarekomponenten warten auf diese Anweisungen und beanspruchen nur geringeren Ressourcenbedarf. Damit stehen genügend Ressourcen zur Verfügung. Des Weiteren erlaubt die Struktur des Algorithmus eine problemlose Verteilung auf mehrere CPU-Kerne, somit können die beiden Versuche in separaten Threads gestartet werden. Die resultierende Ausführungszeit ist identisch mit der Zeit, die der ursprüngliche Algorithmus in Anspruch nahm.

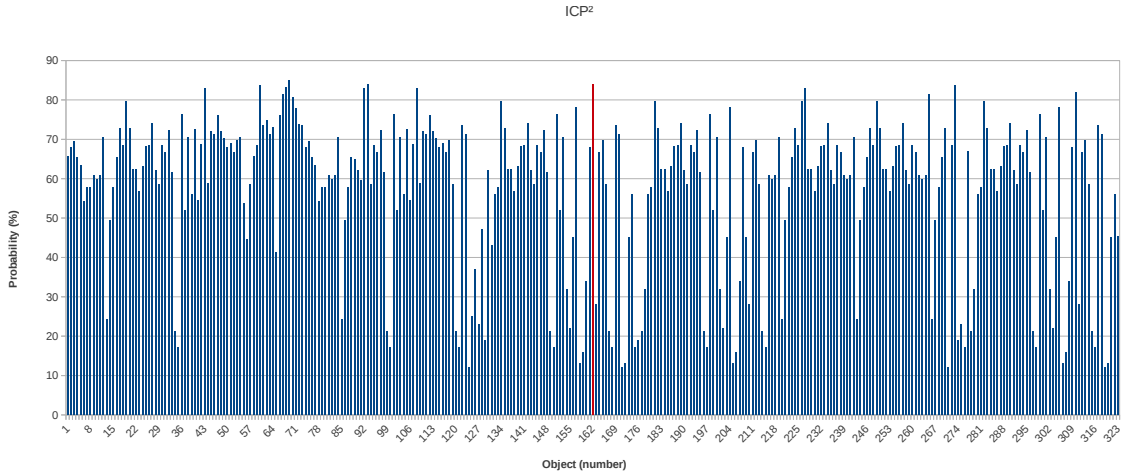


Abbildung 4.8.: ICP^2-Algorithmus - Vergleich mit der Household-Database mit 323 Objekten.

Jedoch weist auch dieser Algorithmus Defizite auf, Abbildung 4.8 visualisiert die Grenzen des ICP^2-Algorithmus. Im obigen Beispiel erreichen vier Objekte größere oder gleiche Wahrscheinlichkeiten wie die Tasse: ein Zylinder, eine tonnenförmige Flasche, ein Glas und ein Tumbler. Dies sind alles Objekte, die sich in ihrer Oberflächenstruktur stark ähneln. Damit der Algorithmus verwendet werden kann, werden aufgrund der nahezu identischen Oberflächen weitere Merkmale benötigt, damit die genaue Objektklassifikation erfolgen kann.

4.2.5. Iterative Distance Fitter (IDF)

IDF[1] ist ein bereits implementierter und zur Verfügung stehender Algorithmus zur Objekterkennung in ROS. Der Detektor basiert auf dem ICP-Algorithmus und ist um eine Dimension, also auf 2-D, reduziert. Der Algorithmus funktioniert nur für rotationssymmetrische Objekte mit einer bestimmten Orientierung, was seine Verwendung extrem einschränkt. Die Performanz des Algorithmus ist in der Abbildung 4.6 dargestellt.

Die Methode arbeitet auf Clustern. Daher wird zuerst ein Tisch, die in der Szene dominierende planare Fläche, mithilfe des RANSAC-Algorithmus bestimmt. Danach wird nur die Punktwolke oberhalb der planaren Fläche betrachtet und unter Zuhilfenahme der euklidischen Distanz mit einem zuvor festgelegten Schwellenwert in Cluster unterteilt. Die Cluster werden einzeln dem Algorithmus übergeben und mit den Modellen der Datenbank in 2-D verglichen, wobei das Modell zur einer Punktwolke reduziert wird. Als

[1] http://wiki.ros.org/tabletop_object_detector

Ergebnis liefert der Algorithmus eine Rotationsmatrix sowie einen Translationsvektor, die es erlauben, das Cluster sowie das Modell ineinander zu transformieren. Daraufhin wird das Ähnlichkeitskriterium bestimmt, für das die Differenz der Quadrate der Distanz zwischen den benachbarten Voxel summiert wird. Je kleiner dieser Wert, desto exakter passen die beiden Punktwolken zueinander.

Im Laufe des Entwicklungsprozesses der vorliegenden Arbeit wurde beschlossen, die Methode trotz der oben genannten Nachteile zu übernehmen. Die Verwendung dieses Detektors ist mit seiner Schnelligkeit, Robustheit und Eindeutigkeit der Ergebnisse zu begründen. Liefert die Methode einen deutlich kleineren Wert gegenüber dem Durchschnitt (mindestens 40 % kleiner), kann mit hoher Wahrscheinlichkeit davon ausgegangen werden, dass es sich um das entsprechende Objekt handelt. Zusätzlich kann überprüft werden, ob das erkannte Objekt tatsächlich rotationssymmetrisch ist. Trifft das zu, steigt die Wahrscheinlichkeit, und die Ergebnisse des Detektors können in die gemeinsame Entscheidung miteinbezogen werden. Bei negativer Überprüfung bleiben die Ergebnisse unberücksichtigt.

4.2.6. Gruppierung korrespondierender Merkmale

Wie bereits beschrieben, steht eine Menge unterschiedlicher Detektoren zur Verfügung. Jedoch können viele von ihnen nur bedingt eingesetzt werden. So benötigen einige Detektoren Parameter, die exakt entweder zu einem Objekt oder zu einer Objektgruppe passen müssen. Werden Objekte mit unterschiedlichen Eigenschaften, Formen und Größen eingesetzt, ist es dem Detektor kaum möglich, diese mit ein und derselben Parameterkonfiguration zu erkennen. So sind solche Detektoren zwar bestens für bestimmte Aufgaben geeignet, aber im Kontext der vorliegenden Arbeit kaum verwendbar.

Ein Beispiel für einen solchen Detektor stellt die Gruppierung korrespondierender Merkmale (CG^2 engl. für „Correspondence Grouping“) dar [CF10]. Zuerst erwartet der Algorithmus eine Punktwolke, die statisch vorliegt oder über Sensoren akquiriert wird. Für die Cluster-Auffindung wird standardmäßig ein Hough-Algorithmus verwendet. Danach sucht die Methode nach den korrespondierenden Punkten zwischen dem Objekt und dem gefundenen Cluster. Die so bestimmten Punkte werden geometrisch zusammengefasst, was eine Berechnung der Orientierung ermöglicht.

Abbildung 4.9 stellt die Ergebnisse des CG-Algorithmus dar. Die linke Abbildung veranschaulicht die sichere Erkennung des Objekts. Wird die Szene durch weitere Objekte erweitert, erschwert dies die Erkennung. Dem Autor ist es nicht gelungen, die Parameter so einzustellen, dass die gleichzeitige Erkennung aller verwendeten Objekte sichergestellt werden konnte. Die meisten Versuche führten zu mehrmaligen falschen Klassifikationen des Objekts innerhalb einer Szene. Die rechte Abbildung zeigt, dass dasselbe Objekt nicht mehr richtig erkannt wird. Sogar wenn alle Parameter beibehalten werden, in-

[2] http://pointclouds.org/documentation/tutorials/correspondence_grouping.php

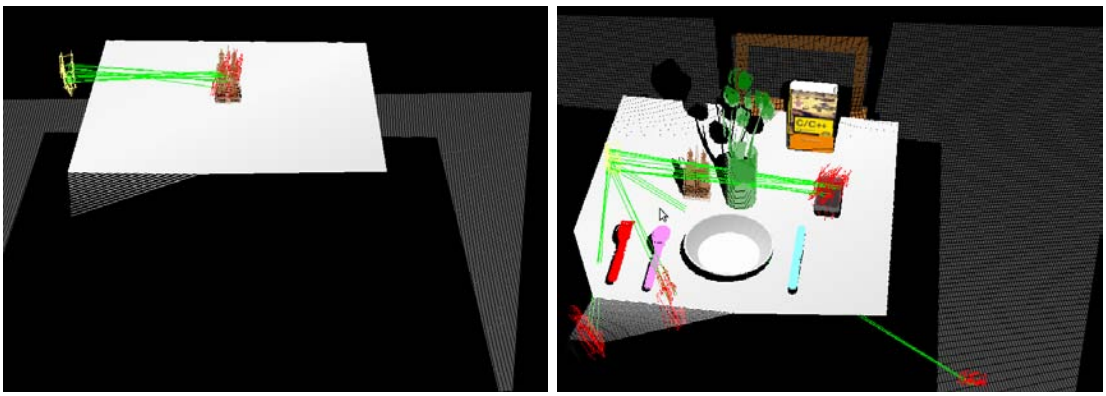


Abbildung 4.9.: Ergebnisse des CG-Algorithmus. Erfolgreiche Erkennung eines Öl- und Essigspenders links. Mit bloßem Auge ist ein Fehler in der Orientierung erkennbar. Das gleiche Objekt wird innerhalb der komplexeren Szene rechts (Gazebo-Simulator) nicht mehr erkannt.

klusive der Entfernung und des Blickwinkels, und die Szene durch weitere unverdeckte Objekte erweitert wird, kann eine zuverlässige Erkennung nicht mehr garantiert werden.

4.3. Szenenanalyse

Dieser Abschnitt beschäftigt sich mit der Auswertung der Erkennungsdetektoren sowie der anschließenden Szenenanalyse. Zuerst wird auf die formale Beschreibung der gewichteten Abstimmung eingegangen, ein Prozess, der für die spätere Detektorenkombination notwendig ist. Danach wird die Kombination der Detektoren ausführlich erläutert. Abschließend wird auf die von den Ergebnissen der Detektoren abhängigen nächstmöglichen Schritte zur Szenenanalyse eingegangen.

4.3.1. Gewichtete Abstimmung

Die gewichtete Abstimmung (engl. „Weighted Voting“) ist ein gängiges Verfahren in der Mathematik zur Analyse von Wahlverfahren, bei denen die Wählerstimmen unterschiedlich gewertet werden. Diesem Verfahren liegt die Annahme zugrunde, dass jede Wählerstimme mit einem nicht negativen Wert gewichtet ist. Die Entscheidung kann nur dann gefällt werden, wenn das zusammengefasste gewichtete Ergebnis über einem Schwellenwert liegt, zum Beispiel größer als 51 %. Meistens findet dieses Verfahren in Politik und Wirtschaft Anwendung [APL07]. Aber auch in der Informatik wird die gewichtete Abstimmung erfolgreich eingesetzt, zum Beispiel bei der farbbasierten Objekterkennung [vdGS05] oder der Mustererkennung, in der Überwachung und in Mensch-

Maschinen-Systemen [NP99].

Im Folgenden werden einige Definition präsentiert, die für die spätere Zusammenfassung der Ergebnisse einzelner Detektoren mittels gewichteter Abstimmung notwendig sind. Dabei wird davon ausgegangen, dass $N = \{1, ..., n\}$ den Satz der Wähler darstellt.

Definition 1 *Damit ein gewichtetes Abstimmungssystem beschrieben werden kann, müssen einzelne Gewichte für jeden Teilnehmer $w_1, w_2, ..., w_n$ sowie das notwendige Pensum für eine Entscheidung q definiert werden. Die folgende Notation ist die kürzeste Möglichkeit solcher Spezifizierung:* $[q : w_1, w_2, ..., w_n]$

Definition 2 *Die einfache Abstimmung ist ein Paar (N, v), wobei $v : 2^N \rightarrow \{0, 1\}$ hat folgende Eigenschaften, $v(\emptyset) = 0$, $v(N) = 1$ und $v(S) \leqq v(T)$ falls $S \subseteq T$. Die Koalition $S \subseteq N$ gewinnt falls $v(S) = 1$ und verliert wenn $v(S) = 0$.*

Definition 3 *Die einfache Abstimmung (N, v), wobei $W = \{X \subseteq N, \sum_{x \in X} w_x \geqq q\}$, wird als WVG zusammengefasst. WVG ist bezeichnet durch $[q; w_1, w_2, ..., w_n]$, wobei w_i die Gewichtung für den Detektor i ist. Allgemein ist $w_i \geqq w_j$, falls $i<j$.*

Definition 4 *Ein Detektor i ist kritisch in der Koalition S, wenn $S \in W$ und $S \backslash i \notin W$. Für jedes $i \in N$ wird die Anzahl der Koalitionen, in denen i kritisch für die Entscheidung v mit $\eta(v)$ war, notiert. Der Banzhaf-Index für den Detektor i in einer gewichteten Abstimmung v ist $\beta = \frac{\eta_i(v)}{\sum_{i \in N} \eta_i(v)}$.*

Definition 5 *In einer einfachen Abstimmung wird ein Detektor mit einem Banzhaf-Index von 0 als Dummy bezeichnet.*

Definition 6 *Ein Diktator in einer einfachen Abstimmung ist ein Detektor, der in jeder gewonnenen Koalition vorhanden und in jeder verlorenen Koalition abwesend ist.*

Definition 7 *Ein Detektor hat ein Vetorecht dann und nur dann, wenn er in jeder gewonnenen Koalition vorhanden ist.*

Die Anzahl der möglichen Permutationen ist die Fakultät aller Teilnehmer ($n!$). Im Rahmen der vorliegenden Dissertation stellen die Teilnehmer die verwendeten Detektoren dar.

Natürlich können nicht alle Regeln direkt umgesetzt werden. Vielmehr schafft die Einführung der gewichteten Abstimmung eine theoretische Basis, die die Verwendung mehrerer unterschiedlich gewichteter Detektoren für die Auffindung einer gemeinsamen Entscheidung erlaubt. Des Weiteren wird die Verwendung mehr oder weniger verlässlicher Detektoren ermöglicht. So kann das System problemlos erweitert werden. Liefert ein Detektor oder eine Detektorengruppe fundierte Ergebnisse, kann nur auf dessen/deren Basis eine Entscheidung getroffen werden. Somit kann das resultierende System an die meisten Szenarien sowie einzelne Detektoren / Detektorengruppen angepasst werden. Das System liefert mindestens das Ergebnis des besten Detektors oder verbessert dieses durch die Informationen weiterer Detektoren.

4.3.2. Kombination der Detektoren sowie Bestimmung der Orientierung

Für die Evaluation des in dieser Arbeit vorgestellten Systems werden Detektoren eingesetzt, die an unterschiedlichen Objekteigenschaften arbeiten. Diese sind Farbhistogramme, Größe, Volumen (ICP^2 und ROS-eigenes IDF) sowie lokale Merkmale (SIFT/SURF). Basierend auf der gewichteten Abstimmung werden die Ergebnisse der Detektoren zusammengefasst. Dabei werden die Detektoren nach ihrer Performanz eingeordnet und die Gewichte verteilt. So wird der ICP^2 höher bewertet als auf Größe basierende oder farbbasierte Detektoren. Sind aber die beiden letzteren sich einig in Bezug auf ein Objekt, steigt dessen Wahrscheinlichkeit und ermöglicht die Erkennung auch ohne den Einbezug weiterer Detektoren. Der SIFT-/SURF-Detektor fungiert dabei als ein Diktator, dies liegt darin begründet, dass wenn dieser Detektor ein Objekt erkennt, kann die Richtigkeit des Erkennungsprozesses mit sehr großer Wahrscheinlichkeit angenommen werden. Dennoch soll an dieser Stelle erwähnt werden, dass die Kombination der Detektoren jeweils an die Simulation und realen PR2 angepasst werden muss. So liefert die Simulation deutlich bessere Sensorwerte und unterliegt kaum der Abhängigkeiten in Bezug auf Lichtverhältnisse. Wohingegen üben die ändernde Lichtverhältnisse sowie Reflexionen auf dem realen PR2 einen extremen Einfluss auf die Ergebnisse der Detektoren aus. Leider konnte im Rahmen dieser Arbeit kein Verfahren entwickelt werden, der die Suche nach den bestmöglichen Parametern automatisiert.

Des Weiteren wird eine Datenbank benötigt, die unterschiedliche Objekteigenschaften bereitstellt. Dies macht es unmöglich bereits etablierte Evaluationsmetriken zu benutzen, da die meisten von denen entweder auf 2-D-Bildern basieren oder wie zum Beispiel in 3-D-Modellen nur eine Objekteigenschaft bereitstellen. Daher wird eine eigene Metrik, basierend auf einer eigenen Datenbank und mehreren Szenen, aufgebaut.

Die Erkennung der Objekte ist grundlegend, reicht aber allein nicht aus. Damit mit den erkannten Objekten interagiert werden kann, wird zusätzlich deren Position und Orientierung benötigt. Die Segmentierung liefert die Position eines Objekts, in dem der Schwerpunkt des gefundenen Clusters berechnet wird. Die Berechnung der Orientierung gestaltet sich etwas schwieriger. Einige der Detektoren, wie der ICP^2 oder IDF, berechnen die Orientierung während des Erkennungsprozesses. Somit kann diese für die Weiterverarbeitung genutzt werden. Werden aber die Objekte über SIFT-/SURF-, Farb- oder Größenmerkmale erkannt, muss die Orientierung separat bestimmt werden.

In der vorliegenden Arbeit wird in solchen Fällen ein ICP-Algorithmus genutzt. Diesem werden die erkannte Punktwolke sowie das dazugehörige Modell übertragen. Zurückgeliefert wird die homogene Transformation (4×4-Matrix), die aus einer Rotationsmatrix (3×3-Matrix) sowie einem Translationsvektor besteht. Dieser 3×1-Vektor stellt die affinen Transformationen dar. Der letzte Parameter ist der Skalierungsfaktor, der in der vorliegenden Arbeit, ermöglicht durch die Multi-Sensor-Fusion, immer eine eins ist.

In ROS kann die Orientierung des erkannten Objekts aber nur mit einer Quaternion geändert werden. Um die Quaternion zu berechnen, wird zuerst der Rotationsanteil

aus der homogenen Matrix extrahiert. Im Folgenden wird angenommen, dass eine 3×3-Rotationsmatrix M aus einzelnen Elementen $m_{i,j}$ besteht, wobei i die Nummer innerhalb der Zeile und j innerhalb der Spalte ist. Danach kann die Quaternion Q mit Elementen (q_x, q_y, q_z, q_w) wie folgt berechnet werden:

$$q_w = \frac{\sqrt{1 + m_{0,0} + m_{1,1} + m_{2,2}}}{2} \tag{4.7}$$

$$q_x = \frac{(m_{2,1} - m_{1,2})}{4q_w} \tag{4.8}$$

$$q_y = \frac{(m_{0,2} - m_{2,0})}{4q_w} \tag{4.9}$$

$$q_z = \frac{(m_{1,0} - m_{0,1})}{4q_w} \tag{4.10}$$

Die berechnete Quaternion stellt die Orientierung des erkannten Objekts dar und ermöglicht erst das Greifen. Je genauer die erkannte Position und Orientierung, desto wahrscheinlicher ist es, dass das Greifen erfolgreich ausgeführt werden kann. Zusätzlich besteht auch die Möglichkeit, statt nach einem Objekt nach einem Cluster zu greifen. Leider besteht dabei keine Möglichkeit, einen Einfluss auf die bestimmte Orientierung während der Bewegung auszuüben. Damit wäre es möglich, dass zum Beispiel der Kaffee während der Bewegung aus der Tasse schwappt. Ein weiterer Nachteil ist, dass die Griffe für das jeweilige Cluster zur Laufzeit berechnet und nicht wie bei einem bekannten Objekt aus der Datenbank geladen werden, was eine zusätzliche Wartezeit nach sich ziehen könnte. Auf die Genauigkeit der Berechnung der Orientierung wird in Kapitel 6 eingegangen.

4.3.3. Aktive Komponenten

Im Folgenden werden die aktiven Komponenten der vorliegenden Dissertation vorgestellt. Dabei wird zuerst auf die aktive Wahrnehmung und anschließend auf den aktiven Eingriff in die Szene eingegangen.

4.3.3.1. Aktive Wahrnehmung

Der Großteil der Veröffentlichungen zur Beeinflussung der Objekterkennung durch Verdeckung (engl. für „occlusion“) geht von dem Erkennen der Objekte innerhalb einer Szene unter Verwendung unterschiedlicher Blickwinkel aus, das sogenannte aktive Sehen, das bereits in Kapitel 3.4 vorgestellt und zum Beispiel in [MWLS11] ausführlich beschrieben wurde.

Auch in der vorliegenden Arbeit wird das Konzept des aktiven Sehens verwendet, zu Evaluationszwecken jedoch eingeschränkt auf die Bewegung des Torsos (hoch und runter) sowie der Plattform um ca. 650 mm nach links und rechts. Die Abbildung 4.10 visualisiert die Bewegung der Plattform und die daraus resultierende veränderte Perspektive in 2-D. Später in Kapitel 6 in der Abbildung 6.11 werden alle zuvor beschriebenen Bewegungen in 3-D visualisiert.

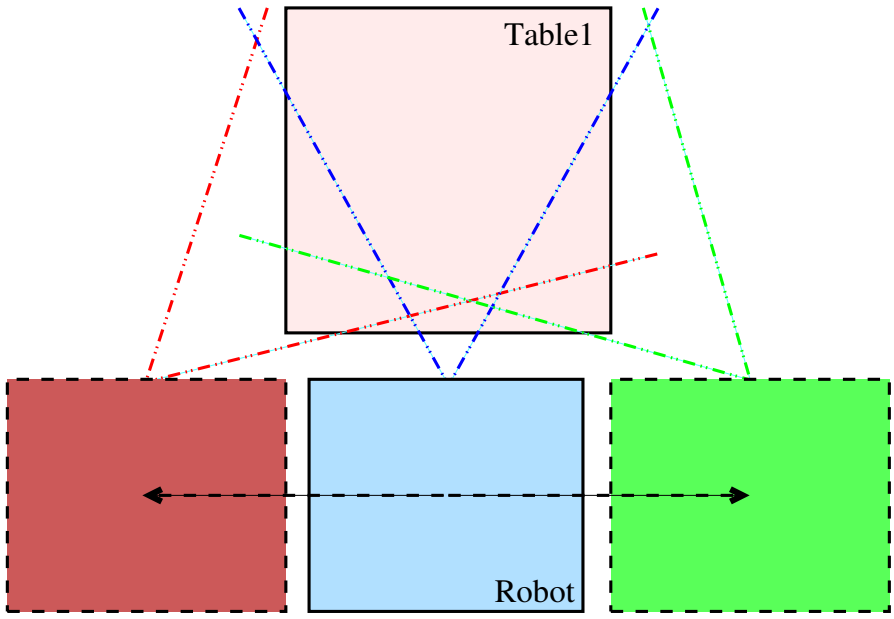


Abbildung 4.10.: Änderung der Perspektive durch die Bewegung des Roboters.

Die Bewegung des Roboters ist durch die bereits in ROS integrierte Komponente realisiert. Dabei kann auf die im Anhang D beschriebene Umgebungserfassung oder eine Karte, die zuvor durch einen Roboter, wie zum Beispiel den PR2, und die in ROS vorhandenen Algorithmen erstellt wurde, zurückgegriffen werden. Dabei bietet ROS die Möglichkeit, die Pfade in 2-D zu berechnen. Parallel findet die Kollisionsvermeidung in 3-D statt. Dadurch wird die Navigation in Bezug auf die Performanz optimiert, ohne das Sicherheitsrisiko zu erhöhen.

4.3.3.2. Aktiver Eingriff in die Szene

Eine weitere Innovation dieser Arbeit ist der aktive Eingriff in die Szene. Dieser wird im Fall noch vorhandener, aber auch nach der Bewegung des Torsos und der Plattform nicht klassifizierter Cluster ausgeführt. Ein anderer Auslöser für den aktiven Eingriff ist eine fehlende Übereinstimmung der Detektoren. Oft ist es ein Hinweis auf eine vorhan-

dene Verdeckung, wenn zum Beispiel ein Objekts durch einen Detektor sicher erkannt wird, seine volumetrischen Größen aber nicht mit der Datenbank übereinstimmen. Zur Auflösung der Verdeckung wird der Torso des Roboters nach oben gefahren, das im Vordergrund stehende erkannte Objekt oder Cluster gegriffen und im Manipulator behalten oder außerhalb der Szene abgestellt. Danach wird die Szene erneut analysiert. In dieser Arbeit werden die im Vordergrund stehenden Objekte und/oder Cluster nacheinander aus der Szene entfernt, wobei die Szene jedes Mal erneut analysiert wird. Die Annahme ist dabei, dass die im Vordergrund stehenden Objekte die Verdeckung verursachen. Hier können natürlich auch deutlich intelligentere Verfahren eingesetzt werden. Somit wird die Szene nach und nach vereinfacht in der Erwartung, alle beteiligten Objekte erkennen und die Szene komplett analysieren zu können.

Leider ist die Anzahl der vorhandenen und verlässlichen Detektoren, die innerhalb einer Verdeckung ein Objekt sicher erkennen können, extrem gering. Die auf den Farbeigenschaften basierten Detektoren sind wegen der ändernden Lichtverhältnisse unzuverlässig. Die volumetrischen Detektoren sind in einem Aspekt unzuverlässig, die Orientierung eines Objektes ist vor der Erkennung nicht bekannt. Zusätzlich sind diese für die Verwendung bei einer vorhandenen Verdeckung ungeeignet. Auch der ICP² basiert auf volumetrischen Objekteigenschaften und kann deswegen nicht eingesetzt werden. In der vorliegenden Arbeit wies nur der SIFT-/SURF-Algorithmus die benötigten Eigenschaften und die erforderliche Robustheit auf. Daher kann die Effizienz eines Eingriffs in die Szene nur Anhand dieses Detektors demonstriert werden. Die notwendige Bedingung für die Durchführung dieses Konzepts ist das „nicht Erkennen“eines Objekts. Diese Bedingung wurde bereits vorgestellt und basiert auf dem nötigen Konsens der Detektoren bzw. auf der Höhe der Wahrscheinlichkeit, mit der ein Objekt erkannt wurde. Ist ein Objekt nicht erkannt worden, wird, wie bereits erwähnt, basierend auf der Kostenbasis die Perspektive des Roboters auf die Szene geändert, vergleiche Kapitel 4.3.1 und 4.3.2. Bringt die zweifache Änderung der Perspektive keinen Erfolg und liefert der SIFT/SURF-Algorithmus ein positives Ergebnis, kann versucht werden, in die Szene einzugreifen.

Die nächste zu überwindende Hürde stellt die Berechnung der Griffe dar. Da das SIFT-/SURF-Verfahren 2-D ist, wird mithilfe der 3-D-Punktwolke die Position (Schwerpunkt im Bild) und die Orientierung des Objektes bestimmt. Falls die Datenbank mehrere Griffe für das erkannte Objekt bereitstellt, werden diese getestet, sonst können die Griffe neu kalkuliert werden. Ist einer der möglichen Griffe positiv bestimmt worden, kann mit dem Roboter in die Szene eingegriffen werden. Die erfolgreiche Ausführung des Griffes hängt natürlich von mehreren Parametern ab. So wird in ROS nur die Vor- und die eigentliche Griffposition und Orientierung mitberücksichtigt. Offenbart sich während der Transitphase zwischen der aktuellen Position des Arms und der Zielposition ein Problem, wie zum Beispiel eine oder mehrere Singularitäten, so wird das Greifen abgebrochen. Des Weiteren ist der Payload des PR2 auf 1.8 kg beschränkt, wird zusätzlich noch die Hebelwirkung berücksichtigt, wird die Anzahl der greifbaren Objekte stark reduziert.

Auch die Stabilität des Griffs ist ein nicht komplett gelöstes Problem der Informatik, so kann ein Objekt aus der Hand des Roboters herausgleiten. Des Weiteren können auch die kleinsten Fehler bei der Berechnung der Orientierung und Position des Objekts das Greifen negativ beeinflussen.

Verläuft das Eingreifen in die Szene erfolgreich, wird die Umgebung erneut analysiert. Falls das Objekt beziehungsweise die Objekte nicht erkannt werden oder die Verdeckung weiterhin besteht, wird der Kreis mit Änderung der Perspektive, Analyse und eventuellem Eingreifen noch einmal durchlaufen. In der vorliegenden Arbeit wird nach zwei erfolglosen Iterationen abgebrochen, und die Daten werden an den Operator gesendet (eine Message innerhalb von ROS). Es wäre interessant, die möglichen Abbruchkriterien und Verhaltensweisen des Roboters bzw. des Systems zu untersuchen, würde aber den Rahmen dieser Dissertation beim Weiten sprengen.

Falls ein erkanntes Objekt nicht gegriffen werden kann, besteht die Möglichkeit, die Objekte zu schieben und damit deren Position und Orientierung zu verändern. Innerhalb des RACE-Projektes, bei dem die Objekte weitere Teile wie Essen und/oder Flüssigkeiten beinhalten können, bleibt dieser Ansatz eher fraglich.

Abschließend soll betont werden, dass auch die Evaluation anhand der Ergebnisse eines einzigen Algorithmus extrem erschwert sei. Die Objektanordnung sowie die resultierenden Szenen wirken stark präpariert, viele interessante Möglichkeiten und Problemstellungen können gar nicht realisiert werden.

4.3.4. Regelbasiertes System

Regelbasierte Systeme stellen eine der primitivsten Formen der künstlichen Intelligenz dar. Dabei werden die Regeln als eine Art Wissensrepräsentation genutzt und sollen dabei helfen, Probleme zu lösen, die ein spezifisches Wissen erfordern. Im Vergleich zum reinen Wissen, das nur die Auskunft „wahr“oder „falsch“geben kann, bieten die regelbasierten Systeme Unterstützung bei der Entscheidungsfindung oder können diese sogar selber treffen [GA11].

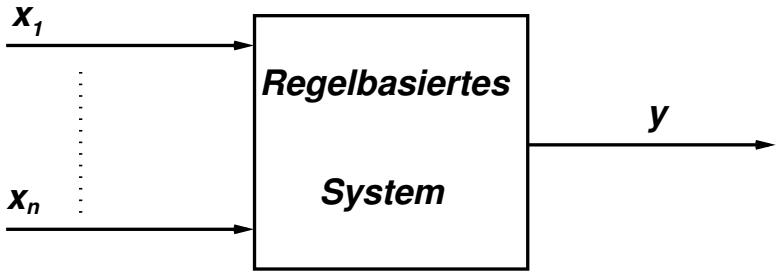


Abbildung 4.11.: Visualisierung eines regelbasierten Systems.

In der vorliegenden Dissertation bringen die Detektoren über die Sensoren die Daten,

also das Wissen, in das System ein. Diese Daten zu einer Entscheidung zusammenzufassen ist die Aufgabe eines regelbasierten Systems. Dieses soll aber auch helfen, wenn die Entscheidung nicht getroffen werden kann oder als unsicher betrachtet wird. Damit kann die im vorherigen Abschnitt beschriebene Innovation genutzt werden. Die verwendete Roboterplattform verändert die Perspektive der Sensoren auf die Szene oder greift sogar in diese ein, vgl. Abschnitt 1.4.

Normalerweise besteht ein regelbasiertes System aus einem System von Inferenzregeln und einem Inferenzschema, das die Verarbeitungsvorschrift enthält. Dabei werden die Eingangsgrößen x_i mithilfe der Inferenzregeln zu einer Ausgangsgröße y verarbeitet, vgl. Abbildung 4.11. Jede Inferenzregel entspricht einem Wissensteil des Typs „Wenn - dann“.

Damit ist das in den vorherigen Kapiteln präsentierte innovative Konzept komplett. Die Schleife von Wahrnehmung, Segmentierung, Objekterkennung, Änderung der Perspektive und/oder aktivem Eingriff in die Szene kann mehrmals durchlaufen werden. Sie soll nicht nur dabei helfen, komplizierte Szenen zu analysieren, sondern auch die Problematik der Verdeckung zumindest teilweise zu lösen.

4.3.5. Lokale Architektur

Am Ende dieses Kapitels soll, die in Kapitel 1 vorgestellte Architektur, verfeinert werden. Dabei wird die Objekterkennung, vorgestellt in der Abbildung 1.3, weiter unterteilt, wie die Abbildung 4.12 zeigt.

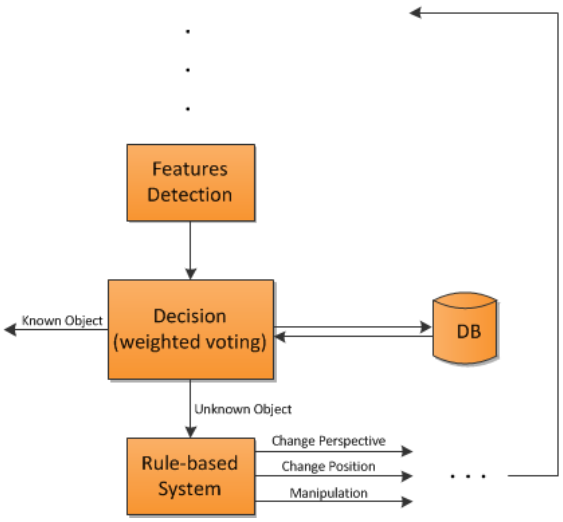


Abbildung 4.12.: Verfeinerung der, in Kapitel 1 vorgestellten Architektur.

Kann ein Objekt nicht eindeutig erkannt werden, ermöglicht das regelbasierte System

passende Reaktionen. So kann die Perspektive durch die Bewegung des Kopfes oder der kompletten Plattform verändert werden. Bringt auch das keinen Erfolg, greift die verwendete Plattform aktiv in eine Szene ein. Jeder dieser Schritte wird durch die erneute Analyse der Szene abgeschlossen.

Das regelbasierte System übernimmt die Daten der Detektoren, versucht, eine Entscheidung anhand der gewichteten Abstimmung zu treffen und ermöglicht, falls eine Entscheidung getroffen wurde, eine Reaktion auf diese Entscheidung. Besonderes in Fällen, wo die Entscheidung nicht getroffen werden kann, bietet das regelbasierte System die Möglichkeit, unterschiedlich zu reagieren und anschließend die Szenenanalyse erneut einzuleiten.

Die Realisierung mithilfe eines regelbasierten Systems stellt nur eine der Möglichkeiten dar. Es existieren durchaus mehrere Alternativen wie zum Beispiel eine State-Machine, eine Art Zustandsautomat, ähnlich dem, der für die Parallelisierung der Roboterbewegungen bei TAMS genutzt und in [EKRZ13] veröffentlicht worden ist.

Damit sind alle notwendigen Komponenten beschrieben und diskutiert worden. Im nächsten Kapitel wird auf die technische Realisierung des vorliegenden Dissertationsvorhabens eingegangen.

Kapitel 5

Technische Realisierung

Basierend auf den Ideen und Neuerungen der vorliegenden Arbeit, wird in diesem Kapitel die Architektur sowie die resultierende Implementierung vorgestellt und diskutiert. Dementsprechend wird dieses Kapitel in zwei grundlegende Abschnitte unterteilt. Der erste Abschnitt „Architektur“beschäftigt sich mit der Entwicklung der eigentlichen Architektur. Der zweite Abschnitt beschreibt die darauf basierende Implementierung.

5.1. Architektur

Die Architektur stellt einen wichtigen und notwendigen Meilenstein auf dem Weg zur Realisierung dar. Dieser Schritt ermöglicht die Visualisierung und die erste Evaluation des resultierenden Systems. Werden dabei einige Fälle nicht berücksichtigt, wird die Implementierung dadurch erschwert, wenn nicht sogar unmöglich gemacht. Aus diesem Grund wird in der vorliegenden Dissertation versucht, ingenieurtechnisch vorzugehen und die Endarchitektur nach diesen Maßstäben zu entwerfen.

5.1.1. Systemübersicht

Begonnen wird mit einer klassischen Systemübersicht in Form eines USE-CASE-Diagramms (Unified Modeling Language (UML)). Dabei wird zuerst das System als eine allein stehende Komponente betrachtet. Später wird diese in eine typische, deutlich umfangreichere Architektur integriert. Die Abbildung 5.1 visualisiert das System als eine einzelne Komponente. Das System wird von außen angeregt, hier durch ein Akteur/User realisiert. Alle weiteren Schritte laufen automatisch ab und benötigen keinen weiteren Eingriff von außen. Dabei werden folgende Schritte durchlaufen, auf die später noch ausführlich eingegangen wird: Zuerst wird die eingehende, bereits fusionierte Punktwolke segmentiert und so weit wie möglich in Cluster unterteilt. Danach wird der Erkennungs-

vorgang gestartet, der die Daten für den notwendigen Vergleich aus einer bereitsvorhandenen Datenbank lädt. Dabei werden die Ergebnisse mehrerer Detektoren gesammelt.

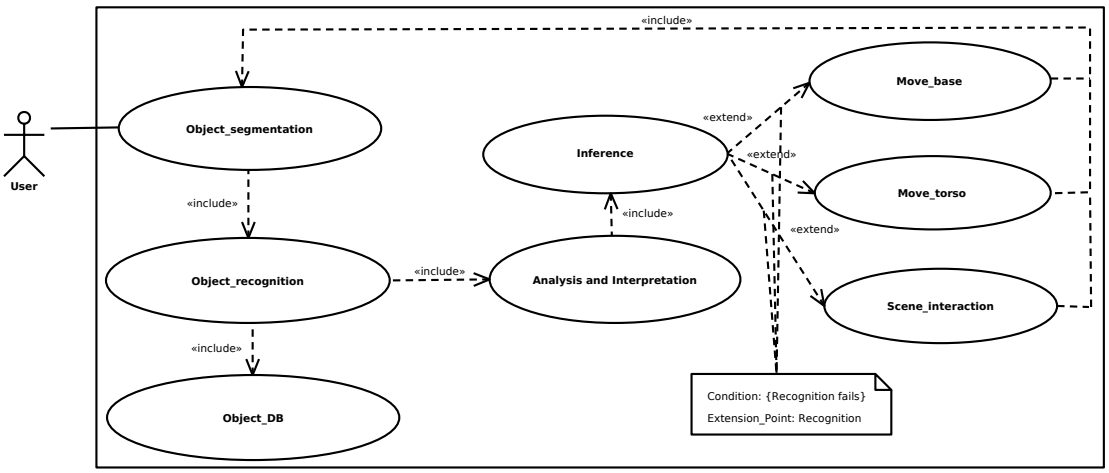


Abbildung 5.1.: USE-CASE-Diagramm des resultierenden Systems.

Diese Daten werden im Folgenden analysiert und interpretiert. Dieser Schritt ist durchaus mit einer intelligenten Fusion vergleichbar, da die ausgegebenen Daten einzelner Detektoren zu einem gemeinsamen Ergebnis verarbeitet werden. Ist das Ergebnis dieses Schritts eindeutig, werden die Resultate ausgegeben, und die Anwendung wird erfolgreich terminiert. Sind die Ergebnisse einzelner Detektoren nicht eindeutig, liegt entweder ein unbekanntes Objekt vor oder die Szene weist eine partielle Verdeckung auf. Daher erfolgt ein aktiver Eingriff des Roboters in die Szene. Dabei stehen mehrere Alternativen zur Verfügung: Es kann nach dem Prinzip des „aktiven Sehens“die Perspektive durch die Bewegung des Kopfs, des Torsos oder der Plattform verändert werden. Die Wahl der Alternativen erfolgt nach dem Kostenprinzip, die Bewegung des Torsos kostet mehr als die des Kopfes und weniger als die des kompletten Roboters. Sind diese Mittel ausgeschöpft und bringen dennoch keinen Erfolg, greift der Roboter in die Szene aktiv ein, basierend auf der im Kapitel „Szenenanalyse und Interpretation“erarbeiteten Vorgehensweise. Es ist ersichtlich, dass der gesamte Ablauf nach jeder Änderung der Perspektive oder nach dem Eingriff erneut ausgeführt werden muss. Werden alle Mittel ausgeschöpft, ist die Interaktion mit dem Benutzer und dessen Eingriff unvermeidlich.

5.1.2. Integration und Entwicklung

Ausgehend von dieser groben Skizze, wird in diesem Abschnitt zuerst die Integration der Anwendung in eine deutlich umfangreichere Architektur betrachtet. Da die vorliegende

Arbeit in das EU-Projekt RACE integriert und von diesem stark inspiriert ist, orientiert sie sich an dessen Architektur. Die Abbildung 5.2 stellt die Gesamtarchitektur des Projekts RACE grafisch dar.

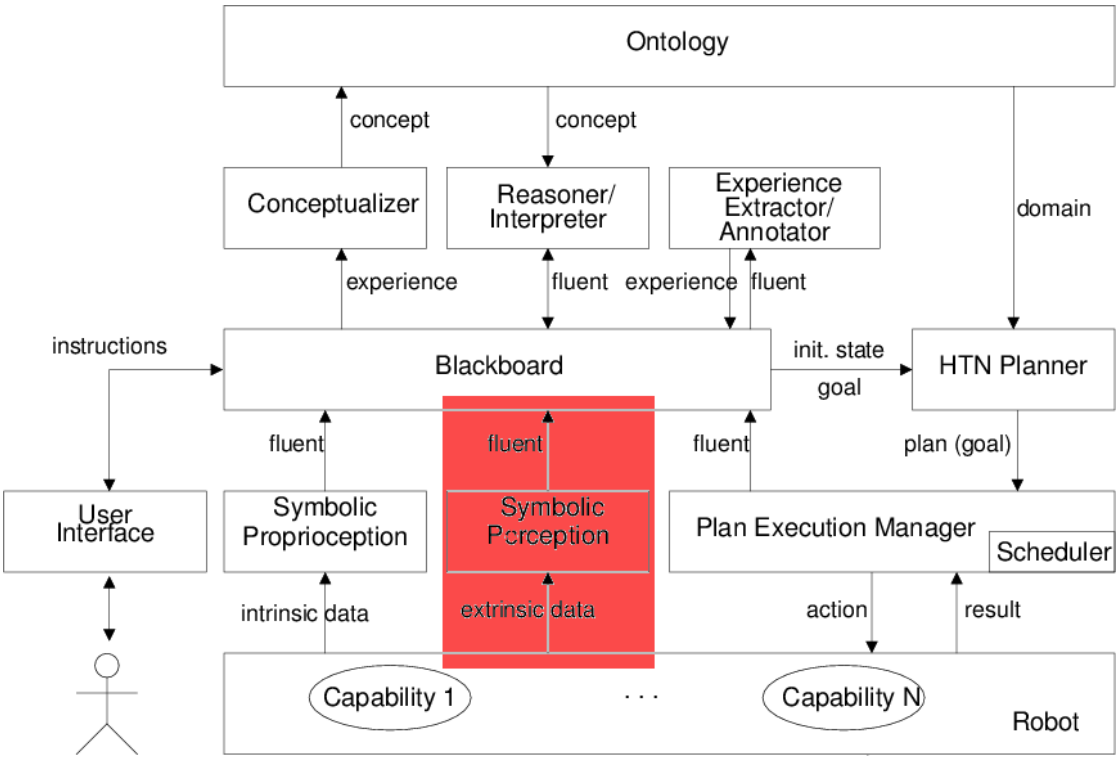


Abbildung 5.2.: Visualisierung der Architektur des EU-Projekts RACE. Der Bezug zur vorliegenden Dissertation ist rot markiert.

Wie bereits beschrieben, wird ein Roboter in dem oben genannten Projekt als Kellner eingesetzt. Dabei stehen die Bedienung des Kunden und das spätere Abräumen des Geschirrs im Vordergrund. Das Ziel ist es, dass der Roboter aus den eigenen Erfahrungen lernt und das daraus resultierende „intelligente“Verhalten aufgezeichnet werden kann. Die Objekterkennung ist eine notwendige und essenzielle Komponente dieses Systems. Ohne die eindeutige Zuweisung eines Objekts zu einer bestimmten Kategorie würde das Erreichen der Projektziele enorm erschwert, wenn nicht schlichtweg unmöglich gemacht.

Der Schnittpunkt der vorliegenden Dissertation mit der RACE-Architektur ist in der Abbildung 5.2 rot hervorgehoben. Werden die notwendigen Schnittstellen betrachtet, greift das Objekterkennungssystem auf die Sensordaten einer Roboterplattform zurück und liefert die Wahrscheinlichkeiten der Zugehörigkeit eines oder mehrerer Objekte zu

einer oder mehreren Objektkategorien an das Blackboard zurück. Basierend auf dem USE-CASE-Diagramm und der RACE-Architektur kann die Architektur der vorliegenden Arbeit präzisiert werden. Die Abbildung 5.3 stellt die funktionale Architektur der vorliegenden Dissertation ausführlich dar.

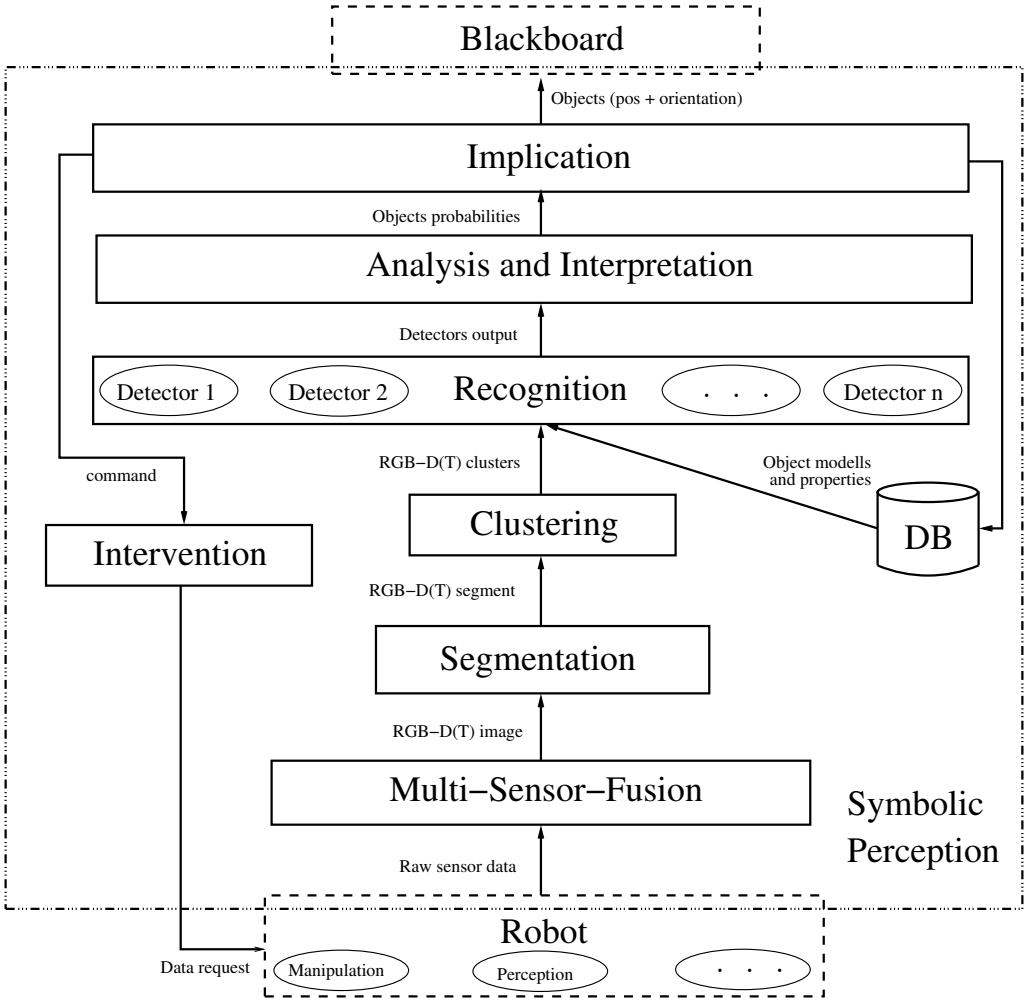


Abbildung 5.3.: Visualisierung der funktionellen Architektur der vorliegenden Dissertation angelehnt an die Architektur des EU-Projektes RACE. Die Zeichnung visualisiert den in der Abbildung 5.2 rot markierten Bereich.

Dabei wurden zum besseren Verständnis die vom Roboter zur Verfügung gestellten Ressourcen und das Blackboard mit eingezeichnet. In der Regel wird die funktionelle Architektur nach dem Bottom-up-Prinzip betrachtet. Die von einer Roboterplattform

stammenden Sensordaten werden vorverarbeitet und fusioniert. In einem weiteren Schritt wird die Tischoberfläche detektiert, und alle darüber liegenden Voxel werden segmentiert. Danach wird das resultierende Segment, abhängig von einem auf der euklidischen Distanz basierenden Schwellenwert, in Cluster unterteilt.

Sind die Daten in der beschriebenen Art und Weise vorverarbeitet, folgt der Schritt, der den Hauptgegenstand dieser Arbeit darstellt. Zuerst werden mehrere Detektoren, die auf unterschiedlichen Objekteigenschaften operieren, parallel angewandt. Optimal einsetzbar wären die Detektoren, die alle Objekteigenschaften, nämlich Form, Textur, Farbe, Größe, markante Merkmale und Bewegung (vgl. Kapitel 1.3), nutzen. Für den notwendigen Abgleich werden die Daten einer Datenbank genutzt, abhängig von den eingesetzten Detektoren sind das 3-D-Modelle, Bilder, Merkmale, Textur, Farbinformationen, etc.

Die Kombination der Ergebnisse von einzelnen Detektoren kann auf unterschiedlichen Informationsebenen geschehen. Die Verwendung der Multi-Sensor-Fusion nutzt die Vorteile der Kombination auf der Merkmalsebene bereits teilweise aus, somit liegt es nahe, die Detektoren erst einzeln anzuwenden und sie danach, auf der Entscheidungsebene zusammenzufassen. Die Kombination auf der Entscheidungsebene begünstigt die Verwendung der parallelen Architektur, zumindest während der Detektorenanwendung ([Kun04]). Das restliche System folgt einer typischen hierarchischen Architektur.

Die ausgegebenen Daten einzelner Detektoren werden im nächsten Schritt analysiert und interpretiert. Sind die Ergebnisse eindeutig, terminiert die Anwendung. Dabei werden die Wahrscheinlichkeiten für das jeweilige Cluster an das Blackboard weitergereicht. Sind sie nicht eindeutig, werden basierend auf den Daten der Detektoren weitere Schritte eingeleitet, wie bereits oben beschrieben. Dadurch wird die verwendete Roboterplattform aktiv in den Erkennungsprozess involviert. Dabei wird die Erkennungsschleife immer wieder durchlaufen. Sind die Ergebnisse eindeutig, folgt die Terminierung, anderenfalls werden die Daten an den Benutzer weitergegeben und bei Bedarf in die Datenbank eingefügt, falls es sich zum Beispiel um unbekannte Objekte handelt. Des Weiteren wird im Falle eines Scheiterns einzelner Komponenten auf dem Weg zur Erkennung die Schleife immer wieder durchlaufen.

Somit basiert die Erkennung auf mehreren heterogenen Detektoren sowie dem aktiven Eingriff der Roboterplattform in ihre Umgebung. Das erlaubt nicht nur die Verbesserung der Objekterkennung, sondern auch die Auflösung eventuell vorhandener partieller Verdeckungen. Nach Bedarf kann auf die Ergebnisse aller Detektoren gewartet, oder bei den schnellen eindeutigen Ergebnissen bereits davor terminiert werden. Das resultierende Erkennungssystem ist in sich geschlossen und kann als Stand-Alone-Komponente beliebig eingesetzt werden. Wie bereits in Kapitel 4 dargestellt, können weitere Detektoren eingebunden werden, was die Erweiterbarkeit des Systems garantiert. Durch die Anpassung der Schnittstellen, die Sensordaten als Eingabe und die Objektwahrscheinlichkeiten als Ausgabe, kann das System mit wenig Aufwand in eine beliebige Architektur integriert werden. Durch die mögliche Anpassung der Formate für die Eingabedaten (hier ROS-

Messages) ist das System auch unabhängig von der verwendeten Roboterplattform. Die Robustheit bleibt dabei erhalten und sollte nur nach der Integration weiterer Detektoren erneut überprüft werden.

5.1.3. Kombination der Detektoren

Jeder Detektor hat eigene, teilweise ganz spezifische Vor- und Nachteile, daher existieren unterschiedliche Verfahren, um die Detektoren miteinander zu kombinieren. Grundsätzlich ist die Erwartung an diese Art von Kombination, dass die Fehlklassifikationen eines Detektors durch die anderen Klassifikatoren korrigiert werden können. In der Praxis erwies sich die Kombination der Detektoren als ein verlässliches Mittel zur Steigerung der Detektorenperformanz. Dabei treten einige Fragen auf, die zuerst beantwortet werden müssen: Auf welcher Ebene erfolgt die Fusion beziehungsweise Kombination? Welche Systemarchitektur und Kombinationsmethoden sollen genutzt werden? Wie sollen die einzelnen Detektoren für eine bessere Kombination beschaffen sein? Hier wird ein kurzer Überblick gegeben sowie in der vorliegenden Dissertation gewählte Weg skizziert.

Da in dieser Arbeit mehrere unterschiedliche Sensorarten genutzt werden, erfolgt bereits auf dem Level der Sensordaten eine Multi-Sensor-Fusion. Daher bietet es sich an, dass zuerst einzelne Detektoren mit individuellen - hier mit fusionierten und bereits vorverarbeiteten - Daten und Merkmalen arbeiten und ein Ergebnis erzielen. Danach werden die Ergebnisse einzelner Detektoren verglichen und gemeinsam ausgewertet. Dementsprechend findet die Kombination der Detektorenergebnisse auf der Entscheidungsebene statt.

In Bezug auf die Architektur kommen drei Varianten infrage: die serielle, die parallele und die hierarchische Architektur. Die serielle Architektur wird eher zur Steigerung der Verarbeitungsgeschwindigkeit eingesetzt, das heißt, dass im Normalfall nie alle Klassifikatoren zum Einsatz kommen.

Die Detektoren werden nach ihrer Komplexität sortiert und in dieser Reihenfolge angewendet. Meldet einer der Klassifikatoren einen Erfolg, wird an dieser Stelle unterbrochen und mit dem Ergebnis weitergearbeitet. Bei vielen unterschiedlichen Klassen wird eine hierarchische Architektur verwendet. Dabei dienen die Klassifikatoren, die sich nicht im Blattknoten befinden, dazu, die Teilmengen weiter zu selektieren. In dieser Arbeit sollen aber alle Detektoren eingesetzt werden, da sie mit verschiedenen Objekteigenschaften arbeiten. Auch Xie Tian und Zhang [XTZ13] zeigen, dass die besten Ergebnisse durch die Verwendung mehrerer unterschiedlicher Merkmale und Detektoren zu erreichen sind. Das Ergebnis soll also eine Fusion der einzelnen Ergebnisse dieser Detektoren sein, daher wird in der vorliegenden Arbeit die am häufigsten verwendete parallele Architekturform genutzt. Dementsprechend werden alle Detektoren parallel angewendet und die einzelnen Resultate als ein gemeinsames Ergebnis in einem Kombinationsmodul zusammengefasst.

Im Weiteren werden die Detektoren nach Kuncheva [Kun04] in drei unterschiedliche Typen unterteilt. Es wird angenommen, dass unterschiedliche Detektoren $D_1, ..., D_l$ so-

wie verschiedene Klassen $K_1, ..., K_n$ existieren.

- Typ 1: Die Ausgabe besteht aus exakt einer Klasse K_i; die SIFT-/SURF-Merkmalsdetektoren sind ein gutes Beispiel für diesen Typ.
- Typ 2: Die Ausgabe besteht aus einer Rangliste $(K_{i_1}, ..., K_{i_j})$, wobei $1 \leq j \leq n$, von Kandidaten mit abnehmender Plausibilität (K_{i_1} hat die höchste und K_{i_j} die niedrigste Plausibilität). Ein Beispiel sind die auf dem Entscheidungsbaum basierenden Detektoren.
- Typ 3: besteht aus einem Plausibilitätswert $s(K_i)$ für jede Klasse K_i. Ein Beispiel ist der hier verwendete ICP^2-Detektor, der die Klasse sowie eine dazugehörige (Ähnlichkeits-)Wahrscheinlichkeit liefert.

Schon aus dieser Auflistung wird klar, dass in der vorliegenden Arbeit unterschiedliche Detektoren-Typen zum Einsatz kommen. Die klassischen Kombinationsmethoden sehen an dieser Stelle das Weglassen der überflüssigen Informationen vor, indem die Ergebnisse der Typ-3- in solche der Typ-2- und die Ergebnisse der Typ-2- in solche der Typ-1-Detektoren umgewandelt werden. Hier wird eher eine entgegengesetzte Richtung eingeschlagen, indem versucht wird, dem einzelnen Ergebnis eines Detektors eine Wahrscheinlichkeit zuzuordnen. Am Beispiel von SIFT/SURF-Detektoren kann die Anzahl der Merkmale im Originalbild zur Menge der gefundenen Merkmale in Relation gesetzt werden.

Für den Verzicht auf das Weglassen von Informationen sind zwei Gründe zu nennen. Einerseits wurde in der vorliegenden Arbeit der Versuch unternommen, so viele Informationen wie möglich zusammenzutragen. Andererseits unterliegt die Gesamtarchitektur als Teil eines EU-Projektes ständigem Wandel, sodass permanente Anpassungen unvermeidbar sind. Aus dieser Sicht ist es vorteilhafter, zunächst alle möglichen Informationen zur Verfügung zu stellen, ein späteres Weglassen der Informationen ist dann einfacher zu realisieren, als die Schnittstellen eventuell komplett neu entwerfen zu müssen.

Nach der Anwendung aller Detektoren werden die Ergebnisse als ein kombinierter Klassifikator (Merkmalsvektor) dem System zur Verfügung gestellt. Die nächste Frage ist dann, wie diese gemeinsam ausgewertet werden können. Wie schon beschrieben, nutzen divergente Detektoren verschiedene Objekteigenschaften und sind unterschiedlich verlässlich. Daher liegt die Entscheidung, die gewichtete Abstimmung zu nutzen, relativ nahe. Wird die Nomenklatur benutzt, die bereits oben eingeführt worden ist, wird jedem Detektor D_i ein Gewicht w_i zugeordnet. Die Entscheidung erfolgt für diejenige Klasse K_m, die die maximale Summe der Gewichte erhält. Für die Bestimmung der Gewichte kann zum Beispiel die Erkennungsrate an einer Trainingsmenge genutzt werden. Des Weiteren kann ein Optimierungsverfahren hinzugezogen werden. Die Gewichte sind so anzupassen, dass der Fehler bei der Trainingsmenge bzw. Trainingsstichprobe minimiert wird. Anhand des Merkmalsvektors wird dann durch ein regelbasiertes System eine Entscheidung getroffen, und falls notwendig werden weitere Aktionen eingeleitet.

Im nächsten Kapitel wird auf die charakteristische Realisierung des in dieser Arbeit verwendeten Prototyps eingegangen. Dabei wird die funktionelle Architektur weiter spezifiziert sowie auf die Implementierung einzelner Bestandteile ausführlich eingegangen.

5.2. Implementierung

Nachdem die funktionelle Architektur entwickelt und diskutiert worden ist, wird mit der Beschreibung der Implementierung des, in dieser Arbeit verwendeten, Prototyps fortgefahren.

Der entworfene und realisierte Prototyp soll die Ideen dieser Arbeit kritisch prüfen und die Möglichkeiten der Evaluation offenbaren. Daher wurde nicht ein marktreifes Produkt, sondern ein breit angelegter Prototyp entwickelt, der das Potenzial und die Chancen der vorliegenden Dissertation verdeutlicht und visualisiert. Die meisten Kernkomponenten sind in Eigenimplementierung entstanden, andere wurden aus bereits existierenden Komponenten übernommen oder angepasst. Das Ziel ist es, möglichst viele in ROS existierende Methoden und Algorithmen zu verwenden, sodass auf der einen Seite das eigene ROS-Package möglichst schlank bleibt und es auf der anderen Seite einfacher von anderen ROS-Nutzern verwendet oder weiterentwickelt werden kann. Werden die externen Komponenten verbessert und bleiben die Schnittstellen nach dem ROS-Prinzip erhalten, so profitieren alle, diese Funktionalität nutzenden Module automatisch ohne weitere Anpassungen. Auch im umgekehrten Fall profitiert die ROS-Gemeinschaft.

Wie bereits erwähnt, wird ROS[1] als Framework eingesetzt und beeinflusst damit die Realisierung sowie die Kommunikation. Eine Visualisierung aller beteiligten Nodes und Transformationen wird in Appendix A.1 gegeben. Die Hauptfunktionalität ist als ein ROS-Package gekapselt, das mit anderen Packages kommuniziert und mehrere Nodes, die teilweise als Topics und teilweise als Services auftreten, beinhaltet.

Die Kommunikation zwischen den ROS-Elementen wird über einen sogenannten ROS-Master organisiert. Der ROS-Master verfügt über die Informationen aller im System laufenden Nodes. Dafür ist der Master als Node realisiert und wird als erster Node von ROS gestartet. Alle Nodes, die gestartet werden, melden sich beim ROS-Master und geben Auskunft über benötigte und zur Verfügung gestellte Informationen. Damit das System beschleunigt wird, findet ein direkter Informationsaustausch zwischen den Nodes statt. Der ROS-Master fungiert dabei als Vermittler und ist an der späteren Kommunikation nicht beteiligt. Das grundlegende Konzept ist in [2] visualisiert und beschrieben.

Alle Algorithmen und Methoden sind in C/C++ implementiert. Im Sinne der oben vorgestellten Idee werden die bereits in ROS integrierten Komponenten verwendet. So werden für das Akquirieren der Kamerabilder und für die Bildverarbeitung und Ope-

[1] http://wiki.ros.org/
[2] http://wiki.ros.org/de/ROS/Concepts

rationen mit den Matrizen die Bibliotheken der OpenCV eingesetzt[3]. Für die 3-D-Algorithmen und deren Visualisierung wird die Point Cloud Library[4] (PCL) verwendet. Dennoch werden die meisten vorgestellten Abbildungen mit einem ROS-eigenen Werkzeug namens RVIZ[5] visualisiert.

Die Datenrepräsentationsstruktur basiert auf der in ROS bereits existierenden PointCloud2-Struktur. Wie bereits in Kapitel 3 beschrieben, soll die Struktur gleichzeitig alle Daten beinhalten sowie flexibel einsetzbar sein, z.B. für 2-D-Detektoren, aber auch für 3-D-Detektoren oder SIFT-/SURF-Features. Dabei sollen nur die notwendigen Daten einem Detektor zur Verfügung gestellt werden. Der Aufbau besteht aus Koordinaten und Farben und kann wie folgt schematisch dargestellt werden: $[x_1, y_1, z_1, r_1, g_1, b_1, ..., x_n, y_n, z_n, r_n, g_n, b_n]$. Die Struktur ist generisch und kann über das zur Verfügung stehende Interface angepasst und/oder erweitert werden.

Natürlich kamen auch viele andere Werkzeuge zum Einsatz. Der Autor bemüht sich, alle Anwendungen und Werkzeuge an den jeweiligen Stellen zu nennen und zu referenzieren.

5.2.1. Ergänzende Komponenten sowie deren Konfiguration und Integration

In den folgenden zwei Abschnitten werden die ergänzenden Komponenten vorgestellt, die in dieser Arbeit Verwendung fanden. Dabei wird auch deren Einbindung in die vorgestellte Architektur präsentiert.

5.2.1.1. Datenbank

Eine der notwendigen ergänzenden Komponenten ist die Datenbank, die eine permanente Speicherung aller für die Objekterkennung notwendigen Objekteigenschaften erlaubt. In der vorliegenden Arbeit wird die PostgreSQL-Datenbank (in den Versionen 8.4 und 9.1) verwendet [EH10]. Die PostgreSQL wird seit den 1980er-Jahren entwickelt und ist seit 1997 Open Source. PostgreSQL orientiert sich weitgehend am SQL-Standard (ANSI-SQL 2008) und ist vollständig ACID-konform.

Die PostgreSQL ist sehr mächtig und benötigt dementsprechend viele Ressourcen. Natürlich werden hier bei Weitem nicht alle Möglichkeiten von PostgreSQL ausgeschöpft. Dennoch stellt die Datenbank durch ihre Verfügbarkeit unter Ubuntu, ihren offenen Quellcode und die vorhandene Integration in ROS eine perfekte Wahl für die vorliegende Dissertation dar.

Für die administrative Zwecke wird PGAdmin3[6] verwendet, ein sehr populäres, grafisches Tool, das nicht nur die Pflege, sondern auch die Manipulation und Erweiterung

[3] http://opencv.org/
[4] http://pointclouds.org/
[5] http://wiki.ros.org/rviz
[6] http://www.pgadmin.org/

einer angelegten Datenbank erlaubt. Dadurch ist ebenfalls die problemlose Verwendung mehrerer Datenbanken und die Umschaltung zwischen verschiedenen Versionen möglich.

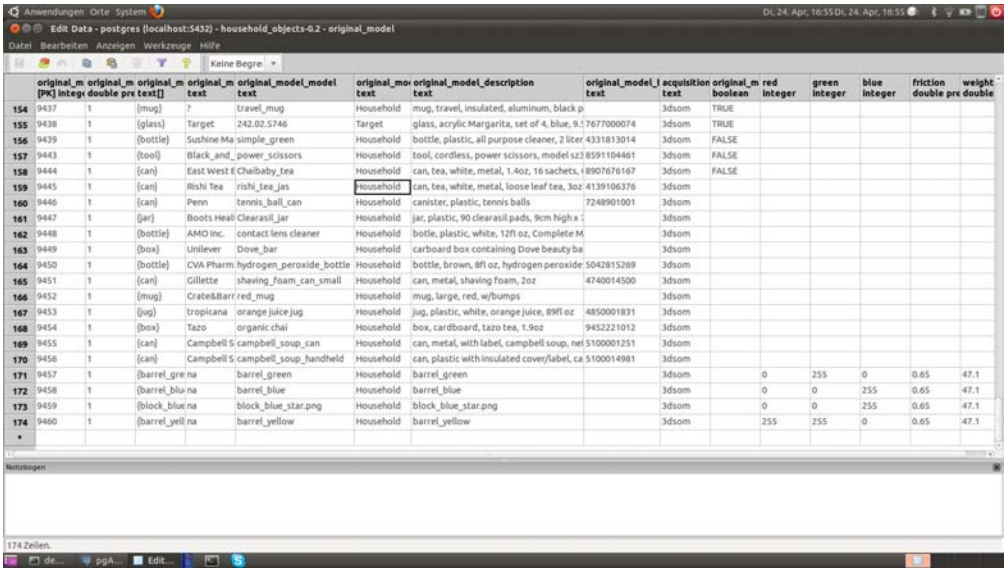


Abbildung 5.4.: Visualisierung der verwendeten Datenbank mit zusätzlich integrierten Objekteigenschaften.

Das verwendete Datenbankschema basiert auf dem bereits in ROS existierenden und für die Objekterkennung (IDF) genutzten Stack namens „household_objects“[7] mit bereits mehr als 300 integrierten Objekten. Das vorhandene Schema ist durch weitere notwendige Objekteigenschaften, wie zum Beispiel Farbinformationen, Textur, Gewicht und Klasse, erweitert worden. Die Abbildung 5.4 visualisiert eine der verwendeten Datenbankvarianten mithilfe des PGAdmin3. Die beschriebene Erweiterung durch verschiedene Objekteigenschaften ist partiell auf der linken Seiten zu sehen.

Damit stehen alle Möglichkeiten zur Verfügung, den Vergleich der Ergebnisse mehrerer Detektoren mit der Datenbank durchzuführen. Für den aktiven Eingriff eines Roboters in die Szene werden zudem die davor berechneten Griffe benötigt. Diese stellen dabei einen permanenten Bestandteil der verwendeten Datenbank dar. Der nächste Abschnitt geht auf die Berechnung und die Integration der Griffe in die Datenbank ein.

5.2.1.2. Berechnung der Griffe

Sind die notwendigen Objektinformationen in der Datenbank gespeichert, können die existierenden 3-D-Modelle zur Berechnung der möglichen Griffe verwendet werden. Die

[7] http://www.ros.org/wiki/household%20objects

Griffe werden benötigt, damit die Objekte durch einen Roboter manipuliert werden können. In der vorliegenden Arbeit wird das Programm GraspIt![8] des Robotik-Labs der Columbia-Universität verwendet.

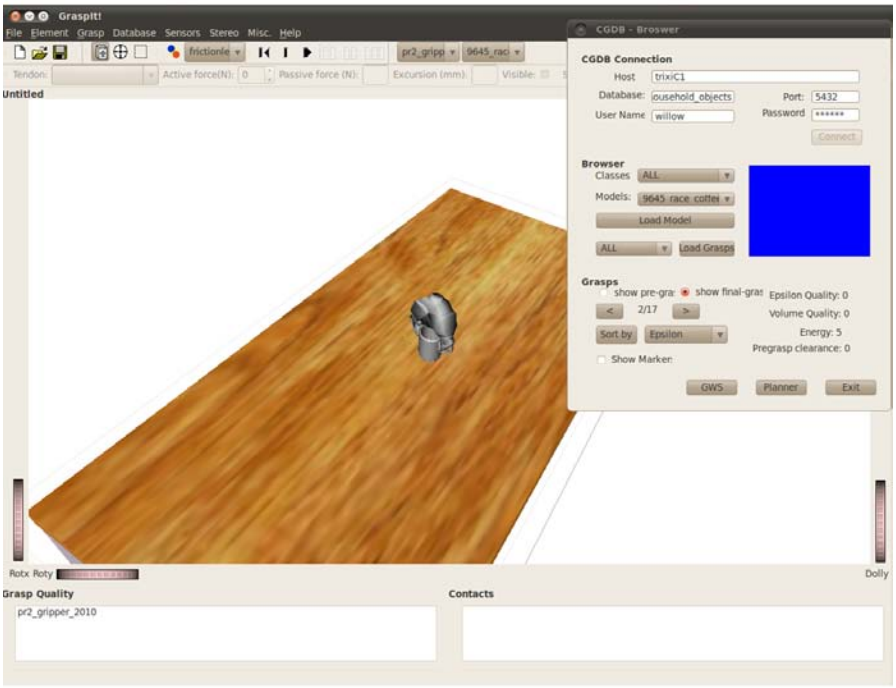


Abbildung 5.5.: Berechnung der möglichen Griffe mit einem GraspIt!-Simulator.

Die Simulation erhält die 3-D-Daten eines Objekts sowie die Parameter der genutzten Roboter-Tools und liefert alle möglichen Griffe. Durch die GUI können die Griffe einzeln eingesehen, dem Szenario entsprechend ausgewählt und in der Datenbank dauerhaft gespeichert werden.

Abbildung 5.5 visualisiert die Griffkalkulation für eine Tasse und den PR2-Gripper durch GraspIt!. Wie aus der Abbildung entnommen werden kann, können zusätzlich Hindernisse, wie in diesem Fall ein Tisch, geladen werden. Somit können die Griffe an die späteren realen Szenarien angepasst werden. Da die Position und die Orientierung des Manipulators dabei nicht berücksichtigt werden, müssen mehrere Griffe präsent sein. Zur Laufzeit, abhängig von den Positionen des Roboters und des Manipulators, werden dann die passenden Griffe gewählt und abhängig von den Energieeffizienzkriterien ausgeführt. Dabei können dieselben Griffe in der Simulation und in der Realität genutzt werden. In der vorliegenden Arbeit wurde die GraspIt!-Simulation in Version 2.3, die be-

[8] http://www.cs.columbia.edu/ cmatei/graspit/

reits in ROS integriert ist, verwendet. Durch die Integration in ROS können 3-D-Objekte direkt aus der Datenbank geladen und die bereits vorhandenen Informationen durch die berechneten Griffe ergänzt werden. Nach der Berechnung werden die Pre- und Greifpositionen in der Datenbank persistent gespeichert, die physikalischen Randbedingungen werden dabei mitberücksichtigt.

Die Kalkulation der Griffe erlaubt einem Roboter im Rahmen dieser Dissertation, faktisch aktiv in die Szene einzugreifen und diese so zu manipulieren, dass nach der erneuten Anwendung der Detektoren alle beteiligten Objekte erfolgreich erkannt werden können.

Kapitel 6

Evaluation

Dieses Kapitel stellt die Evaluation der vorliegenden Dissertation dar. Nach Madhavan, Tunstel und Messina [MTM09] existieren zwei unterschiedliche Methoden, um ein System zu evaluieren: einerseits die Measures of Effectiveness (MoE), die ein System in einer bestimmten Umgebung quantitativ beurteilen, und andererseits die Measures of Performance (MoP), mit denen einzelne Bestandteile des Systems qualitativ getestet werden. Es wird versucht, durch mehrere Experimente die Vor- und Nachteile des hier entwickelten Systems aufzuzeigen und zu interpretieren. Dabei wird auf das komplette System sowie auf die einzelnen Komponenten eingegangen und versucht, deren Einfluss für die gesamte Objekterkennung festzustellen und zu analysieren (MoE und MoP). Doch zunächst soll die Evaluationsumgebung vorgestellt werden. Ein Roboter ist eine endliche Ressource, reale Tests sind deshalb entsprechend lang, müssen permanent überwacht werden und sind kaum exakt zu wiederholen. In dieser Arbeit wurde entschieden, die meisten Tests in der Simulation durchzuführen. Einige signifikante Beispiele sind danach zwecks Vergleich und Auswertung auf dem realen Roboter getestet worden. Daher stellt die Simulation ein wichtiges Evaluationswerkzeug dar und wird im nächsten Abschnitt vorgestellt und ausführlich beschrieben.

6.1. Evaluationsplattform und Framework

Wie bereits erwähnt, wird in dieser Arbeit ein PR2-Roboter[1] der Firma WillowGarage[2] verwendet. Der Roboter ist derzeit einer der am weitesten entwickelten Serviceroboter der Welt. In der Abbildung 6.1 ist der Roboter des Arbeitsbereichs TAMS der Universität Hamburg zu sehen. Dieselbe Plattform wird auch zur Evaluation im Rahmen des EU-Projekts RACE eingesetzt.

[1] http://www.willowgarage.com/pages/pr2/overview
[2] http://www.willowgarage.com/

Abbildung 6.1.: Die im Rahmen der Dissertation verwendete Evaluationsplattform: ein PR2-Roboter der Firma WillowGarage.

Der Roboter basiert auf einer Omnidrive-Plattform, ist dadurch extrem wendig und erreicht eine maximale Geschwindigkeit von ca. 1 m/s. Des Weiteren ist der Roboter mit zwei 7-DOF-Armen und jeweils einem Zwei-Finger-Greifer ausgestattet. Jeder der Finger beinhaltet 21 Drucksensoren, die bei einer Greifbewegung zur Überprüfung von deren Stabilität herangezogen werden. Der Torso kann um ca. 35 cm nach oben gefahren werden, damit erreicht die Plattform eine Gesamtgröße von ca. 1645 mm. Der Kopf, in dem die meisten Sensoren Platz finden, kann unabhängig in drei Richtungen bewegt werden. Als Sensoren stehen zwei Laserscanner, ein 2-D- und ein 3-D-Scanner, realisiert über eine Schwenkeinheit, sowie mehrere Kameras, ein Infrarotprojektor und zwei Stereokamerasysteme zur Verfügung. In dieser Konfiguration produziert die Plattform ca. 50 MB/s an Daten. Zusätzlich ist der hier verwendete Roboter mit einem PrimeSense-Sensor von ASUS und einer FLIR-IR-Kamera (vgl. Appendix A.2) ausgestattet, beide sind jeweils auf und am Kopf montiert. Diese Sensoren liefern zusätzlich 30 fps an Farb- und Tiefenbildern sowie ein Einkanal-Infrarotimage. Für weitere Spezifikationsdetails sei der Leser auf die Internetseiten der WillowGarage[3] verwiesen.

Als Framework wird ROS[4] (Akronym für Roboteroperationssystem, engl. für „Robot Operation System“) in der Version Fuerte verwendet. Das Framework bietet nicht nur die Infrastruktur und Kontrollelemente für den PR-2, sondern auch viele bereits implementierte Algorithmen und Methoden.

Die Infrastruktur besteht aus einem Master-Knoten, der für die Kommunikation zwischen einzelnen Prozessen (engl. „nodes“) sorgt. Dabei meldet sich jeder Node während

[3] http://www.willowgarage.com/pages/pr2/specs
[4] http://wiki.ros.org/

des Starts beim Master an und gibt eine Auskunft über die bereitgestellten Daten oder meldet die Abhängigkeiten und Daten, die benötigt werden. Die anschließende Kommunikation läuft direkt zwischen voneinander abhängigen Prozessen ohne Beteiligung des ROS-Masters ab. Die Nachrichten innerhalb des Systems werden veröffentlicht (engl. „publish“) und können von jedem Node in Anspruch genommen werden (engl. „subscribe“). Falls die Daten nicht permanent zur Verfügung stehen müssen, können die Nodes ihre Daten auch auf Nachfrage liefern. Dafür werden diese als Services implementiert. ROS ist kein Echtzeitsystem, kann aber mit Echtzeitsystemen zusammenarbeiten.

Des Weiteren beinhaltet ROS einen, mit einer Physik-Engine ausgestatteten Simulator, der im nächsten Abschnitt vorgestellt wird.

6.2. Simulation mit Nachbildung physikalischer Eigenschaften

Das Robot Operating System (ROS) beinhaltet von vornherein eine Simulation, basierend auf dem Gazebo-Projekt[5]. Dabei werden die physikalischen Eigenschaften unter Zuhilfenahme der ODE-Bibliothek[6] realisiert. Eine Schnittstelle zur Bullet[7] ist auch vorhanden.

Die Arbeit von Harris und Conrad [HC11] gibt einen sehr guten Überblick über die existierenden Frameworks und Simulatoren. Die Gazebo-Simulation in Verbindung mit ROS schneidet dabei sehr gut ab. Die Autoren bestätigen die Möglichkeit der Verwendung des Simulators als Evaluationswerkzeug, da das beobachtete Verhalten sowie die physikalischen Eigenschaften das Original sehr gut nachbilden.

Damit stellt Gazebo eine physikalische und dynamische Open-Source-Software dar, die die physikalische Simulation mehrerer Roboter sowie die Anbindung an mehrere Bibliotheken bereitstellt. Dabei wird nicht nur die Umgebung physikalisch simuliert, sondern auch mehrere Sensoren bzw. Sensorarten und eine oder mehrere heterogene Roboterplattformen. Harris und Conrad loben, dass die Simulation präzise Ergebnisse liefert und dadurch, in Verbindung mit der neuesten Hardware, sogar schnellere Ausführungszeiten als unter realen Bedingungen ermöglicht. Abbildung 6.2 visualisiert einen PR2-Roboter in der simulierten Restaurantumgebung (RACE-Szenario).

Aus dem Bild ist ersichtlich, dass die Simulation sich nach Belieben gestalten lässt und somit eine sehr gute Annäherung an die echte Umgebung ermöglicht. Sind die Objekte modelliert, können sie während der Laufzeit dynamisch platziert werden, ihre Position und Orientierung kann geändert werden oder sie können durch einen Roboter ergriffen werden.

Für die Objekterkennung werden dieselben Modelle aus der Datenbank verwendet, sowohl für die Integration in Gazebo als auch für den notwendigen Vergleich. Das

[5] http://www.gazebosim.org/
[6] http://wiki.ros.org/physics_ode/ODE
[7] http://wiki.ros.org/bullet

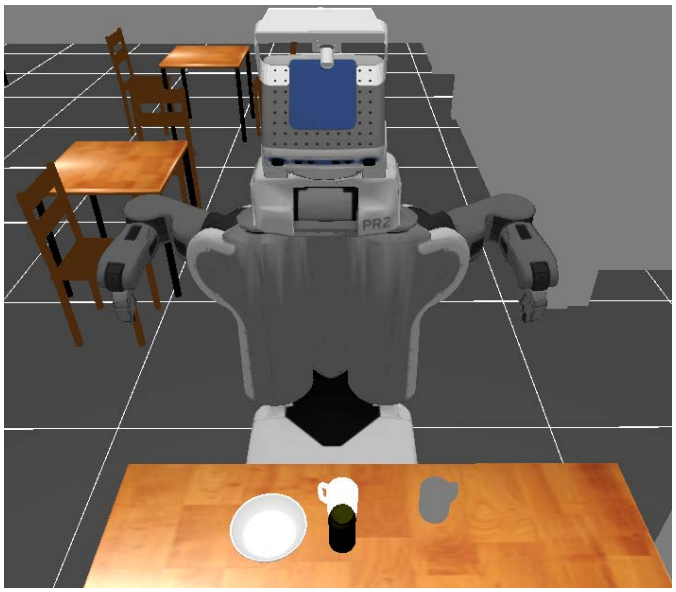


Abbildung 6.2.: In Gazebo simuliertes Restaurantszenario mit mehreren Objekten und einem PR2-Roboter.

ermöglicht die Realisierung und mehrmalige Wiederholung beliebiger Szenen und Szenarien. Natürlich muss unbedingt berücksichtigt werden, dass die Sensorinformationen, anders als in der Realität, innerhalb der Simulation rauschfrei sind.

Eine der tragenden Säulen der vorliegenden Dissertation ist die Verwendung mehrerer, auf unterschiedlichen Objekteigenschaften operierender Detektoren. Eigenschaften wie Form und Größe sind durch die Modelle vorgegeben und werden in der Simulation ausreichend detailliert präsentiert. Für die Verwendung weiterer Eigenschaften, wie Farbe und Textur, wurden einige vorhandene Datenbankmodelle durch diese Eigenschaften erweitert. Dafür wurden verschiedene Programme wie Blender[8] und Meshlab[9] verwendet. Abbildung 6.3 stellt einen Vergleich zwischen einer realen und einer simulierten Szene dar. Dabei sind zwei in der Realität existierende und zur Verfügung stehende Objekte, hier zwei Bücher samt Textur und Farbe, simuliert und mithilfe eines SIFT-/SURF-Detektors detektiert und erkannt worden. Deutlich zu sehen ist einerseits, dass die realen Modelle auch in der Simulation verwendet werden können und andererseits, dass die simulierte Szene kaum Rauschen aufweist.

Dabei wurde durch die mehrmalige Wiederholung der Experimente mit unterschiedlichen Objekten festgestellt, dass sich die Anzahl der durch den Detektor erkannten Merkmale in der Realität und Simulation nur geringfügig voneinander unterscheidet. Da

[8] http://www.blender.org/
[9] http://meshlab.sourceforge.net/

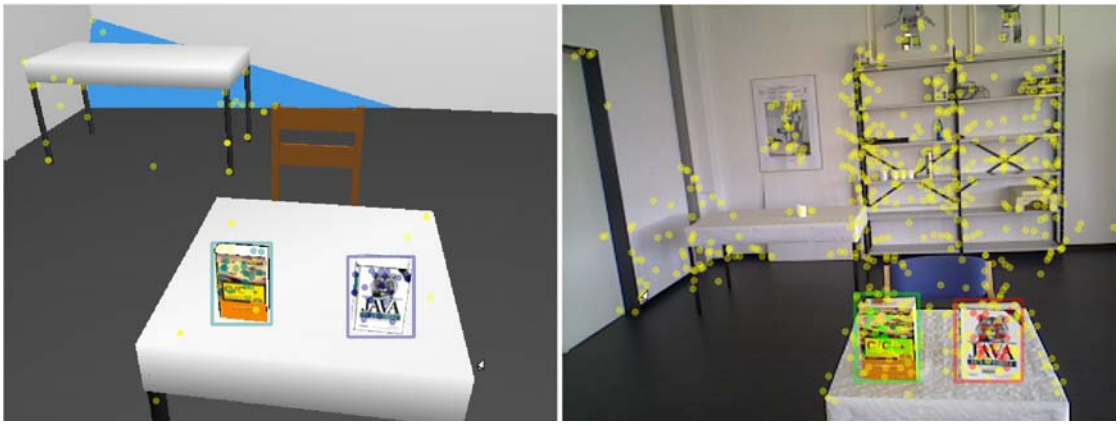


Abbildung 6.3.: Vergleich einer realen und einer simulierten Szene, die die Verwendung eines SIFT-/SURF-Detektors zeigen. Zum besseren Verständnis werden die beiden Bilder horizontal gespiegelt präsentiert.

die Anzahl dieser Merkmale auch während laufender realer Experimente variiert, kann dieser Umstand durchaus vernachlässigt werden. Damit werden die Ergebnisse von Harris und Conrad [HC11] durchaus bestätigt. Die Simulation stellt somit einen gleichwertigen Ersatz für die reale Umgebung dar und kann für die Evaluation eingesetzt werden. Des Weiteren erlaubt die Simulation die Benutzung aller im Rahmen dieser Dissertation notwendigen Ressourcen und bietet durchaus eine geeignete Evaluationsumgebung. Mehrere signifikante Experimente werden danach ausgewählt und unter realen Rahmenbedingungen wiederholt.

6.3. Umgebung und Roboterkonfiguration

Alle Experimente sind auf die Innenräume beschränkt. Wie bereits erwähnt, stammen die meisten Szenen aus der Arbeit im Rahmen des EU-Projektes RACE. Der Roboter erkennt seine Umgebung und fährt an den Tisch (vgl. Anhang D), danach werden die Sensoren ausgerichtet (Roboterkopf), und die Szene wird aufgenommen und segmentiert. Auf die gefundenen Cluster und weitere Merkmale werden die Detektoren angewandt und die Ergebnisse mit der Datenbank verglichen. Durch die vorgegebene Umgebung und festgelegte Roboterkonfiguration kann von einem konstanten Abstand der Sensoren zum Tisch während der Experimente ausgegangen werden.

Zur Evaluation der verwendeten Detektoren werden in den meisten Fällen Modelle der bestehenden Datenbank des RACE-Projekts verwendet, unter anderem: eine Tasse, ein Teller, ein Löffel, ein Messer und eine Gabel; ein Ölspender und eine Vase sowie ein

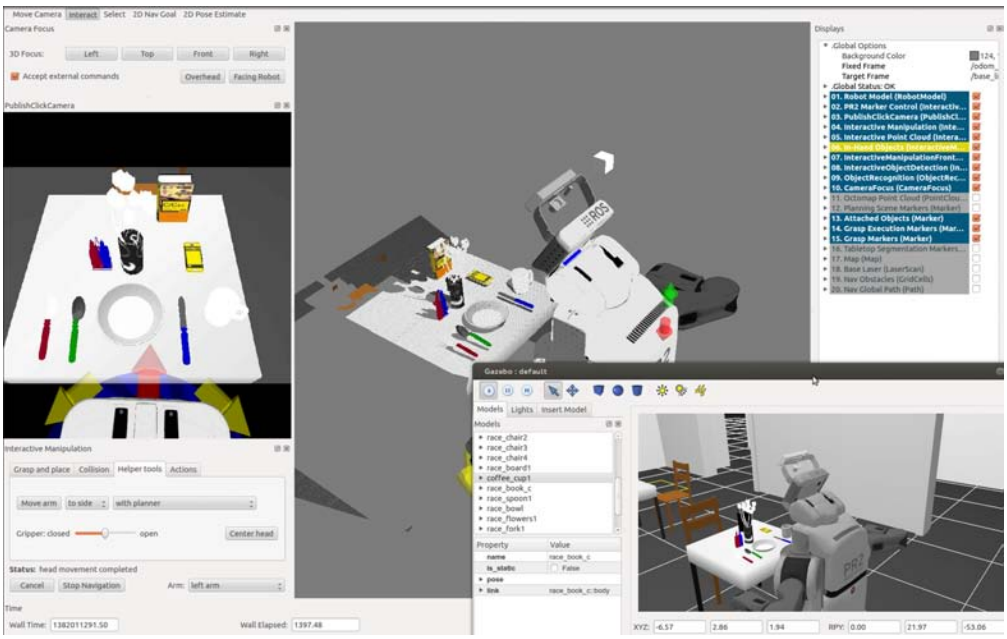


Abbildung 6.4.: Ansicht der zur Evaluation verwendeten Objekte in der Simulation, links das Kamerabild, in der Mitte die Punktwolke und rechts die Simulation.

Mobiltelefon und ein Buch. Dabei sind die meisten Objekte, bis auf Teller und Tasse, so präpariert, dass sie neben der Form weitere unterschiedliche Eigenschaften aufweisen. Abbildung 6.4 visualisiert die verwendeten Objekte als Bild und Punktwolke aus der Sicht des Roboters. Die Anzahl der Objekte kann weiterhin variieren, einerseits indem Objekte hinzukommen oder wieder herausgenommen werden. Im Weiteren steht die ROS-eigene Datenbank mit mehr als 300 Objekten zur Verfügung. Leider gibt es keine ausführliche Dokumentation für die Household Object Database, somit ist es teilweise notwendig, „reverse“-Engineering zu betreiben, um die Art, Form und Größe der verwendeten Modelle nachvollziehen zu können. Außerdem werden in dieser Arbeit auch andere Objekteigenschaften benötigt, die dann teilweise nur mühsam integriert werden können.

Wie aus der Abbildung 6.4 ersichtlich ist, weisen die Vase, das Buch und das Mobiltelefon texturelle Merkmale auf. Zusätzlich besitzen die beiden letzteren Farbinformationen. Auch die Griffe des Bestecks und des Ölspenders sind zur besseren Erkennung unterschiedlich eingefärbt.

6.4. Szenenanalyse

Nachdem der Roboter die Szene aufgenommen hat, erfolgt die Segmentierung. Diese basiert auf der euklidischen Distanz: Überschreitet der Abstand zwischen den einzelnen Punktwolken mehr als einen bestimmten Schwellenwert, zum Beispiel zwei Zentimeter, werden die Cluster voneinander separiert. Darüber hinaus wird zusätzlich für jedes Cluster die Farb- und Texturinformation aus dem RGB-Bild bestimmt, transformiert und in der Punktwolke gespeichert. Die Multi-Sensor-Fusion erlaubt die Anwendung aller hier vorgestellten Detektoren auf eine sogenannte gefärbte 3-D-Punktwolke, was die Rückführung der 2-D-Detektoren in 3-D möglich macht. Die verwendete Struktur dieser Punktwolke ist ausführlich in Kapitel 3 vorgestellt. Die detaillierte Vorgehensweise bei der Segmentierung der Punktwolke sowie die Wiedergewinnung der Farbinformation kann in Kapitel 4 nachgeschlagen werden.

6.4.1. Objekterkennung

Wie bereits in Kapitel 4 ausführlich beschrieben wurde, hat jeder Detektor spezifische Vor- und Nachteile. Aus der Sicht des Autors ist eine robuste, effiziente sowie akkurate Objekterkennung, die nur auf einem einzelnen Algorithmus oder einer Methode basiert, kaum realisierbar. Daher wird hier ein Weg eingeschlagen, der viele bis alle zur Verfügung stehenden Daten kombiniert, fusioniert und auswertet. Dafür wird eine Erkennungspipeline gebildet, die auf der Verwendung unterschiedlicher Detektoren basiert. Dabei wird darauf geachtet, dass die verwendeten Detektoren mit den verschiedenen Daten unterschiedlicher Objekteigenschaften arbeiten. So werden hier Informationen basierend auf den Volumen-, Größen-, Farbe- und Textureigenschaften von Objekten verwendet.

	Größe	Farbe	Textur	IDF	ICP^2	Komb.
Objekte (Orientierung fest)	53.2	63.2	23.5	20.7	89.2	98.7
Objekte (Orientierung variierend)	51.1	54.4	22.2	16.6	82.4	96.6

Tabelle 6.1.: Ergebnisse einzelner Detektoren sowie das kombinierte Resultat.

Die Tabelle 6.1 präsentiert die Ergebnisse einzelner Detektoren sowie das Resultat ihrer Kombination. Dabei soll explizit erwähnt werden, dass das Ergebnis einzelner Detektoren nur eine bedingte Aussagekraft über die Güte einzelner Algorithmen besitzt. Es ist absolut verständlich, dass eine Methode, bei der bestimmte Merkmale fehlen, erfolglos sein wird. Die Idee hinter dieser Tabelle ist vielmehr, zu verdeutlichen, dass bei der Verwendung vieler nicht zuvor präparierter Objekte, die Objekterkennung nur durch die Kombination mehrerer Detektoren zum Erfolg führt.

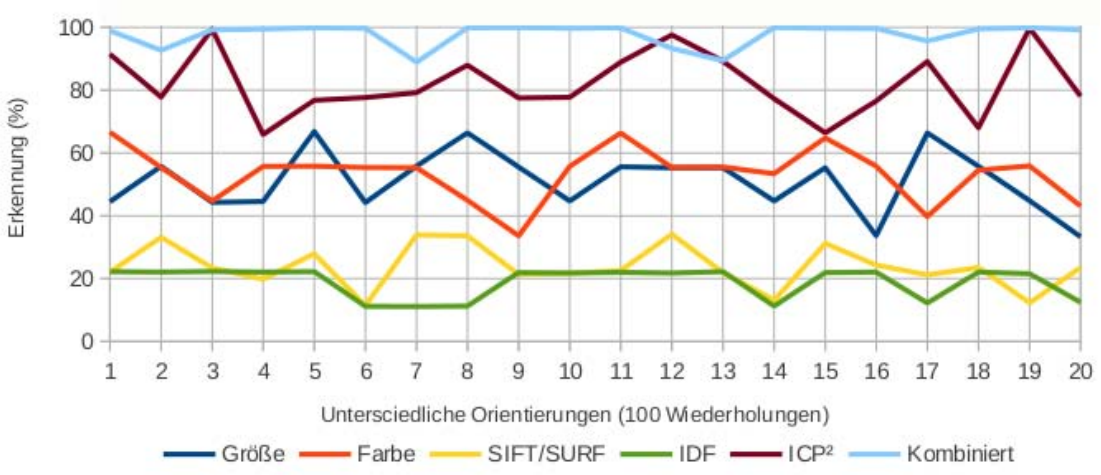


Abbildung 6.5.: Performanz einzelner Detektoren sowie die kombinierten Ergebnisse in der Simulation ohne Verdeckung - mit 20 verschiedenen Objektanordnungen ohne Beibehaltung der Orientierung, jeweils 100 mal wiederholt.

Außerdem veranschaulicht die oben präsentierte Tabelle, dass die Kombination verschiedener Detektoren das Ergebnis der Objekterkennung deutlich steigern kann. Erkennen die einzelnen Detektoren die Merkmale der jeweiligen Objekte robust und sicher, kann von einer stabilen und hochperformanten Objekterkennung ausgegangen werden. Der Verlust der Erkennungsrate von $1,1\,\%$ bei der variierenden Orientierung ist damit zu begründen, dass ein oder mehrere Objekte aus den entstehenden Perspektiven eine deutlich geringere bis nicht ausreichende Anzahl an Merkmalen für den jeweiligen Detektor zur Verfügung stellen.

Außerdem veranschaulicht die oben präsentierte Tabelle, dass die Kombination verschiedener Detektoren das Ergebnis der Objekterkennung deutlich steigern kann. Erkennen die einzelnen Detektoren die Merkmale der jeweiligen Objekte robust und sicher, kann von einer stabilen und hochperformanten Objekterkennung ausgegangen werden. Aus den entstehenden Perspektiven weist ein oder weisen mehrere Objekte eine deutlich geringere bis nicht ausreichende Anzahl an Merkmalen für den jeweiligen Detektor auf, womit der Verlust der Erkennungsrate von $1,1\,\%$ zu begründen ist.

Listing 6.1: Konsolenausgabe für die reale Szene aus Abbildung 6.6

```
Number of clusters: 6 models: 6
Model 0 (database id 5) has 262 triangles and 432 vertices
Model 1 (database id 9) has 1290 triangles and 3738 vertices
Model 2 (database id 2) has 712 triangles and 358 vertices
```

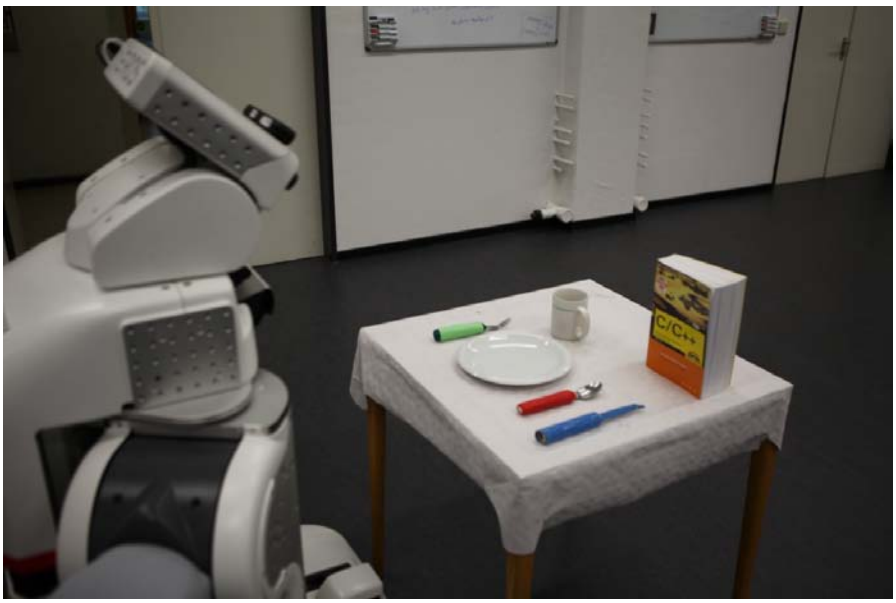


Abbildung 6.6.: Ansicht einiger zur Evaluation verwendeter Objekte in der Realität sowie die Position des Roboters in Relation zur Szene.

```
Model 3 (database id 7) has 16480 triangles and 45749 vertices
Model 4 (database id 6) has 780 triangles and 2314 vertices
Model 5 (database id 8) has 726 triangles and 2164 vertices
Detected 6 graspable object(s):
Adding mesh. #triangles=786
(0): book (tags: book) in frame base_link
Adding mesh. #triangles=3870
Adding mesh. #triangles=2136
Adding mesh. #triangles=49440
(1): bowl (tags: bowl) in frame base_link
(2): race_coffee_mug (tags: race_coffee_mug) in frame base_link
Adding mesh. #triangles=2340
(3): knife (tags: knife) in frame base_link
(4): fork (tags: fork) in frame base_link
Adding mesh. #triangles=2178
(5): spoon (tags: spoon) in frame base_link
Action finished: 6 of 6 objects recognized
```

Ein weiterer Evaluationsschritt ist in der Abbildung 6.5 zu sehen. Dabei werden 20 unterschiedliche Anordnungen der oben beschriebenen Objekte jeweils 100-mal nachein-

ander in der Simulation erkannt. Auch hier erweist sich die Objekterkennungspipeline als robust und hochperformant.

Zwar kommt die Simulation der Realität sehr nahe, dennoch soll hier auch gezeigt werden, dass die entwickelte Objekterkennung auch unter realen Bedingungen vergleichbare Ergebnisse liefert. Die Simulation bietet optimale Bedingungen, konstante Lichtverhältnisse und ideale Sensorwerte. Daher ist davon auszugehen, dass die Performanz unter realen Bedingungen schlechter ausfällt.

Prediction / Truth	Mug	Bowl	Spoon	Knife	Fork	Oil dispenser	Vase	Mobile phone	Book	Salt shaker	Pepper shaker
Mug	94	2	0	0	0	2	0	0	0	3	4
Bowl	0	84	0	0	0	0	0	2	1	0	0
Spoon	0	0	88	4	4	0	0	0	0	0	0
Knife	0	0	3	91	2	0	0	0	0	0	0
Fork	0	0	5	2	89	0	0	0	0	0	0
Oil dispenser	0	0	0	0	0	86	0	3	0	1	0
Vase	0	0	0	0	0	0	97	0	0	0	1
Mobile fon	0	1	0	0	0	4	0	90	0	0	0
Book	0	1	0	0	0	0	0	0	98	0	0
Salt shaker	1	0	0	0	0	0	0	0	0	85	0
Pepper shaker	3	0	0	0	0	0	2	0	0	3	91

Abbildung 6.7.: Auflistung der Erkennungsraten der konstruierten Pipeline für einzelne Objekte. 100-malige Wiederholung mit unterschiedlicher Position des zu erkennenden Objekts ohne Beachtung der Orientierung.

Die Abbildung 6.7 stellt die Ergebnisse der Objekterkennung für die einzelnen Objekte in Form einer Tabelle dar. Dafür wurden die Objekte einzeln, ohne Beachtung der Orientierung, 100-mal sichtbar auf dem Tisch platziert und durch die Plattform detektiert und erkannt. Dabei kann nicht nur die Erfolgsrate, sondern auch die Anzahl falsch erkannter Objekte abgelesen werden. Die präsentierten Erkennungsraten für reale Objekte sind erwartungsgemäß niedriger als für die simulierten Objekte, kommen aber den idealen Werten sehr nahe und unterstützen die bereits für die simulierten Ergebnisse präsentierten Schlussfolgerungen.

Sind die Ergebnisse für die einzelnen Objekte evaluiert, wird aus diesen ein komplexes Szenario (ohne Verdeckung) konstruiert und einer Analyse durch das Robotersystem unterzogen. Abbildung 6.6 visualisiert einige reale Objekte, die für die Evaluation verwendet wurden. Untersucht wurden 20 unterschiedliche Objektanordnungen, wobei die Objekte denen in der Simulation ähnelten, die Anzahl der Objekte variierte abhängig von ihrer Dimension sowie von der Größe des durch die Sensoren erfassten Bereichs. Dabei erreichte die Erkennungsrate 93,3 %. Das Diagramm 6.8 visualisiert die Ergebnisse für einzelne Szenen. Auch hier liegen die Ergebnisse nur geringfügig hinter den Ergebnissen der Simulation zurück.

Ein weiterer interessanter Aspekt ist, dass die Schwankungen in der Erkennungsrate einzelner Detektoren nur einen geringfügigen Einfluss auf das kombinierte Ergebnis

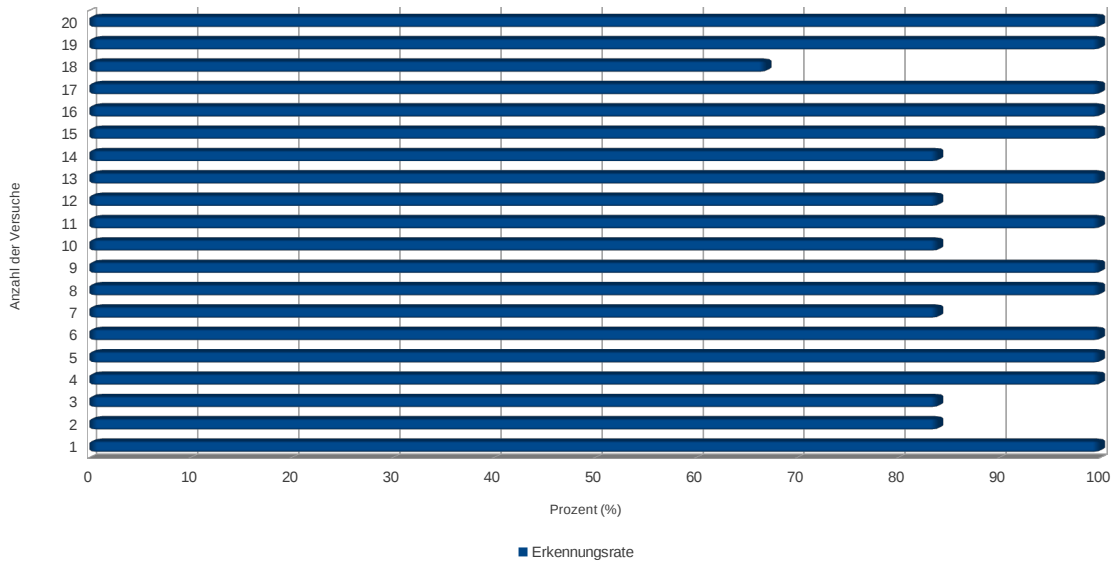


Abbildung 6.8.: Performanz der Objekterkennungspipeline unter realen Bedingungen.

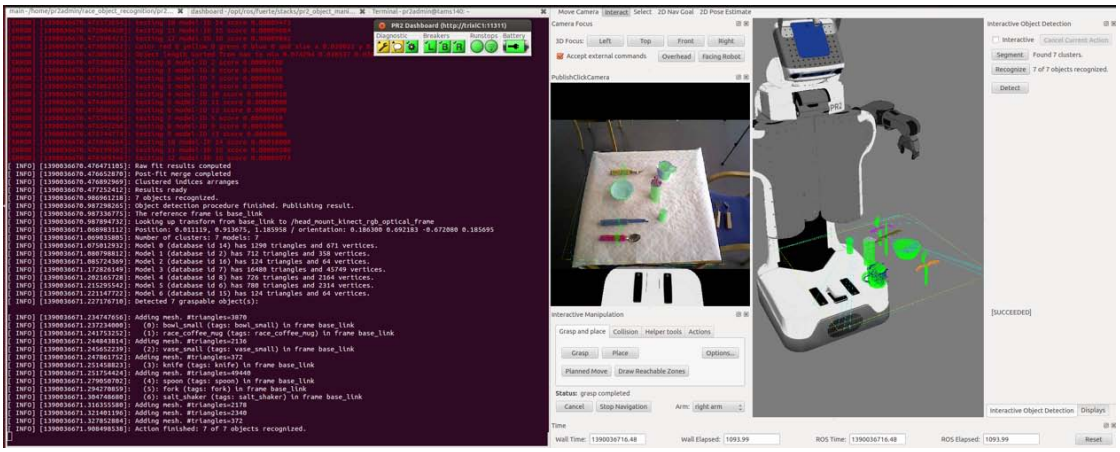


Abbildung 6.9.: Analyse einer realen Szene (von rechts nach links): Information über die Anzahl gefundener Cluster und erkannter Objekte. Daneben die Visualisierung der Umgebung, der Plattform und der Analyse, sowie ein Image von einer Kamera der Plattform. Links: die Textausgabe der Konsole über die Ergebnisse der Analyse.

ausüben. Daher kann davon ausgegangen werden, dass die Erkennung umso stabiler und sicherer ist, je mehr Detektoren an dem Erkennungsprozess beteiligt und mehr Objekteigenschaften involviert sind. Auf der anderen Seite sollte beachtet werden, dass nicht alle Algorithmen die gleiche Zeitperformanz aufweisen und parallel angewendet werden können. Auch die Ressourcenanforderungen sind sehr unterschiedlich.

Nach der Erkennung der Objekte stehen zwei Möglichkeiten zur Verfügung, um die Ergebnisse zu präsentieren. Auf der einen Seite visuell über Markierungen, die in RVIZ, dem Visualisierungswerkzeug von ROS, dargestellt werden (vgl. Abbildung 6.9). Diese Visualisierung dient mehr dem intuitiven Verständnis, die Information kann aber nicht weiter für die Evaluation verwendet werden. Auf der anderen Seite können die Ergebnisse in Form einer Konsolenausgabe präsentiert werden, wie etwa in der Auflistung 6.1 für die in der Abbildung 6.6 vorgestellte Szene.

Die Abbildung 6.9 stellt die Visualisierung für die Analyse einer realen Szene dar. Die Daten mehrerer Sensoren, der Objektdetektion und der gefundenen Cluster sowie die Resultate in Form der Konsolenausgabe und der Bildmarkierungen sind sichtbar. Unter der Bezeichnung Objekt-ID ist die Nummer angegeben, unter welcher die erkannten Objekte in der Datenbank abgelegt sind.

Eine weitere Neuerung gegenüber der Standard-ROS-Objekterkennung ist die Möglichkeit, dass ein oder mehrere Cluster als nicht bekannt erkannt werden. In vielen Verfahren, wie zum Beispiel bei den Standardalgorithmen in ROS, werden die Wahrscheinlichkeiten, egal wie klein sie auch sein mögen, für jedes mögliche Objekt ausgegeben und damit als erkannt markiert. Dies behindert aber die Entwicklung der Objekterkennung auf lange Sicht, da alle Objekte erkannt werden und damit auf die unbekannten Objekte nicht reagiert werden kann. Zwar existiert die Möglichkeit, einen Schwellenwert bei der Berechnung der Wahrscheinlichkeiten einzubauen, dennoch unterliegen die Wahrscheinlichkeiten einer starken Schwankung, was eine dynamische Anpassung des Schwellenwerts notwendig macht. Durch die Verwendung unterschiedlicher Detektoren ist es relativ einfach möglich, eine Aussage über ein unbekanntes Objekt zu treffen und eventuelle weitere Schritte einzuleiten.

Doch zunächst soll im folgenden Kapitel der Einfluss einer Änderung der Torsoposition auf die Objekterkennung und somit auf die gesamte Szenenanalyse untersucht werden. Dies ist notwendig, damit der spätere Ablauf der Erkennungspipeline nachvollzogen werden kann. Außerdem wurden beim Entwurf und bei der Realisierung des dieser Arbeit zugrunde liegenden Systems diese Erkenntnisse berücksichtigt und beeinflussen somit das resultierende System.

6.4.2. Einfluss der Torsoposition auf die Szenenanalyse

Im späteren Verlauf dieses Kapitels werden auch das Greifen und die Manipulation in die Erkennungspipeline eingebunden.

Daher spielen die Roboterkonfiguration und der Einfluss der Torsoposition und der

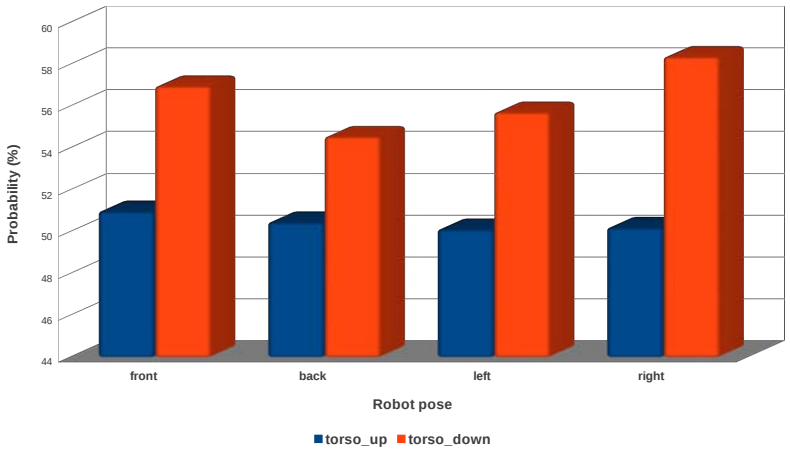


Abbildung 6.10.: Einfluss der Positionsänderung auf die Objekterkennung.

Perspektive auf das Erkennen und Greifen in dieser Arbeit eine indirekte, aber dennoch wichtige Rolle. Durch die Experimente konnte nachgewiesen werden, dass durch die Bewegung des Torsos nach oben die durchschnittliche Qualität der berechneten Griffe um ca. 32,5 % steigt.

Dabei wurden der Torso und die Plattform nacheinander bewegt und die Objekte auf dem Tisch nach Beendigung der Bewegung detektiert und erkannt. Die meisten Experimente wurden in der Simulation ausgeführt, danach wurden mehrere signifikante Beispiele in die Realität übertragen. Die Abbildung 6.11 visualisiert die durchgeführten Bewegungen. Die oberen beiden Bilder visualisieren die Bewegungen nach links und rechts, wobei der Torso in allen Positionen hoch und runter gefahren wurde. Die unteren Bilder veranschaulichen die Änderung der Torsoposition bei dem geraden Blick des Roboters auf die Szene. Die Abbildung 6.10 zeigt den Einfluss der Änderung der Position des Roboters und des Torsos auf die Objekterkennung.

Durch die Experimente wird sichtbar, dass die Qualität der Griffe am höchsten ist, wenn der Torso komplett hochgefahren ist. Im Gegensatz dazu steigt die Performanz der Objekterkennung mit der nach unten gefahrenen Torsoposition deutlich um ca. 10 %. Somit konnte folgende Strategie daraus abgeleitet und realisiert werden: Die Objektdetektion erfolgt mit einem heruntergefahrenen Torso. Danach wird der Torso hochgefahren, und die Trajektorie, angepasst an die bereits berechneten Griffe, wird kalkuliert und ausgeführt. Des Weiteren wird deutlich, dass kleine Änderungen oder Ungenauigkeiten der Perspektive die Objekterkennung sowie das Greifen kaum beeinflussen und daher nicht weiter betrachtet werden müssen. Die hier präsentierten experimentellen Ergebnisse sind zum Teil in der Publikation [KRZ12] veröffentlicht.

6.4.3. Bestimmung der Orientierung

Wie bereits in Kapitel 4 beschrieben, ist die Erkennung der Objekte grundlegend, reicht aber allein nicht aus.

Für die weiteren Schritte, wie zum Beispiel das Greifen oder Platzieren von erkannten Objekten, ist zusätzlich deren Position und Orientierung notwendig. Die Segmentierung liefert die Position eines Objekts, nicht aber dessen Orientierung. Einige der Detektoren berechnen die Orientierung der Objekte in der Szene während der Erkennung. Andere Detektoren, wie zum Beispiel die SIFT-/SURF- und Farbdetektoren, benötigen keine Orientierung und berechnen sie deshalb auch nicht. In dieser Arbeit wird in solchen Fällen der bereits vorhandene ICP-Algorithmus genutzt. Diesem wird die gefundene

Abbildung 6.11.: Einfluss der Roboterposition und Konfiguration auf die Objekterkennung.

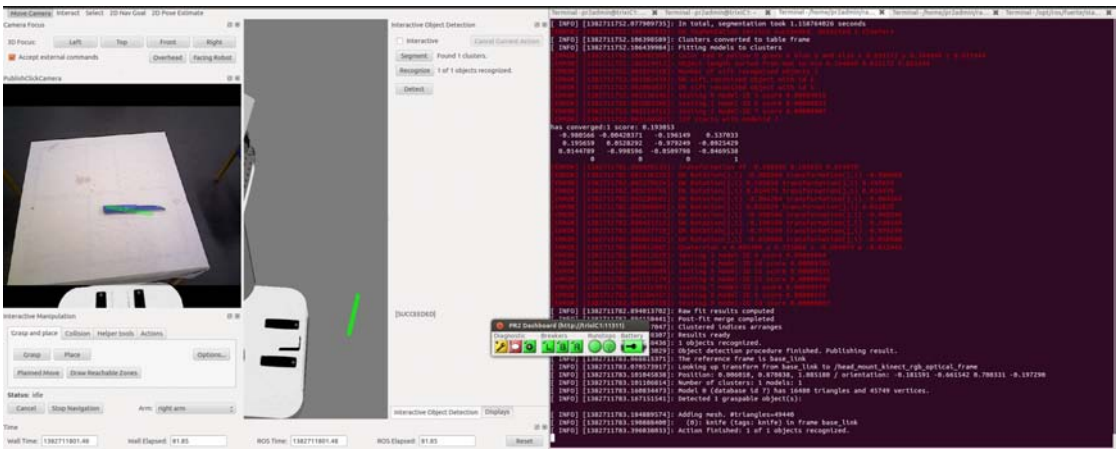


Abbildung 6.12.: Erkanntes reales Objekt mit bestimmter Orientierung. Wie deutlich zu sehen ist, beinhaltet die Orientierung einen leichten Fehler.

Punktwolke sowie das dazugehörige Modell übergeben. Zurückgeliefert wird die homogene Transformation, die für die weitere Anwendung später in Quaternion konvertiert wird.

Die Abbildungen 6.12 und 6.13 visualisieren die berechnete Orientierung mithilfe der Visualisierungsmarker unter realen und simulierten Bedingungen. Beide berechneten Orientierungen weisen Fehler auf. Erstaunlicherweise sind in der Simulation bestimmte Werte fehlerhafter als die realen. Einer der Gründe dafür ist die Rechnerperformanz, die auf dem Roboter deutlich höher ist als bei dem Rechner, auf dem die Simulationsversuche durchgeführt wurden. Dennoch reicht die berechnete Orientierung, um das erkannte Objekt sicher zu greifen. Jedoch benötigt die akkurate Berechnung extrem viel Zeit, deshalb wird hier ein Kompromiss aus annehmbarer Genauigkeit und Zeitperformanz eingegangen.

6.4.4. Verdeckung

Wie bereits erwähnt, ist das Problem der Verdeckung seit mehreren Jahrzehnten ein ungelöstes Problem der Objekterkennung. Die Definition der Verdeckung ist aus dem Bereich der 3-D-Bildverarbeitung übernommen und bereits in Kapitel 1.2 vorgestellt worden. Danach ist die Verdeckung ein Effekt, bei dem ein oder mehrere Objekte die Sicht auf ein oder mehrere andere Objekte komplett (total) oder teilweise (partiell) blockieren. Dadurch wird die Objekterkennung mit Standardverfahren erschwert oder ist gar unmöglich.

Das hier vorgestellte Verfahren bietet durch die Verwendung einer mobilen Plattform

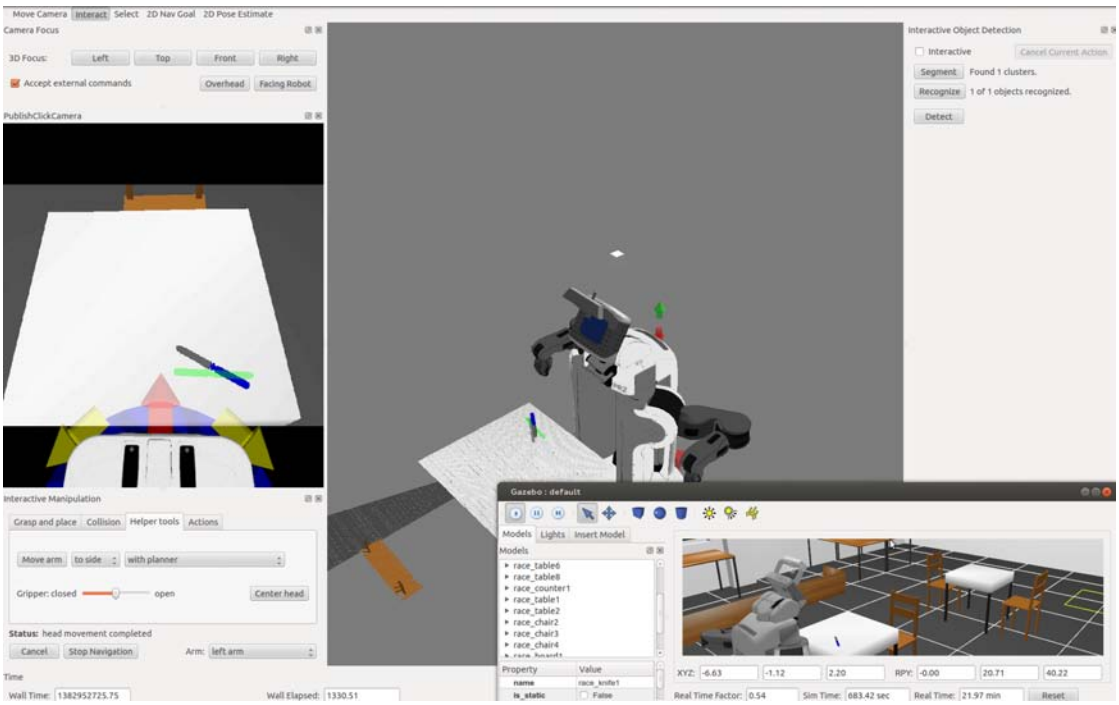


Abbildung 6.13.: Erkanntes simuliertes Objekt mit bestimmter Orientierung. Wie deutlich zu sehen ist, ist die Orientierung stark fehlerhaft.

und des aktiven Wahrnehmens einige Lösungsansätze, um die durch die Verdeckung entstehenden Probleme zu minimieren, wenn nicht sogar komplett zu lösen. Dabei ist extrem wichtig, dass, wie bereits erwähnt, ein Objekt auch nicht erkannt werden kann (als „nicht erkannt“klassifiziert wird). Darauf basierend wird versucht, eine Verdeckung zu erkennen. Eine Verdeckung wird vermutet, wenn ein oder mehrere Objekte nicht erkannt werden. Eine Verdeckung wird auch dann vermutet, wenn ein Objekt erkannt wird, die volumetrischen Angaben aber nicht mit den Angaben in der Datenbank übereinstimmen. Werden ein oder mehrere Objekte „nicht erkannt“, kann ein weiterer Schritt eingeleitet werden. So ist zum Beispiel die Änderung der Perspektive eine von zwei zur Verfügung stehenden Möglichkeiten. Abbildung 6.14 visualisiert dieses Vorgehen.

Eine weitere Möglichkeit stellt die Abbildung 6.15 grafisch dar. Falls die Änderung des Blickwinkels keinen Erfolg gebracht hat, wird, basierend auf den Daten der Detektoren, aktiv in die Szene eingegriffen. Die Flasche wird ergriffen und der Roboterarm zur Seite gefahren. Durch die aufgelöste Verdeckung werden alle Objekte richtig erkannt.

Bei dem in Abbildung 6.15 visualisierten Prozess liefert der SIFT-/SURF-Detektor die nötige Information. Da dieser Detektor sehr zuverlässig ist, wird das Objekt als

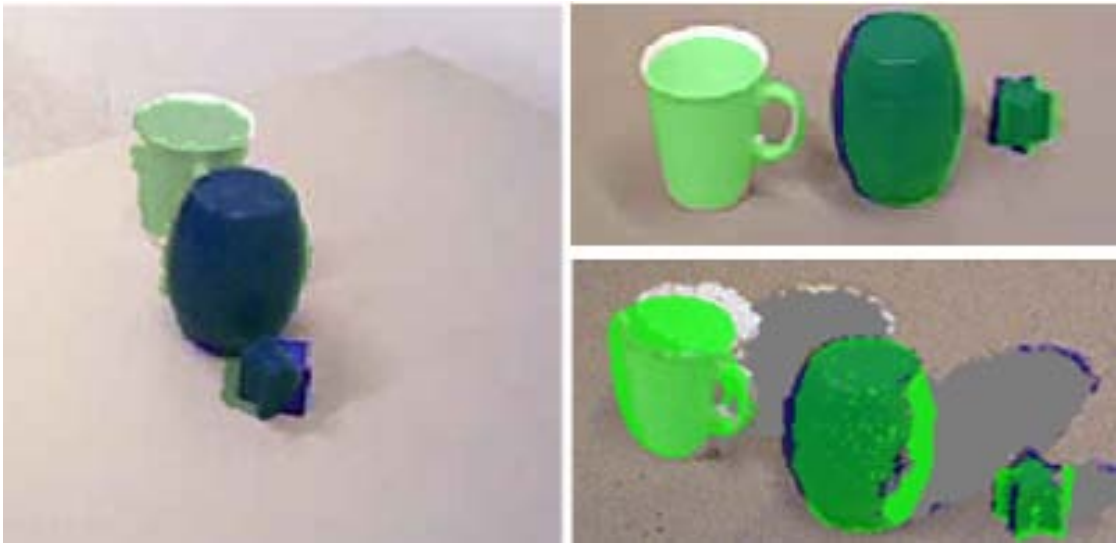


Abbildung 6.14.: Auflösung der Verdeckung durch eine Änderung der Perspektive. Nach der erneuten Analyse der Szene können alle Objekte richtig detektiert und erkannt werden.

erkannt markiert. Da weitere Eigenschaften, wie Farbe, Größe und Textur, nicht mit dem gefundenen Cluster übereinstimmen, wird aktiv in die Szene eingegriffen. Dabei wird das erkannte Objekt ergriffen und außerhalb der Szene platziert. Nach der Entfernung des Objekts wird die Szene noch einmal analysiert. In dem vorliegenden Beispiel werden dabei alle Objekte richtig erkannt.

Die Abbildungen 6.16 bis 6.19 visualisieren die Auflösung der Verdeckung innerhalb einer in der Simulation nachgestellten Szene. Die Szene ist in der ersten Abbildung präsentiert. Der Roboter steht vor einem Tisch aus dem RACE-Restaurantszenario. Auf dem Tisch steht eine Vase, die einen Ölspender verdeckt, außerdem liegt dort ein Buch, das einen Kaffeebecher verdeckt.

Abbildung 6.17 stellt dieselbe Szene mit einer veränderten, seitlichen Perspektive dar. Alle vier Objekte werden richtig segmentiert und erkannt. Trotz der Tatsache, dass die Analyse erfolgreich abgeschlossen werden konnte, wird in weitere Szenen aktiv eingegriffen. Dies soll einerseits die Redundanz aufzeichnen, andererseits soll das präsentierte Konzept verdeutlicht werden. Durch eine Reihe von Möglichkeiten kann eine Szene nach und nach analysiert und damit können alle beteiligten Objekte erkannt werden.

Basierend auf der gleichen Ausgangsszene wird in Abbildung 6.18 aktiv eingegriffen: Der Roboter greift die Vase und stellt sie seitlich auf dem Tisch wieder ab. Die Verdeckung des Ölspenders wird dadurch aufgelöst, und drei Objekte werden segmentiert und

Abbildung 6.15.: Auflösung der Verdeckung durch einen aktiven Eingriff in die Szene. Nach der Manipulation wird die Szene erneut analysiert. Nach der erneuten Analyse sind alle Objekte richtig erkannt worden.

erkannt. Aufgrund der Verdeckung des Bechers durch das Buch kann dieser nicht richtig segmentiert und erkannt werden.

Die Abbildung 6.19 zeigt mögliche Schritte nach der Verschiebung der Vase. Das Buch ist durch die SIFT/SURF-Features richtig erkannt worden, die Größe stimmt aber mit dem Datenbankmodell nicht überein. Daher greift die Roboterplattform das Buch und positioniert den Arm mit dem Buch außerhalb der Szene. Nach der erneuten Analyse können alle Objekte richtig erkannt werden.

Die Übertragung der in der Simulation vorgestellten Szenarien in die reale Umgebung gestaltet sich nicht immer reibungslos und wird nachfolgend beschrieben und visualisiert.

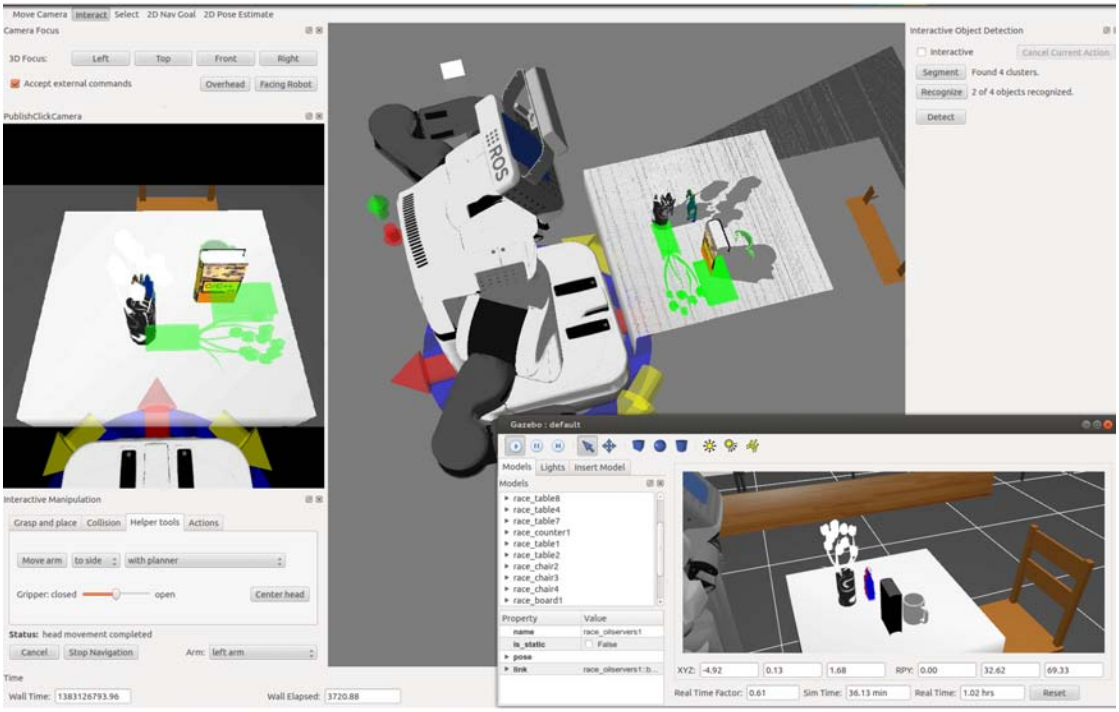


Abbildung 6.16.: Eine in der Simulation nachgestellte Tischszene mit einem Buch, das einen Kaffeebecher verdeckt, und einem durch eine Vase verdeckten Ölspender.

Die ersten beiden Abbildungen visualisieren ein einfaches Szenario, wobei eine Tasse einen Löffel verdeckt (vgl. Abbildungen 6.20 und 6.21). Die Bewegung des Roboters wird vernachlässigt, obwohl die Auflösung der Verdeckung durch eine Änderung der Perspektive durchaus zum Erfolg geführt hätte. Alle nachfolgenden Bilder sind folgendermaßen aufgebaut: Ganz rechts finden sich die Ergebnisse der Detektion und Erkennung; rechts ist die Visualisierung in der Simulation platziert; links ist die Sicht einer der Roboterkameras auf die Szene und ganz links die Ergebnisausgabe in der Konsole dargestellt. Die rote Schrift wird benutzt, um die Ergebnisse der Szenenanalyse hervorzuheben, und soll keinesfalls mit der Fehlerausgabe verwechselt werden.

Da der Löffel einen ausreichenden Abstand zur Tasse aufweist, kann die Tasse mithilfe des ICP² sicher erkannt werden. Ein Teil des Löffels ist sichtbar, liefert aber nicht genug Informationen, um ein Erkennen zu ermöglichen. Nach dem die Tasse erkannt wurde, kann das erkannte Objekt gegriffen werden. Im vorliegenden Fall sind die vorhandenen Griffe aus der Datenbank geladen und getestet worden. Wird ein passender Griff gefunden, kann dieser ausgeführt werden. Bei einer erneuten Analyse der Szene kann ein Löffel

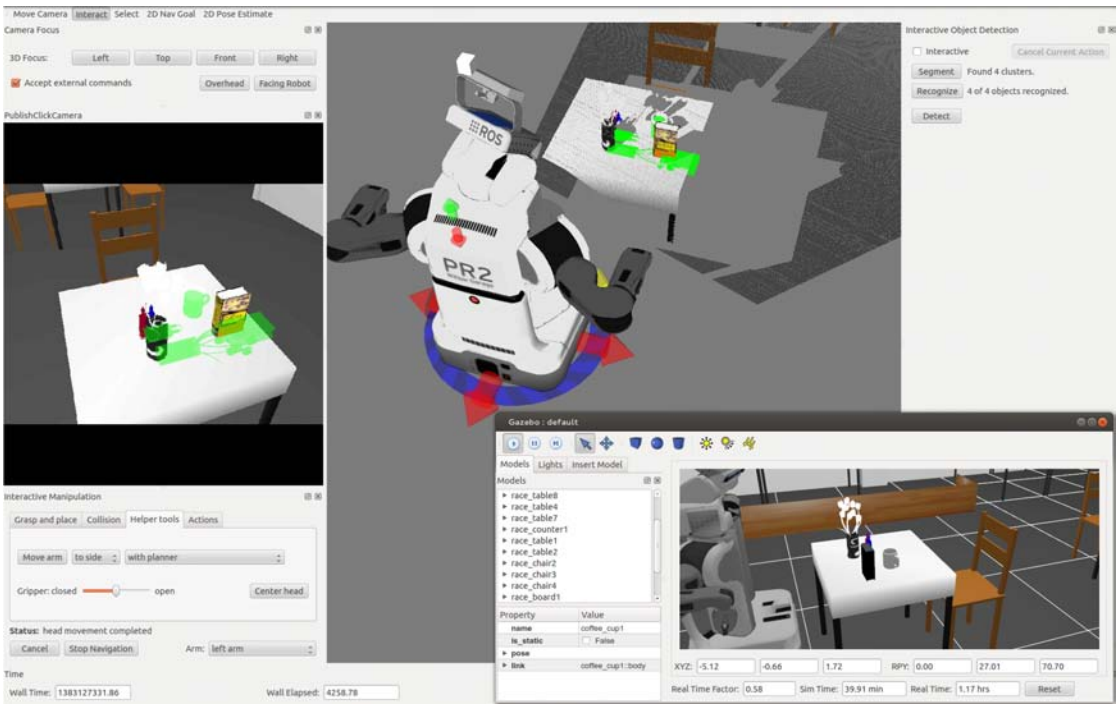


Abbildung 6.17.: Die Szene aus Abbildung 6.16 mit veränderter Perspektive. Alle vier Objekte werden richtig erkannt.

durch die Kombination aus volumetrischen und farblichen Informationen sicher erkannt werden.

Die nächsten beiden Bilder visualisieren eine etwas kompliziertere Szene, die aus vier Objekten besteht. Ein Messer, ein Löffel und eine Tasse bilden dabei den Hintergrund und werden durch ein Buch verdeckt. Auch hier gehen wir auf die Bewegung der Plattform nicht ein. Die Abbildungen 6.22 und 6.23 visualisieren das Vorgehen.

Zuerst wird das Buch mithilfe des SIFT-/SURF-Algorithmus eindeutig erkannt. Wie bereits erwähnt, wird im nächsten Schritt die 3-D-Position des Buches aus der 2-D-Position der gefundenen Merkmale berechnet. Als nächstes werden diese Daten für die Berechnung der Griffe verwendet. Nach der Kalkulation wird das Buch durch den Roboter gegriffen. Dabei offenbart sich das Problem, dass das Gewicht des Buches mit Berücksichtigung der Hebelwirkung zu groß ist. Der Roboterarm benötigt zum Anheben des Buches einen kurzen Schub, erst dann kann das Buch zur Seite gefahren werden. Wird die Erkennungspipeline noch einmal auf die aktualisierte Szene angewandt, werden alle restlichen Objekte sicher erkannt als Kombination aus Farb- und Formdetektoren. Die Szene ist erfolgreich analysiert worden.

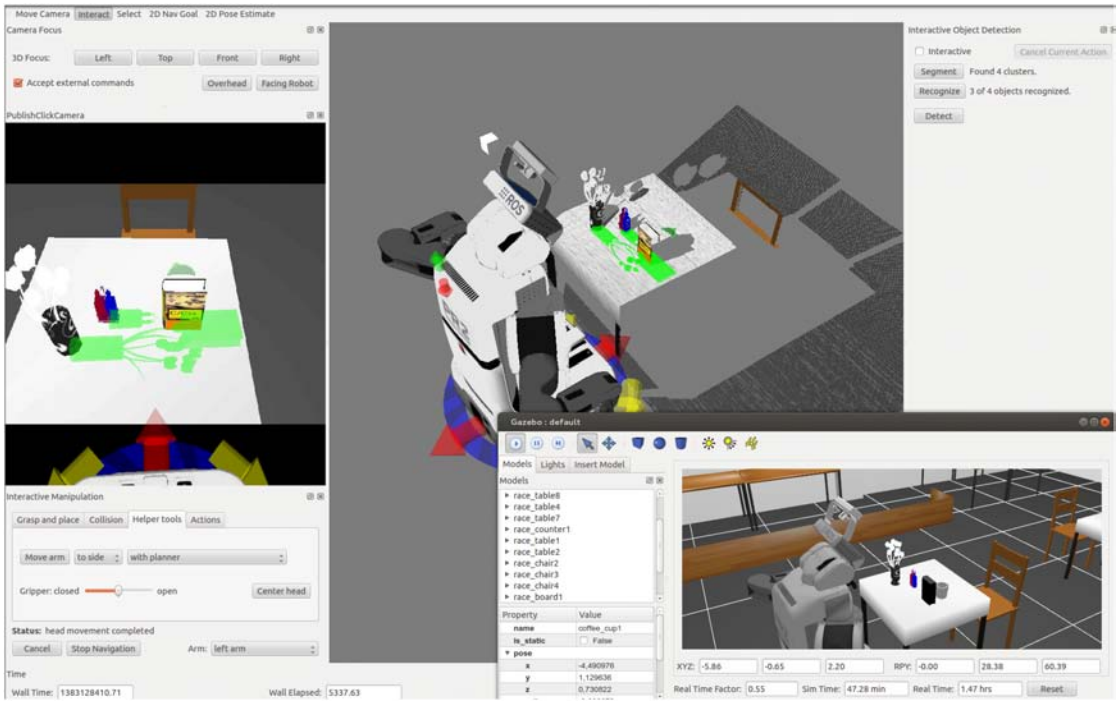


Abbildung 6.18.: Die Auflösung der Verdeckung basiert auf der in Abbildung 6.16 dargestellten Szene. Die Vase wurde durch den aktiven Eingriff in die Szene seitlich verschoben. Dadurch konnte der bis dahin verdeckte Ölspender erkannt werden. Nur der Kaffeebecher bleibt aufgrund der bestehenden Verdeckung unerkannt.

Die nächsten vier Abbildungen demonstrieren die Anwendung der kompletten Pipeline unter realen Bedingungen. Die Anordnung der Szene ähnelt der aus der Abbildung 6.23, erweitert um einen Teller und eine Gabel, damit sind hier sechs Objekten beteiligt.

Das Buch verdeckt alle anderen Objekte innerhalb der Szene und wird genauso wie im vorangegangenen Beispiel erkannt. Im ersten Durchlauf wird das Buch über SIFT-/SURF-Features erkannt, aber nicht über die volumetrischen Detektoren, wegen der zu nah stehenden Tasse. Außerdem werden das Messer und der Löffel über den ICP-Algorithmus und die Farbdetektoren erkannt. Die Bewegung der Plattform nach rechts bringt leider keine weiteren Erkenntnisse. Bei der Bewegung in die entgegengesetzte Richtung wird nur die Tasse erkannt. Da immer noch einige Cluster nicht richtig zugeordnet wurden, wird das Buch gegriffen und beiseite gefahren. Alle Objekte, bis auf den Teller, werden nun detektiert und richtig erkannt. Bei der Erkennung des Tellers stößt der Sensor (ASUS Xtion PRO LIVE) an seine Grenzen, der Teller liegt zu weit entfernt und ist viel zu flach. Einige gefundene Voxel werden dem Tisch zugeordnet, die ande-

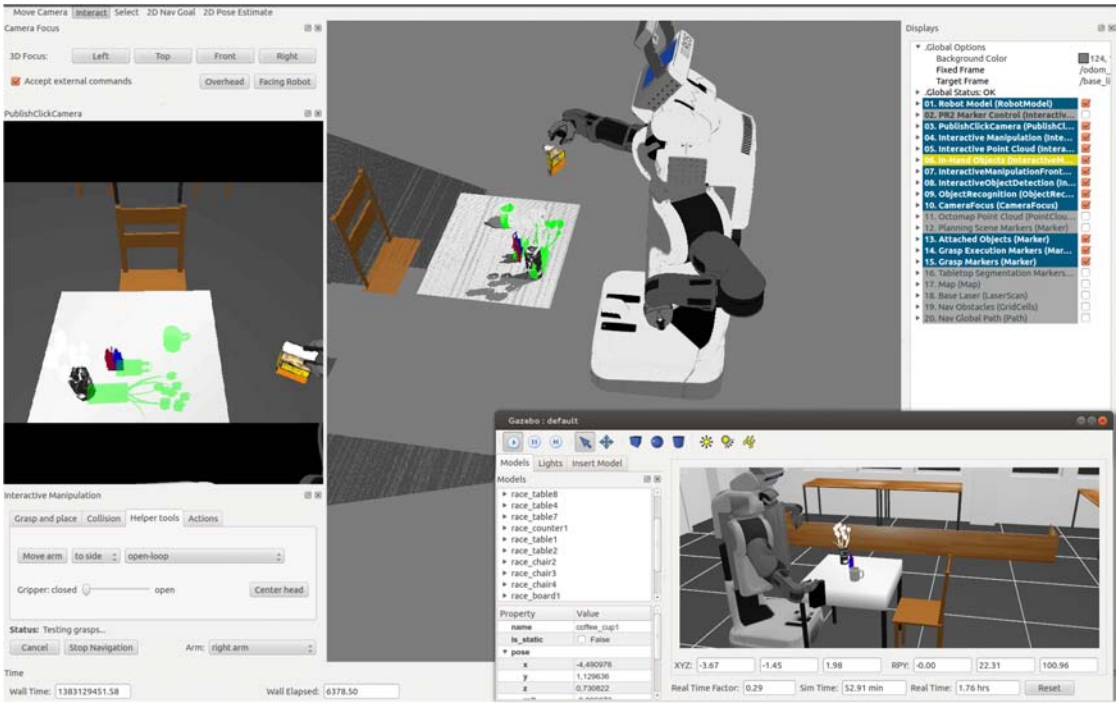


Abbildung 6.19.: Eine weitere Möglichkeit, um die bestehende Verdeckung aufzulösen, zeigt das Szenario aus Abbildung 6.18. Das Buch wird gegriffen und der Roboterarm zur Seite gefahren. Das Buch verbleibt während einer erneuten Analyse der Szene in der Hand. Alle Objekte wurden segmentiert und erkannt.

ren durch die Reflexionen verworfen. Die übriggebliebenen Voxel reichen nicht aus, um den Teller eindeutig zu erkennen. Somit ist die Szene erfolgreich analysiert worden, fünf von sechs Objekten wurden erfolgreich detektiert und erkannt. Die Verdeckung konnte aufgelöst werden.

Damit stellt die Kombination beider Möglichkeiten ein mächtiges Werkzeug zur Szenenanalyse dar. Durch die permanente Abwechslung von Perspektivenänderung und aktivem Eingreifen wird die Szene umfangreicher analysiert und physikalisch vereinfacht. Somit können im positiven Fall alle beteiligten Objekte erkannt werden. Der Engpass wird dabei durch die Objekte gebildet, die zum Beispiel nicht gegriffen oder gehoben werden können, weil sie beispielsweise zu groß, zu flach oder zu schwer sind. Auch die Kalkulation der Griffe stellt kein triviales Problem dar.

Eine andere Herausforderung ist die Auflösung der Sensoren, so ist z. B. das Erkennen der Zinken einer Gabel mit einem Kinect-Sensor schlichtweg unmöglich. Eine weitere bedeutende Problematik ist dann gegeben, wenn keiner der Detektoren ein Objekt in der

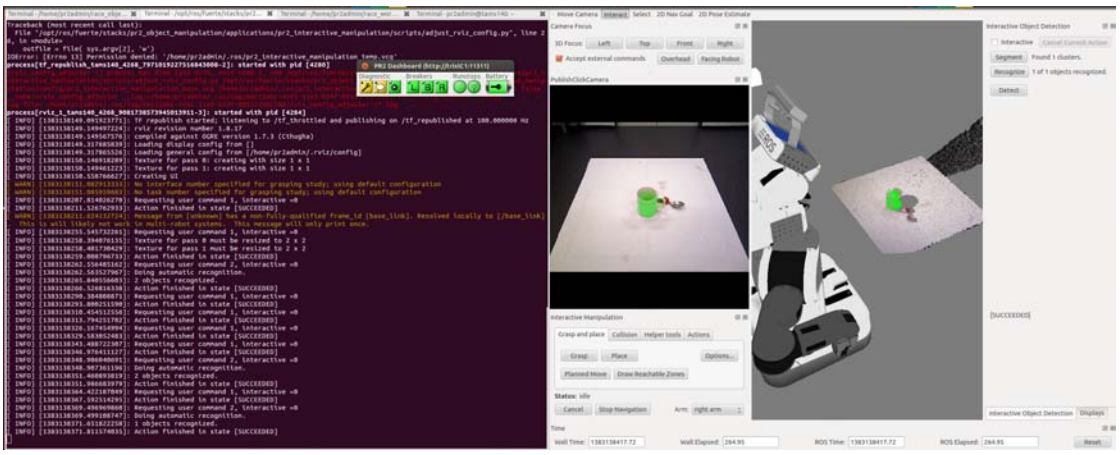


Abbildung 6.20.: Einfache Szene, eine Tasse verdeckt einen Löffel. Durch einen ausreichenden Abstand zwischen den beiden Objekten konnte die Tasse problemlos erkannt werden. Für die Erkennung des Löffels reicht die vorhandene Information aber nicht aus.

Menge eindeutig erkennen kann. Hier kann durch ein „blindes“Eingreifen in die Szene nachgeholfen werden. Beispielsweise kann das Cluster verschoben oder in dessen Mitte eingegriffen werden. Das Ziel ist, die ursprüngliche Anordnung der Objekte zu verändern und dadurch die Szene erfolgreich analysieren zu können. Es wird offensichtlich, dass das Verfahren sehr zeitaufwendig sein kann. Diese Problematik kann durch den Einsatz von ausgeklügelten Algorithmen oder einer Lernlogik, die nicht Bestandteil der vorliegenden Arbeit sind, minimiert werden. Wie bereits beschrieben, wird im negativen Fall ein koloriertes 3-D-Modell erstellt und der Supervisor benachrichtigt.

Der vorgestellte Ansatz zur möglichen Lösung der partiellen/totalen Verdeckung ist vielversprechend und bietet ein enormes Potenzial. Eine tiefer gehende Untersuchung würde aber den Rahmen dieser Dissertation sprengen und wird deswegen nicht weiter unternommen. Die präsentierten Experimente bestätigen die Annahmen des Autors eindeutig und zeigen realistische Strategien auf, um die beschriebenen Probleme durch Verdeckung zu minimieren, wenn nicht sogar komplett zu lösen. Besonders im Bereich der Servicerobotik könnte auf dieser Basis ein Paradigmenwechsel erfolgen. Ein Roboter wird durch die Wechselwirkung von aktivem Sehen und Eingreifen in die Szene in die Lage versetzt, die Komplexität der Objekterkennung und auch der kompletten Szenenanalyse zu reduzieren. Dadurch kann die Szenenanalyse autarker, zielorientierter und erfolgreicher gestaltet werden.

Aufgrund der nicht ausreichenden Anzahl der zur Verfügung stehenden Detektoren sowie der Probleme des Manipulators mit der Auflösung der Verdeckung, die durch klei-

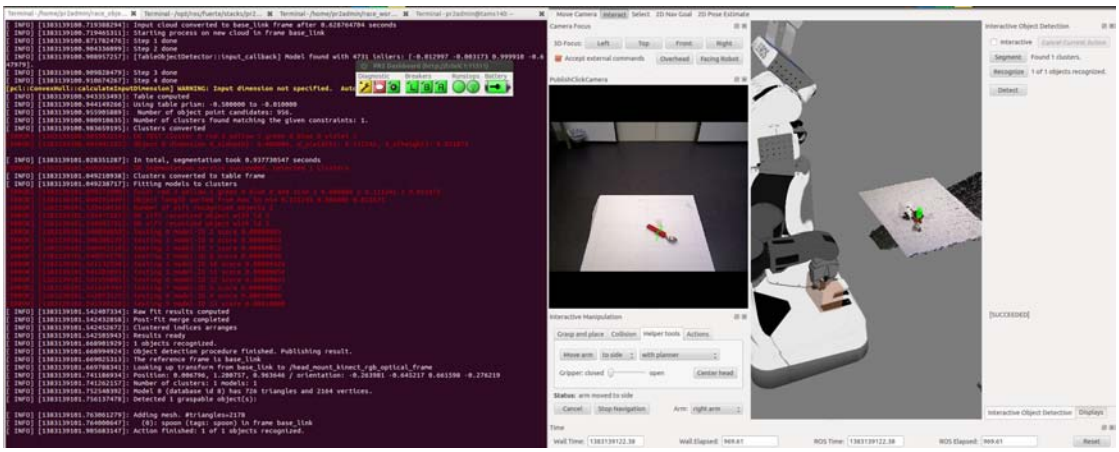


Abbildung 6.21.: Nachdem die Tasse erkannt worden ist, wird einer der möglichen Griffe aus der Datenbank entnommen und ausgeführt. Bei der erneuten Szenenanalyse wird der Löffel sicher erkannt.

nere Objekte verursacht wird, kann der Ansatz nur sporadisch evaluiert werden. Dennoch zeigen die aufgeführten Beispiele deutlich das enorme Potenzial des Verfahrens. Soweit die Szene nicht berührungslos analysiert werden muss, stellt die Methode ein mächtiges Tool dar. Für ein sich dynamisch veränderndes Szenario mit vielen unbekannten Objekten ist das Verfahren definitiv nützlich und kann die Effizienz der Objekterkennung bzw. die Szenenanalyse deutlich steigern. Eine genaue Auswahl der Detektoren sowie der eingesetzten Roboterplattform ist notwendig und benötigt sicherlich viel Zeit. Läuft die Analysepipeline, bietet der aktive Eingriff in die Szene ein Verfahren, mit dem die Objekte erkannt und mögliche Verdeckungen aufgelöst werden können. Darüber hinaus kann es mit unbekannten Objekten umgehen und sogar beim Lernen neuer Objekte aktiv unterstützen.

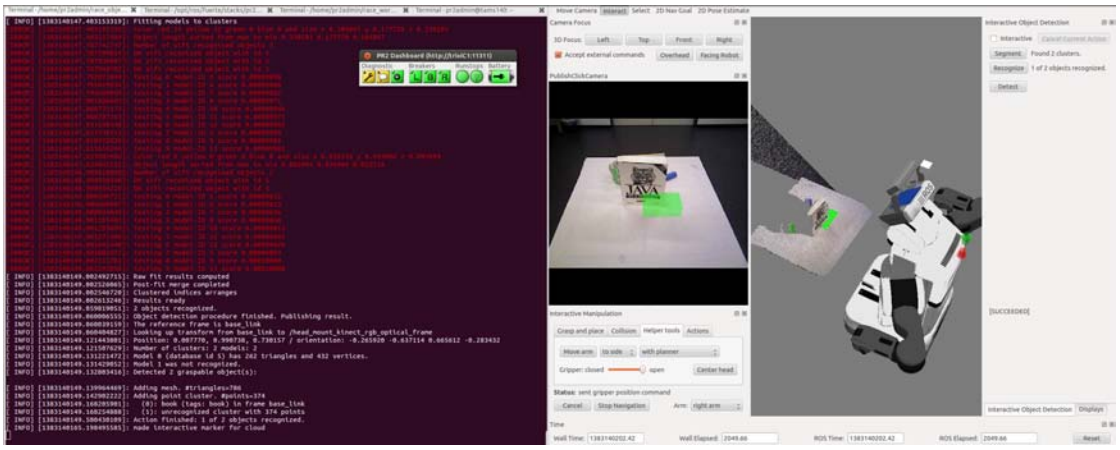


Abbildung 6.22.: Einfache Szene, ein Buch verdeckt eine Tasse, einen Löffel und ein Messer. Durch den SIFT-/SURF-Algorithmus wird das Buch eindeutig erkannt. Ein Teil der Tasse wird detektiert, kann aber nicht zugeordnet werden. Ein Teil des Messers ist zwar sichtbar, reicht aber nicht einmal für die Detektion aus, weil er zu klein und zu weit entfernt ist.

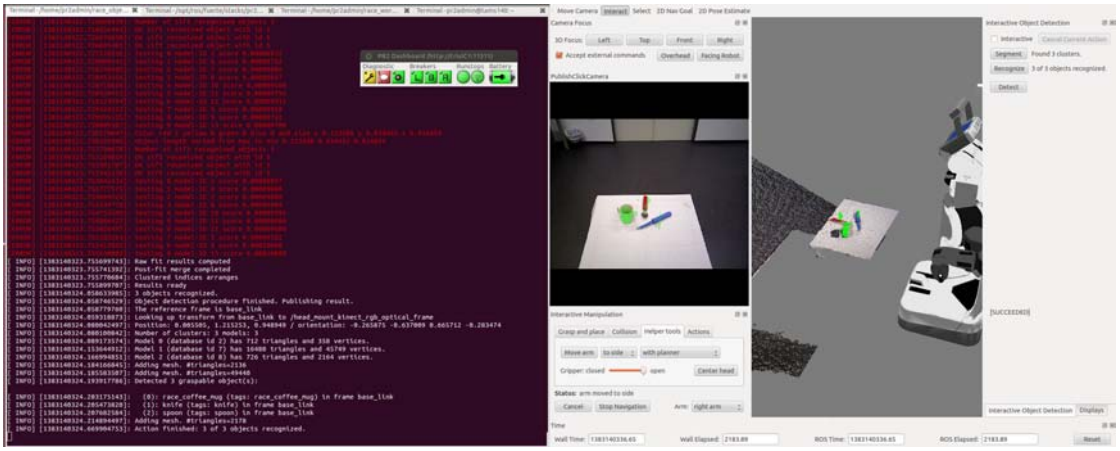


Abbildung 6.23.: Nachdem das Buch aus der Szene durch einen Eingriff entfernt worden ist, kann die Szene bei einer erneuten Analyse erfolgreich verarbeitet werden. Alle Objekte sind sicher erkannt worden.

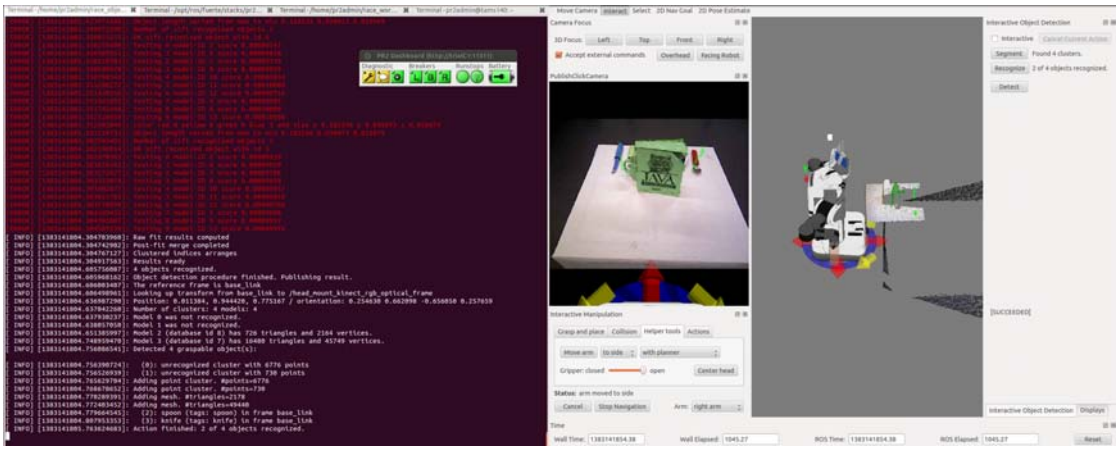


Abbildung 6.24.: Eine komplexe Szene, bestehend aus mehreren Objekten. Aufgrund der Verdeckung werden nur drei unverdeckte Objekte erkannt: ein Messer, ein Löffel und ein Buch. Da das Buch genügend SIFT-/SURF-Merkmale aufweist, wird es in allen nachfolgenden Szenen sicher erkannt.

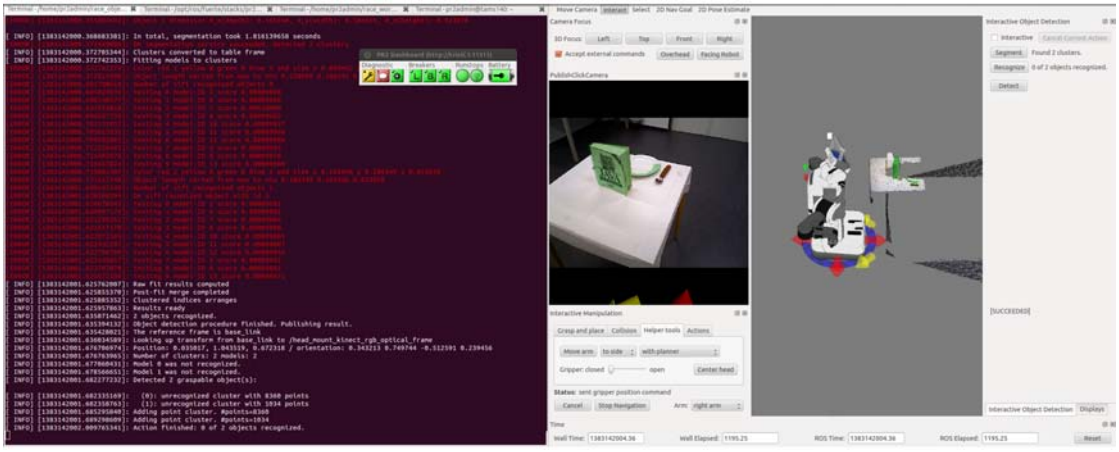


Abbildung 6.25.: Eine komplexe Szene aus der Abbildung 6.24. Änderung der Perspektive durch Bewegung des Roboters zur Seite. Der Teller ist nicht mehr verdeckt, dennoch ist die Entfernung des Roboters zu groß und der Teller zu flach. Viele Punkte werden fälschlicherweise dem Tisch zugeordnet, was die Erkennung des Tellers verhindert.

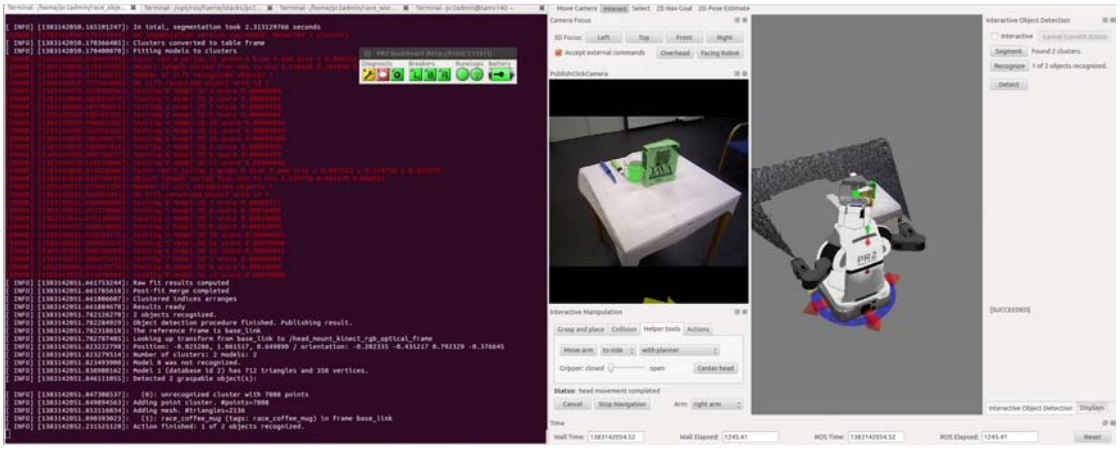


Abbildung 6.26.: Eine komplexe Szene aus den Abbildungen 6.24 und 6.25. Durch die Verschiebung der Perspektive zur anderen Seite wird die Tasse ausreichend detektiert und erfolgreich erkannt.

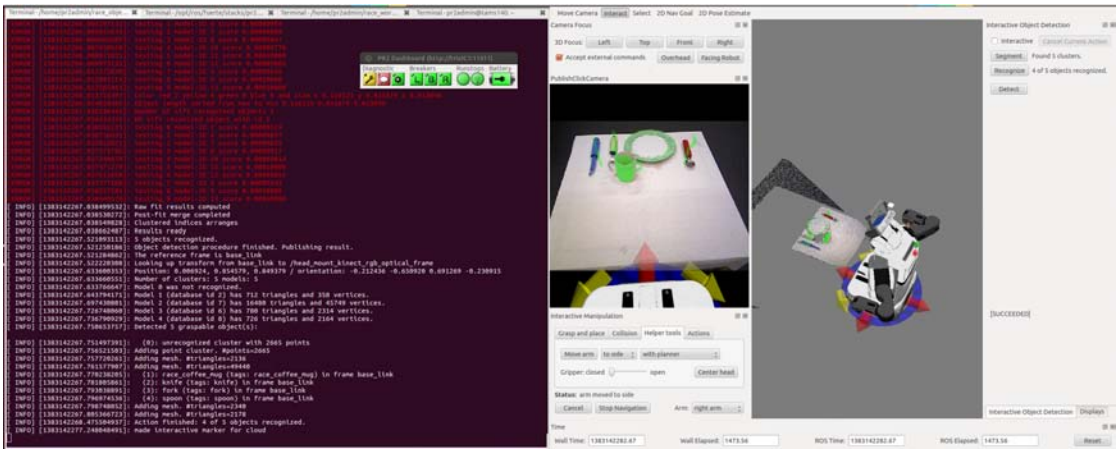


Abbildung 6.27.: Eine komplexe Szene aus den Abbildungen 6.24, 6.25 und 6.26. Durch den aktiven Eingriff wird die Verdeckung aufgelöst, da das Buch mithilfe des Manipulators aus der Szene entfernt wird. Alle Objekte bis auf den Teller (vgl. Abbildung 6.25) werden durch unterschiedliche Detektoren erkannt.

Kapitel 7

Fazit und Ausblick

Die folgenden drei Textabschnitte sollen die vorliegende Dissertation abschließen und einen Überblick über die möglichen Erweiterungen geben. Angefangen wird mit einer komprimierten Zusammenfassung und einer kurzen Gegenüberstellung zwischen geplanten und realisierten Zielen. Es folgt das Fazit, das auf die Zusammenfassung zurückgreift, diese ausführlicher darstellt, von derselben abstrahiert und dadurch eine globale Auswertung und Einordnung der vorliegenden Dissertation ermöglicht. Danach wird, basierend auf der Zusammenfassung und dem Fazit, ein Ausblick präsentiert.

7.1. Komprimierte Zusammenfassung

Das Ziel dieser Arbeit war es, ein System zu entwerfen und zu realisieren, das die Objekterkennung verbessert und Möglichkeiten zur Auflösung der Verdeckung bereitstellt. Eine entsprechende Architektur, basierend auf einer mobilen Plattform, ist vorgestellt und realisiert worden. Der Roboter ermöglicht die Navigation und die Lokalisierung in einer bekannten sowie unbekannten Umgebung. Die Anzahl und Auswahl der Sensoren kann variabel gestaltet werden, dabei soll versucht werden, alle Objekteigenschaften durch die vorhandenen Sensoren abzudecken und damit die Objekterkennung zu verbessern. Die Sensoren werden teils über bereits existierende und teils über eigens entwickelte Algorithmen zueinander registriert, was eine exakte Zuordnung der Pixel/Voxel unterschiedlicher Sensoren erlaubt. Für die Repräsentation der resultierenden Daten wurde eine Struktur entwickelt, die die 3-D-Information um die farblichen Eigenschaften erweitert. Damit stehen für die Objekterkennung deutlich mehr registrierte Daten zur Verfügung, was die Wahrscheinlichkeit einer Erkennung extrem erhöht. Die Struktur ist dynamisch und kann nach Bedarf erweitert werden, zum Beispiel durch die Temperaturangaben einer Wärmebildkamera. Ein weiterer Vorteil der Datenstruktur ist die Möglichkeit, nur auf 2-D-, auf 3-D-, oder auf alle Daten zugreifen zu können.

Dieser Vorteil kann von den Detektoren genutzt werden. So können alle Detektoren, unabhängig davon, ob sie auf einem Bild oder einer Punktwolke operieren, auf eine Struktur zurückgreifen. Werden die Detektoren verwendet, die auf unterschiedlichen Objekteigenschaften basieren, entsteht eine Menge an Ergebnissen. Hier findet die zweite Fusion, nach dem Zusammenführen der Sensordaten, statt. Die Ergebnisse der Detektoren werden über eine gewichtete Abstimmung zusammengefasst. Auch hier können einerseits verschiedene Detektoren verwendet werden. Andererseits bietet die gewichtete Abstimmung einen weiteren Vorteil, so können verlässliche Detektoren bei der Entscheidung bevorzugt werden.

Außerdem ist das Konzept des aktiven Sehens realisiert worden. Die Roboterplattform wird bewegt, was die Änderung der Perspektive auf die zu analysierende Szene zur Folge hat. Eine Erweiterung dieses Konzepts durch das sogenannte „Next Best View“(NBV) wäre wünschenswert, wurde aber im Rahmen der vorliegenden Dissertation aus zeitlichen Gründen nicht realisiert. Durch die Registrierung der Sensoren können unterschiedliche Perspektiven in eine Szene überführt werden.

Ein weiterer interessanter Ansatz ist die Auflösung der partiellen/totalen Verdeckung durch das vorgestellte System. Die Realisierung basiert auf mehreren voneinander unabhängigen Methoden. Die erste basiert auf der Änderung der Perspektive, in der Hoffnung, die verdeckten Objekte voneinander separieren zu können. Einen weiteren Ansatz stellt der aktive Eingriff in die Szene durch einen Robotermanipulator dar. Die Methode wird gestartet, wenn ein verlässlicher Manipulator ein Objekt sicher erkennt, die anderen Daten aber mit der Datenbank nicht übereinstimmen. Als Beispiel wäre ein SIFT-/SURF-Detektor zu nennen, der genügend Merkmale findet, wobei weitere Daten, wie zum Beispiel die Abmessungen, aber nicht mit den Werten für das erkannte Objekt in der Datenbank übereinstimmen. Leider standen nur wenige verlässliche Detektoren zur Verfügung, somit konnte der Ansatz nicht ausreichend evaluiert werden. Dennoch erweisen die Idee und die ersten Tests unter realen und simulierten Bedingungen die Methode als sehr vielversprechend. Der Nachteil des vorgestellten Verfahrens ist die fehlende Möglichkeit einer berührungslosen Abtastung der Szene, falls es zu einem Eingriff durch den Manipulator kommt.

7.2. Fazit

Die vorliegende Dissertation beschäftigt sich mit dem gesamten Zyklus der Objekterkennung. Es resultiert ein Erkennungssystem, das aktive und passive Komponenten beinhaltet. Im Mittelpunkt steht ein Service-Roboter, der mit mehreren beweglichen Sensoren und einem oder mehreren Aktuatoren ausgestattet ist.

Die Auswahl der Sensoren ist dabei extrem wichtig, einerseits sollen die Sensoren möglichst viele heterogene Eigenschaften eines Objekts empfangen und verarbeiten können. Andererseits sollen die Sensoren für die aktive Wahrnehmung beweglich sein. Sind die

beiden Anforderungen erfüllt, stehen genügend Informationen für die Objekterkennung zur Verfügung. Die Schwierigkeit, die sich dabei offenbart, ist die Zuordnung verschiedener Sensordaten zu einem Objekt. In der präsentierten Arbeit wird dies durch Zuhilfenahme der Multi-Sensor-Fusion gelöst. Dabei wurden mehrere Algorithmen und Methoden entwickelt, die es erlauben, heterogene Sensordaten zueinander in Relation zu setzen. In der Literatur ist dieses Vorgehen als Registrierung bekannt. Dieser Prozess ermöglicht es, die Daten in ein gemeinsames Koordinatensystem zu transformieren. Das Ergebnis ist dabei eine kolorierte 3-D-Punktwolke. Diese Repräsentation erlaubt den Einsatz von 2-D- sowie 3-D-Detektoren zur Objekterkennung. Außerdem ist das Wechselspiel zwischen unterschiedlichen Objekteigenschaften möglich, so kann auch nach allen zur Verfügung stehenden Objekteigenschaften gesucht und diese können abschließend zu einer Entscheidung zusammengefasst werden.

Eine weitere wichtige Komponente ist die Auswahl der Detektoren. Zur Erzielung des bestmöglichen Ergebnisses sollen dabei möglichst alle Objekteigenschaften durch die Detektoren abgedeckt werden. Des Weiteren soll bei der Auswahl der Detektoren auf die Verlässlichkeit, Effektivität, Effizienz, Robustheit sowie zeitliche Performanz geachtet werden. Sind diese Kriterien erfüllt und die Sensordaten fusioniert, stehen für die Objekterkennung mehr Informationen zur Verfügung, als die Auswertung einzelner Sensordaten erbracht hätte. Auch die Effizienz einer Nutzung qualitativ minderwertiger Sensoren kann dadurch gesteigert werden.

Für eine weitere Verbesserung dient das aktive Sehen. Durch die Veränderung der Sensorposition können unterschiedliche Perspektiven auf die zu analysierende Szene in die Entscheidung miteinbezogen werden, was die Erfolgschancen der Objekterkennung steigert und bei einem ausreichenden Abstand zwischen einzelnen Objekten die Problematik der Verdeckung teilweise entschärfen, wenn nicht sogar komplett lösen kann.

Eine weitere Neuerung dieser Arbeit ist der aktive Eingriff in die Szene. Liefern die oben beschriebenen Verfahren keinen Erfolg, kann, realisiert durch die Manipulatoren eines Roboters, in die Szene aktiv eingegriffen werden. Dabei kann ein bereits erkanntes oder detektiertes Objekt aus der Szene entfernt werden. Schon eine geringe räumliche Veränderung der beteiligten Objekte führt häufig zu einer erfolgreichen Erkennung. Dabei werden die Auswirkungen der partiellen/totalen Verdeckung erneut minimiert. Des Weiteren bietet dieses Vorgehen die Möglichkeit, die Komplexität einer Szene zu reduzieren und diese damit in vielen Fällen überhaupt erstmals für die Objekterkennung lösbar zu machen. Aber auch wenn zum Schluss einige Objekte unerkannt bleiben, kann solches Vorgehen nützlich sein. So kann von den gefundenen Clustern ein koloriertes 3-D-Modell erstellt und an die Datenbank und/oder den Operator geschickt werden. Der Operator hat dann die Möglichkeit, die fehlenden Daten zu ergänzen sowie auch ein Objekt zu kategorisieren. Damit kann die Performanz des Gesamtsystems erhöht werden.

Die nächste Schwierigkeit, die sich dabei offenbart ist die gemeinsame Anwendung aller genannten Verfahren. Für diesen Zweck wurde eine Architektur entwickelt, die auf mehreren Ebenen eine gemeinsame Lösung ermöglicht. So werden die Detektoren anhand

ihrer Erfolgsraten gewichtet. Eine kollektive Entscheidung wird über die sogenannte gewichtete Abstimmung erreicht.

Ist ein Konsens unmöglich, wird, basierend auf einem regelbasierten System, anhand der Kosten (Energie und Zeit) über den Einsatz des aktiven Sehens oder des Eingreifens entschieden. Dieser Zyklus, hier als Analyse oder Szenenanalyse bezeichnet, besteht aus der Wahrnehmung, Detektion der Cluster sowie Erkennung, und ist beliebig oft wiederholbar. Die Evaluation bestätigt die gemachten Annahmen und hebt die Vorteile des entwickelten Systems gegenüber gängigen Verfahren deutlich hervor. Des Weiteren eröffnet ein solches System die Möglichkeit, den Einsatz des Roboters komplett zu automatisieren, auch ein lebenslanges Lernen wird dadurch realisierbar. Dennoch hat ein solches System auch Nachteile, die hauptsächlich bei dem Mehraufwand durch die Einpflege der Objekte in die Datenbank sowie bei den Ausführungszeiten der Analyse auszumachen sind. Ein weiterer möglicher Nachteil ist die physikalische Veränderung der Szene, was die Wiederholbarkeit beeinträchtigt und eine Vorüberlegung hinsichtlich der Ziele der Applikation erfordert.

Wird die Dissertation zusammengefasst, so resultiert ein, auf einer Roboterplattform basiertes, Erkennungssystem, das autark eine Szene analysieren und die beteiligten Objekte erkennen kann. Die Wahrscheinlichkeit einer richtigen Kategorisierung steigt durch die Verschiedenartigkeit der verwendeten Verfahren und die Qualität der Detektoren enorm. Dem Autor ist ein vergleichbares System nicht bekannt, was die Auswertung vergleichbarer Arbeiten bestätigt.

Die entwickelte Szenenanalyse eröffnet zuvor nicht verfügbare Möglichkeiten und erinnert in seinem Kern an die menschliche Wahrnehmung, vgl. Appendix B. Die Erkennung läuft in mehreren Schritten auf unterschiedlichen Ebenen ab. Als Ergebnis wird am Ende eine gemeinsame Entscheidung präsentiert, aber auch die Ausgabe mehrerer Alternativen, sortiert nach der Wahrscheinlichkeit der Zugehörigkeit zu einem bestimmten Objekt, ist möglich. Das System ist in das Framework ROS integriert, greift auf einige bereits vorhandene Komponenten zurück und wird der Öffentlichkeit zur Verfügung gestellt. Die Evaluationsergebnisse sind vielversprechend, übertreffen die Ergebnisse vergleichbarer Arbeiten und bestätigen die getroffenen Entscheidungen bei Entwurf, Spezifikation und Realisierung. Sogar das seit Jahrzehnten bekannte Problem der totalen/partiellen Verdeckung kann damit, wenn nicht vollständig, so doch zumindest teilweise gelöst werden. Des Weiteren besitzt das System ein großes Potenzial zur Erweiterung, Optimierung und Verfeinerung aller am Prozess beteiligter Ebenen.

Dennoch hat die in der vorliegenden Arbeit vorgeschlagene und untersuchte Strategie auch Nachteile. So wird ein System, wie zum Beispiel beim RACE-Projekt, dadurch deutlich komplexer und schwer überschaubar. Die Evaluation solcher Systeme ist kompliziert und nur wenig erforscht, dennoch nicht unmöglich. Die Publikation [ZRP+13] stellt eine mögliche Evaluation solcher Systeme vor, die für das oben genannte Projekt entwickelt und erfolgreich getestet worden ist.

Im nächsten Kapitel „Ausblick“werden mögliche Erweiterungen und / oder Optimie-

rungen des gesamten Systems ausführlich vorgestellt und diskutiert.

7.3. Ausblick

Nachdem die Ergebnisse der Dissertation im vorangegangenen Abschnitt zusammengefasst worden sind, soll in diesem Kapitel auf mögliche Erweiterungen, Optimierungen und Verfeinerungen eingegangen werden. Natürlich ist das präsentierte System nur ein Prototyp und beinhaltet viel Potenzial. Zuerst werden die einzelnen Ebenen im Hinblick auf die möglichen Erweiterungen betrachtet, später wird auf die konzeptionellen Erweiterungen eingegangen.

Auf der Ebene der Sensorik soll versucht werden, möglichst alle Objekteigenschaften zu erfassen. Daher war schon angedacht, aber nicht mehr im Rahmen dieser Dissertation realisiert, das resultierende System durch eine Wärmebildkamera zu ergänzen. Diese zusätzliche Modalität würde es zum Beispiel ermöglichen, ein heißes Objekt auch bei vorhandener Verdeckung durch kältere Objekte detektieren zu können. Der Hauptvorteil, bezogen auf das EU-Projekt RACE, liegt aber in der Möglichkeit, menschliche Extremitäten eindeutig erkennen zu können. Eine flach auf einem Tisch liegende Hand ist kaum zu erkennen und wird durch die meisten Verfahren ausgefiltert. Hier wäre eine deutliche Aussage möglich. Damit würden die Daten eines Objekts um eine weitere Dimension, nämlich die Temperatur, erweitert. Bei den Detektoren und bei den Entscheidungen könnte noch mehr auf Effektivität und Intelligenz geachtet werden. Auch die Anwendung gruppierter Editoren, basiert auf der zeitlichen Performanz, wäre denkbar. Eine zusätzliche Analyse mit Einfluss des semantischen Aspekts würde das System noch einmal verbessern. Dabei denkt der Autor an solche Situationen, in denen durch die erkannten Objekte auf eine bestimmte Umgebung rückgeschlossen werden kann. Ist ein detektiertes Objekt danach nicht eindeutig kategorisierbar, kann die Umgebungsinformation nützlich werden. Außerdem kann die Reihenfolge beim Vergleich mit den Objekten in der Datenbank optimiert werden.

Auch taktile Sensoren können zu einem positiven Ergebnis beitragen, so kann zum Beispiel durch ein Schieben über die Tischoberfläche durch Zuhilfenahme der Reibungskoeffizienten auf das Gewicht des Objekts rückgeschlossen werden. Sofern das Gewicht als Parameter in der Datenbank vorhanden ist, können die Werte zwecks Erkennung verglichen werden.

Eine der wichtigsten Anwendungen und eins der bedeutendsten Evaluationssysteme für diese Dissertation ist das bereits vorgestellte EU-Projekt RACE, das sich mit dem Lernen eines Service-Robotersystems aus eigenen Erfahrungen und dessen Nutzung in einer Restaurantumgebung beschäftigt. Wird das hier Erarbeitete zusammengefasst, kann die Objekterkennung durch ein umfangreiches Volumen an Parametern und Objekteigenschaften dazu beitragen (und wird auch bereits dafür verwendet), um dieses Konzept erfolgreich realisieren zu können.

Dennoch gibt es kleine und naheliegende Möglichkeiten, das System zu verbessern. Als Beispiel wäre die Berechnung der Orientierung zu nennen, die definitiv schneller, genauer und effektiver sein sollte.

Wie bereits beschrieben, verwendet das System bereits viele innovative Ideen und ermöglicht eine vielfältige und umfangreiche Szenenanalyse. Die Ergebnisse zeigen das Potenzial eines solchen Systems deutlich auf. Natürlich entstehen in solch einem umfangreichen System viele Erweiterungs- und Optimierungsmöglichkeiten. Das System ist eher als richtungweisend gedacht, um die Kombination aus Sensordaten, Multi-Sensor-Fusion, Objekterkennung und Szenenanalyse auf eine weiterführende Ebene zu bringen und damit die Distanz zur semantischen Ebene zu verkürzen.

Nach Meinung des Autors ist es die einzige Möglichkeit, diese Aufgaben in angemessener Weise für die Servicerobotik und die Robotik im Allgemeinen lösen zu können.

Anhang A

Appendix

A.1. Visualisierung aller laufenden Transformationen

Die zwei folgenden Abbildung visualisieren die auf dem ROS basierende Struktur. Das erste Bild zeigt alle auf dem R2-Roboter verwendete Transformationen.

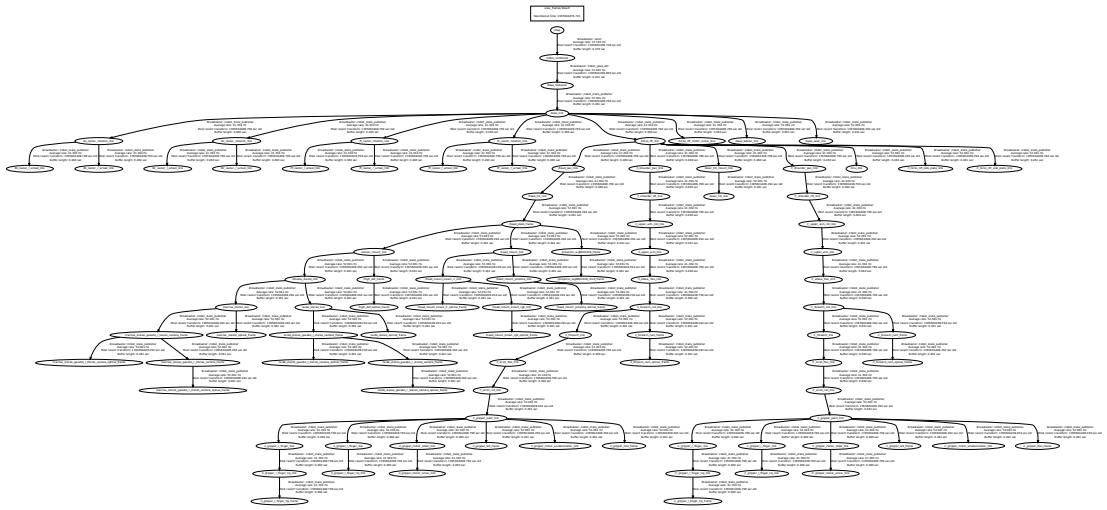


Abbildung A.1.: Visualisierung aller auf dem PR2-Roboter laufenden Transformationen.

Die zweite Abbildung visualisiert alle notwendigen und zur Laufzeit aktive Nodes.

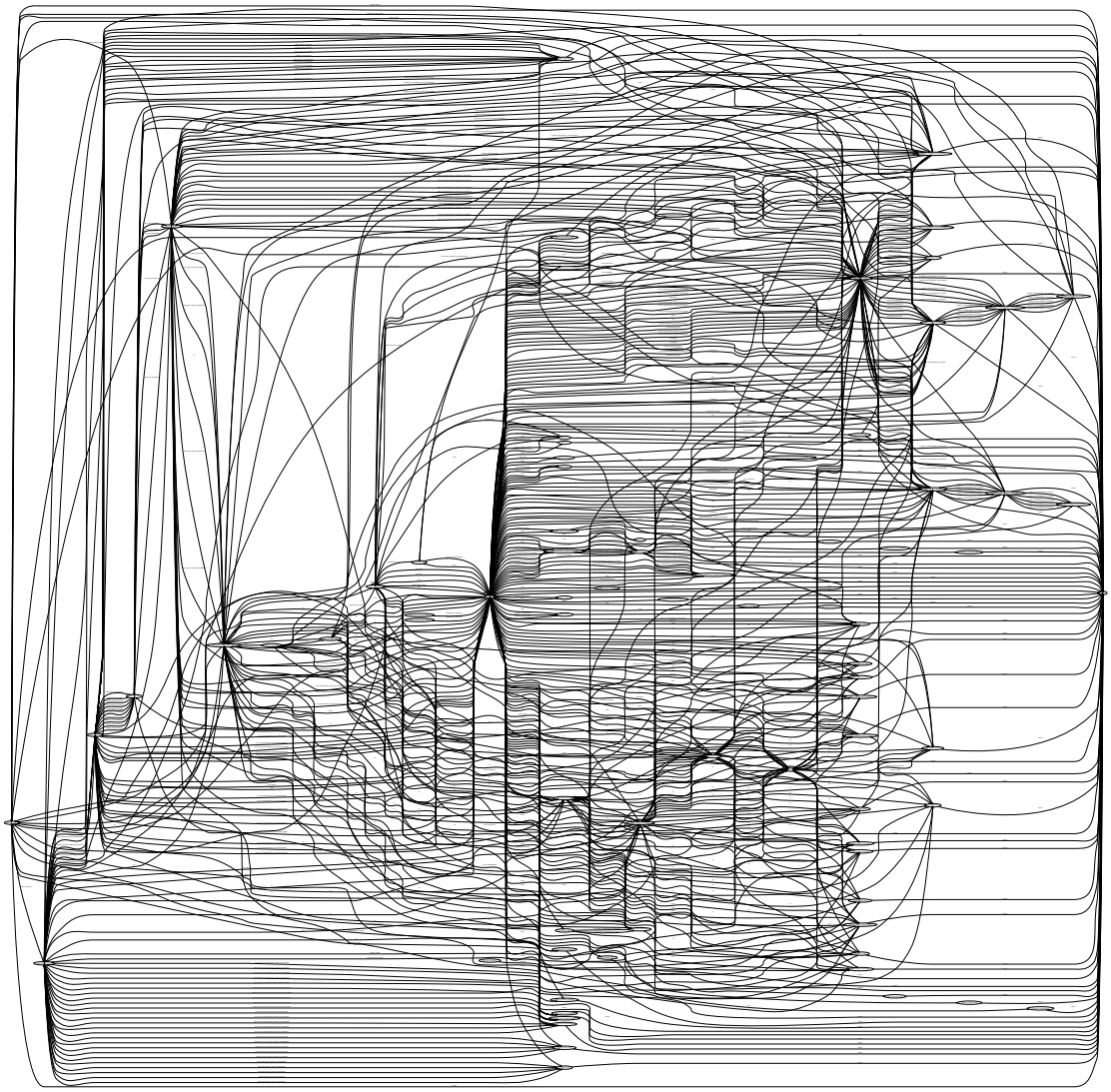


Abbildung A.2.: Visualisierung aller auf dem PR2-Roboter laufenden Prozesse (Nodes).

A.2. Wärmebildkamera als ein Erweiterungsmodul der Objekterkennung

Die Wärmebildkamera, die bereits im Kapitel 7.1 erwähnt worden ist, bietet eine interessante Erweiterung des vorgestellten Robotersystems. Die Kamera erlaubt es jedem Pixel ein Temperaturwert (t_n) zuzuweisen. Die bereits vorgestellte Struktur könnte dann um t_n zur $[x_1, y_1, z_1, r_1, g_1, b_1, t_1, ..., x_n, y_n, z_n, r_n, g_n, b_n, t_n]$ erweitert und genutzt werden. Innerhalb des RACE-Projekts könnte dann die Erkennung von warmen Objekten, wie zum Beispiel einer Tasse mit Kaffee oder Essen erleichtert werden. Unter anderem kann damit verbundene Verdeckungsauflösung profitieren, soweit andere beteiligte Objekte einen Temperaturunterschied aufweisen. Im Weiteren könnte die Temperaturinformation für die Planung als auch für die Bewertung bestimmter Aufgaben herangezogen werden. Ein Plan in dem eine Kaffeetasse mit zu niedriger Temperatur serviert wird, kann dann niedriger bewertet werden, als beim Servieren des Kaffees mit einer optimalen Trinktemperatur.

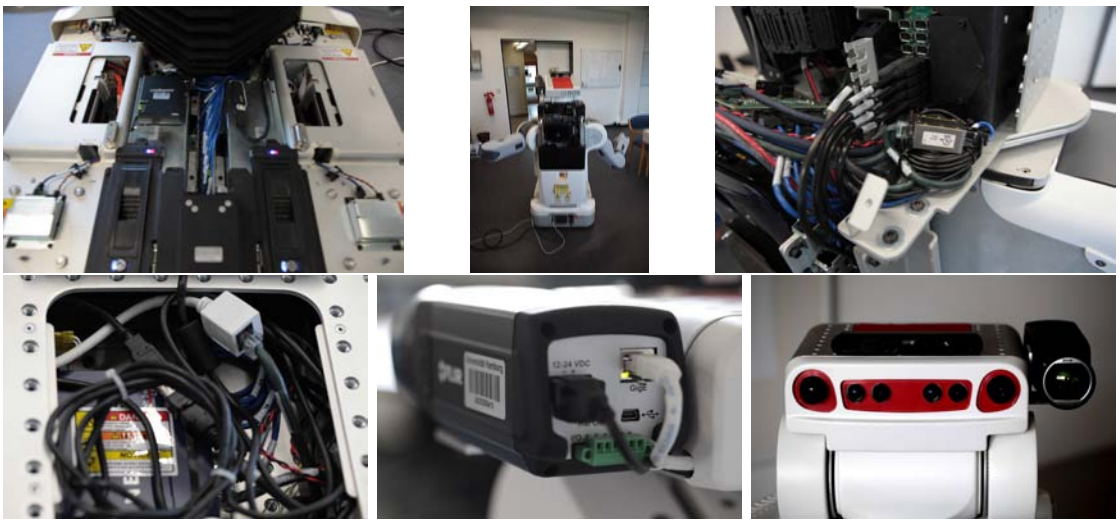


Abbildung A.3.: PR2 - Wärmebildkamera. Integration der Wärmebildkamera auf dem PR2-Roboter.

Eine weitere interessante Anwendung ist die Erkennung der Menschen sowie menschlicher Extremitäten. So stellt zum Beispiel eine flach auf dem Tisch liegende menschliche Hand ein extremes Problem für die Objekterkennung dar und birgt somit Gefahren. Es ist absolut kontraproduktiv, wenn ein Roboter versuchen würde einen heißen Kaffee oder ähnliches mangels Erkennung auf der menschlichen Hand abzustellen. Die Wärmebildkamera könnte hierbei Abhilfe schaffen, denn die Erkennung wäre sicher und

der Planer würde den Bereich sperren. Auf der anderen Seite wäre durch die spezifische Temperatur des menschlichen Körpers sogar eine Kategorisierung, basierend auf den bestimmten Schwellwerten, realisierbar.

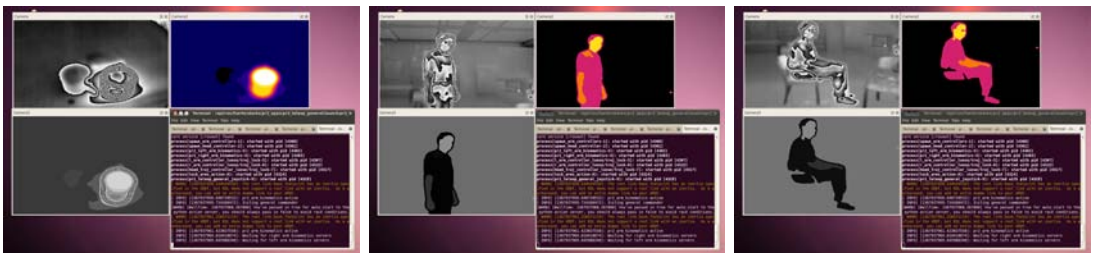


Abbildung A.4.: Wärmebildkamera - Visualisierung der Daten.

Die Abbildung A.3 stellt die Integration der Wärmebildkamera in das verwendete Robotersystem PR2 grafisch dar. Die Kabel sind im inneren des Roboters verlegt und schränken die Bewegung des Roboters nicht ein. Die Stromversorgung wird über die Roboterbatterie realisiert, so dass es keiner externen Stromquellen bedarf. Zur Minimierung der Spannungsspitzen wird ein Kupferkern verwendet. Das System läuft stabil; eine Minderung der Roboterlaufleistung war nicht auszumachen.

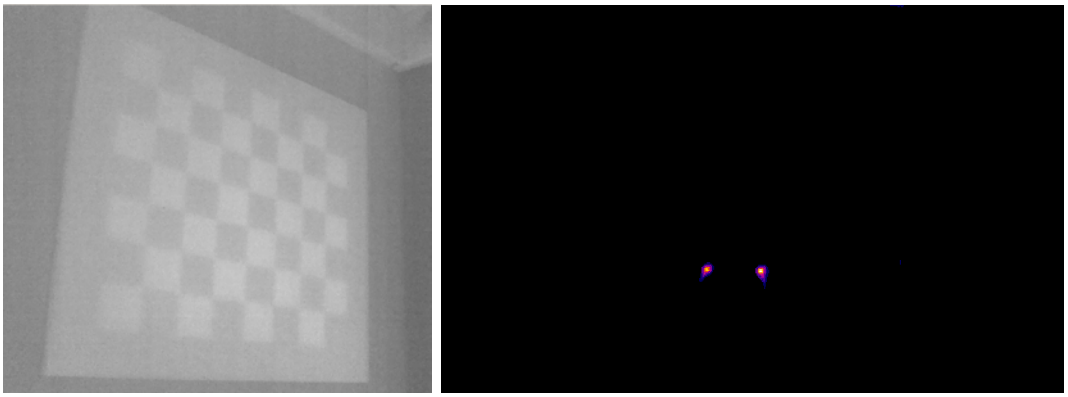


Abbildung A.5.: Wärmebildkamera - zwei Möglichkeiten die Wärmebildkamera innerhalb des PR2-Systems zu registrieren.

Die Abbildungsreihe A.4 veranschaulicht die von der Wärmebildkamera gelieferten Daten. Beide Abbildungen auf der linken Seite stellen die Daten mit unterschiedlicher Graustufenskalierung dar. Rechts werden die Temperaturwerte durch die pseudo Farben visualisiert. Die verwendete Farbpalette geht von blau nach rot, wobei je wärmer desto mehr rot-Anteil. Die Werte sind auf die Standardwerte für Innenräume skaliert, somit ist der Wertebereich verkleinert, womit eine feinere Granularität ermöglicht wird.

Die nächsten zwei Bilder A.5 visualisieren die mögliche Vorgehensweise für die Kalibrierung und Registrierung der Kamera innerhalb des Gesamtsystems. Bei dem rechten Bild sind zwei LED-Sensoren zu sehen. Ist die Anordnung der Sensoren im Raum bekannt und das Kalibrierungsmuster eindeutig kann ein Registrierungsalgorithmus sowie eine Kalibrierungsprozedur entwickelt werden. Das linke Bild visualisiert den Kalibrierungskörper, ein Schachbrettmuster, das durch eine starke Leuchtquelle bestrahlt wurde. Der Körper wird normalerweise für die Registrierung der PR2-Sensoren zueinander verwendet. Durch die Bestrahlung empfängt die Wärmebildkamera ein invertiertes Schachbrettmuster, da die schwarzen Quadrate mehr Energie absorbieren und dadurch heller wahrgenommen werden als die Weißen. Würde das Schachbrettmuster durch eine passende Leuchtquelle während der Kalibrierung bestrahlt, wäre sogar eine Integration in den für den PR2 bereits bestehende Kalibrierungsverfahren denkbar. Ist die Wärmebildkamera mit den anderen Sensoren registriert, können alle oben genannten Szenarien realisiert werden.

Anhang B

Kognitive Aspekte der Objekterkennung

Dieses Kapitel beschäftigt sich mit den kognitiven Aspekten der Objekterkennung. Dabei soll anhand der Objekterkennung bei Menschen die wichtigsten Aspekte dieser herausgearbeitet werden. Im Weiteren wird die mögliche Übertragung dieser Aspekte auf eine Roboterplattform diskutiert. Abschließend wird beleuchtet wie diese Ansätze in vorliegender Arbeit genutzt werden.

B.1. Objekte, Kategorien und Klassen

Der Begriff „Objekt“ beschäftigt seit mehreren Jahrhunderten viele Wissenschaftler, die ihn immer wieder neu definieren oder bestehende Definitionen ergänzen. Daher ist es nicht verwunderlich, dass viele von einander unterschiedliche Definition dieses Begriffs existieren. Vor allem in der Philosophie ist der Begriff „Objekt“ (lat. obiectum, das Entgegengeworfene) ein oft verwendete und uneinheitlich definierte Bezeichnung. Oft wird das Wort als Synonym für „Gegenstand“ verwendet und damit als eine grundlegende ontologische Kategorie mit Eigenschaft und Ereignis angesehen. Seit dem “Descartes Dualismus” wird zusätzlich zwischen Subjekt und Objekt unterschieden. Eine der wichtigsten Kriterien ist dabei die Unterscheidung zwischen der passiven und aktiven Wahrnehmung. Das Subjekt nimmt aktiv das Objekt wahr, das passiv in der Umgebung gegeben ist.

Im Laufe der Geschichte wird dem Begriff „Objekt“ eine immer tiefere Bedeutung zugesprochen. Während im 13. Jahrhundert Thomas von Aquin nur die materiellen Eigenschaften des Objekts betrachtete [Sto93], erweiterte im 18. Jahrhundert Immanuel Kant den Begriff um die Kontextinformation [Kan81]. So sind Objekte bei Kant Erscheinungen, denen eine empirische Realität zukommt. Sind aber transzendental ideal und damit vom Ding an sich zu trennen: „Objekt aber ist das, in dessen Begriff das Mannigfaltige einer gegebenen Anschauung vereinigt ist.“

Eine vergleichbare Entwicklung durchlebt der Begriff in der Informatik. Spätestens mit der Entstehung der objektorientierten Programmiersprachen ist ein Objekt ein abstraktes Element unserer Vorstellung. Je nach Anwendung wird das Wort unterschiedlich interpretiert.

Unbestreitbar bleibt, dass ein Objekt nicht nur über materielle Eigenschaften verfügt. Diese sind nur eine Erleichterung um die Kontextinformationen und Beziehungen richtig einordnen zu können und damit das Objekt schneller zu kategorisieren und zu erkennen.

Wie soll nun etwas erkannt werden, was nicht einmal eine eindeutige Definition besitzt? Eine weitere interessante Frage ist, kann ein Mensch ein Objekt erkennen, was er davor noch nie gesehen hat? Mit anderen Worten, basiert die menschliche Erkennung auf der Erinnerung an ein Objekt oder der Funktionalität desselben? Im Weiteren sollen diese Fragen auf die Objekterkennung angewandt und durch einen Roboter beantwortet werden. So viel sei aber an dieser Stelle gesagt, nach Meinung des Autors sind beide Komponente essentiell. Eine parallel zum Boden ausgerichtete Fläche bietet die Möglichkeit ein oder mehrere Objekte darauf zu platzieren und ist damit funktional gesehen ein Tisch. Wiederum kann ein aus einer kolorierten 3-D-Punktwolke segmentiertes Cluster anhand bekannter Merkmale wie zum Beispiel Größe, Form, Textur und/oder Farbe mit einer Datenbank verglichen und dadurch erkannt werden.

In der Robotik wird das „Erkennen“ als die Zuordnung zu einer Kategorie oder einer Klasse verstanden. Der Begriff der Kategorie hat sich genauso im Laufe der Zeit in seiner Bedeutung verändert wie der Begriff Objekt. Aristoteles gilt als der Urheber einer der ersten Kategorienlehre. Durch die Analyse des Sprechens über die Dinge entwickelte er zehn Kategorien (categorien, gr. Aussage), die die ontologischen Grundprinzipien auf einer allgemeinen Ebene beschreiben. Dabei spielen die sogenannten Akzidentien, wie Qualität, Quantität, Relation, Ort, Zeit, Zustand, Lage, Wirken (aktiv) und Leiden (passiv), eine entscheidende Rolle. Die Kategorien von Aristoteles bestimmten grundlegend alle philosophischen Arbeit bis ins 20. Jahrhundert.

Erst Ludwig Wittgenstein [Wit03] stellt mit seiner Überzeugung, dass die Grenzen der Sprache auch die Grenzen der beschreibbaren Welt darstellen, zum ersten mal die Kategorisierung von Aristoteles in Frage. Mit seinem Vorgehen, die „Bedeutung des Wortes aus dessen Gebrauch in der Sprache“ zu definieren, gilt er als der Begründer moderner kognitiver Kategorisierungstheorien. In seinem Verständnis sind Kategorien keine grundlegenden Prinzipien des Denkens, sondern werden synonym zu „Klasse“ verwendet. Somit ist „kategorisieren“ ein klassifizierender Vorgang, der ein Wahrnehmungsobjekt in einen Bedeutungszusammenhang einordnet.

Die kognitiven Kategorisierungstheorien wurden entscheidend durch die Arbeiten von der amerikanischen Psychologin Eleanor Rosch [RM75][RMGJ75] geprägt. Sie schaffte eine konsequente Methodik, die eine Reihe von überraschenden Effekten offen legte. Der Ausgangspunkt ihrer Kritik an den „klassischen Kategorisierungen“ bilden zwei Argumente. Beide betreffen die gemeinsamen Eigenschaften einer Klasse, die durch ihre Mitglieder bestimmt sind. Wenn das so ist, wie können dann typische Exemplare einer

Klasse im Vergleich zu den anderen existieren? Wird dabei nicht das kategorisierende Subjekt selbst außer Acht gelassen? Spielt dabei die menschliche Neurophysiologie sowie Fähigkeiten zur Wahrnehmung, zur Bildung mentaler Bilder, zum Lernen, Erinnern und Organisieren des Gelernten sowie zur Kommunikation eine wichtige Rolle?

Die Arbeiten von Rosch bilden eine allgemeine Perspektive auf ein Problem, das nicht nur Psychologen interessierte. Ihre Theorien sind unter dem Namen „Prototypen-Theorie“ zusammengefasst und basieren vor allem auf zwei Themengebieten, die unter dem Namen „”basic-levelËffekt“ und den „Prototypen Effekt“ bekannt geworden sind.

Der ”basic-levelËffekt ist zum ersten mal durch Roger Brown beschrieben worden [Bro58]. Er hatte beobachtetet, dass es eine „erste Ebene“ existiert in der Kinder lernen die Objekte in ihrer Umwelt zu benennen. Diese erste Ebene ist weder allgemein noch speziell. Die „erste Ebene“ wird durch die charakteristischen Merkmale sowie kürzere und häufiger benutzte Namen spezifiziert. Brown nannte diese Kategorisierungsebene „die natürliche“, später wurde sie als der „”basic-level’ beziehungsweise „entry-level“ genannt. Typische Beispiele sind:

Wirbeltier - Säugetier - Hund - Dackel
Möbel - Tisch - Gartentisch
Pflanze - Baum - Buche

Die Arbeiten von Rosch stellen die mittlere Ebene der taxonomischen Hierarchie aus psychologischer Sicht als die Wichtigste dar. Damit widerspricht diese der traditionellen Theorie, die alle Klassen als gleichwertig behandelt. Die mittlere Ebene (der ”basic-level”) ist nach Rosch [RMGJ75] wie folgt definiert:

- Die höchste Ebene, auf der die Kategorienmitglieder eine ähnlich wahrgenommene Gesamtgestalt haben,
- Die höchste Ebene, auf der eine einzige Vorstellung die Kategorie voll repräsentieren kann,
- Die höchste Ebene, auf der mit ähnlichen Handlungen und allen Kategorienmitgliedern interagiert werden kann,
- Die Ebene, auf der die Kategorie am schnellsten klassifiziert wird,
- Die Ebene mit den gebräuchlichsten Bezeichnungen für die Kategorienmitglieder,
- Die Ebene, die zuerst von Kindern verstanden und gebraucht wird,
- Die Ebene, die ohne weitere Erklärung ohne Kontext gebraucht werden kann,
- Die Ebene, auf der das meiste Wissen vorhanden ist.

Damit übernimmt die mittlere Ebene der Taxonomie solche grundlegende Tätigkeiten wie die Gestaltwahrnehmung, die Wissensrepräsentation, die Vorstellungsfähigkeit, die Merkfähigkeit sowie die Kommunikation und die Bewegungsinteraktion. Abhängig von der Wahrnehmung und gemachter Erfahrung wachsen die Strukturen von Basic-Level-Kategorien; sie existieren also nie abstrakt.

Rosch ist es gelungen experimentell nachzuweisen, dass Klassen in der Regel typische, „beste“ Exemplare besitzen. Methodisch wurde es auf die folgende Weise erreicht:

- Die Probanden bewerten einige Exemplare einer Klasse danach, wie gut diese als Beispiel die Klasse repräsentieren.
- Die Reaktionszeit für die Beantwortung der Frage, ob ein bestimmtes Beispiel einer Klasse angehöre, wurde gemessen.
- Anhand der Auswertung konnten so die „typischen“ Exemplare einer Klasse bestimmt werden.
- Asymmetrie nach Ähnlichkeit: Die Ähnlichkeit der weniger typischen Beispielen zu typischen innerhalb einer Klasse wurde als größer angesehen als die Ähnlichkeit von typischen zu weniger typischen.
- Asymmetrie in der Generalisierung: Neue Aussagen über ein Exemplar einer Klasse wurden eher vom weniger typischen zum typischen hin übertragen als umgekehrt.
- Es konnte empirisch gezeigt werden, dass die „Familienähnlichkeit“ nach Wittgenstein nicht nur eine philosophische a-priori Spekulation war: Die wahrgenommene „Familienähnlichkeit“ zwischen Exemplaren einer Klassen korrelierte mit der numerischen Häufigkeit, wie „beste“ Beispiele dieser Klasse eingeordnet wurden.

Ludwig Wittgenstein zeigte in seinen „Philosophischen Untersuchungen“ anhand von Beispielen der Begriffe Sprache, Spiel und Sprachspiel, dass bestimmte Kategorien von Dingen mit einer taxonomischen Klassifikation (hierarchischen Systematik) nicht hinreichend erfasst werden können [Wit03]. Nach Wittgenstein gibt keine allgemeinen Merkmale, die für alle Sprachen, Spiele und Sprachspiele gleichermaßen gelten. Es gibt zwar einige Spiele mit den gemeinsamen Merkmalen, diese weisen aber wiederum überhaupt keine Gemeinsamkeiten mit den anderen. Diese Form von lockerer Gemeinsamkeit bekam den Namen Familienähnlichkeit (family resemblance, cluster definition). Die Mitglieder einer Klasse sind wie Mitglieder einer Familie, deren Ähnlichkeit sich durch den statistisch verteilten Anteil an einer Gesamtheit von Merkmalen bestimmt.

Allerdings könnten das Prototypenkonzept und die Familienähnlichkeit nach Wittgenstein nicht vollständig vereint werden. So gibt es unter den Mitgliedern einer Familie kein „bestes Beispiel“, keine typischen und weniger typischen Exemplare. Ein Prototyp einer Klasse wäre darin ein Spezialfall.

Durch Roschs Arbeiten konnte die Existenz von Prototypen auch experimentell belegt werden. Barsalou [Bar83] ging sogar einen Schritt weiter und zeigte, dass die Prototypen-Effekte nicht nur für wichtige Dinge unserer Kultur gelten, sondern auch für die von ihm so benannten „ad-hoc Kategorien“, wie zum Beispiel „was man zum Geburtstag schenkt“ oder „was man aus dem Haus rettet, wenn es brennt“. Dieses Phänomen ist umso erstaunlicher, als diese spontan entworfenen Kategorien, für die es keine explizite Bezeichnung gibt, eigentlich keinerlei Präexistenz haben sollten.

Zuerst interpretierte Rosch, indem sie von den Annahmen der informationsverarbeitenden Psychologie ausging [Lak87], dass die Prototypen auf der Wahrnehmungsebene durch signifikante Merkmale, gute Erinnerbarkeit und Verallgemeinerungstauglichkeit der Exemplare entstehen. Später nahm sie aber von einer eins-zu-eins Entsprechung von Prototyp und Repräsentation wieder Abstand. Rosch akzeptierte, dass Prototypen zwar die Gedächtnisstrukturen bestimmen, sind aber nicht hinreichend genug, um die Repräsentation abzugrenzen. Sie nahm darüber hinaus an, dass hinter dem Prototyp eine nicht näher bestimmbare Quelle liegt, die es ermöglicht, dass bei den Kategorien mit klarer Begrenzung auch vom Prototyp stark abweichende Exemplare (beispielsweise ein Pinguin in der Kategorie „Vogel“) noch zweifelsfrei als innerhalb der Kategorie zugehörig bestimmt werden können.

Die Prototypen-Effekte waren so überzeugend, dass die Prototypen-Theorie in den kognitiven Theorien zu Repräsentation, Gedächtnis und Lernen den Einzug fand, obwohl sie bisher durch kein Modell hinreichend belegt werden konnten.

Abschließend soll noch auf die Generalisierung eingegangen werden. Generalisierung bedeutet die Fähigkeit, Wissen, das von bestimmten Exemplaren einer Kategorie erlernt ist, auf andere ähnliche Exemplare anzuwenden [Squ92]. Für ein erfolgreiches Verhalten ist die Generalisierung von extremer Wichtigkeit. Aber auch die daraus resultierende Diskriminierung ist für ein erfolgreiches Verhalten notwendig, denn Exemplare, die zwar ähnlich sind, aber nicht zur gesuchten Kategorie gehören, sollen von solchen unterschieden werden, die zur Kategorie gehören, auch wenn jene unähnlicher sind als die anderen. Die Eigenschaft der Kategorie muss also für eine erfolgreiche Generalisierung als invariant erkannt werden. Die Fähigkeit der Generalisierung gilt deshalb als ein Beweis für die Existenz einer exakten mentalen Repräsentation und somit einer erfolgreichen Objekterkennung.

B.2. Visuelle Wahrnehmung

Die visuelle Wahrnehmung des Menschen ist ein stark erforschter Bereich der allgemeinen Psychologie. Die Verarbeitung von visuellen Reizen und die Entstehung einer vollständigen Repräsentation eines Objekts werfen immer noch viele Fragen auf. Die Fortschritte lassen aber hoffen, dass in absehbaren Zukunft viele neue Erkenntnisse folgen werden.

Durch Johannes Kepler und René Descartes wurde schon früh erkannt, dass das Sehen durch die Bilder entsteht, die sich auf der konkaven Oberfläche des Auges abbilden. Damit fungiert das Auge als eine Art Kamera, die 2-D-Bilder der 3-D-Umgebung wahrnimmt. Die Tiefeninformation geht bei dieser Abbildung verloren. John Locke postulierte, dass die Wahrnehmung durch die Kombination elementarer Empfindungen entsteht. Max Wertheimer sah die Wahrnehmung als Ergebnis von „Feldenergien“ im Gehirn. Dabei stellte er fest, dass das Ergebnis sich aus mehreren Teilen zusammensetzt und gemeinsam mehr Informationen beinhaltet als ihre reine Summe. Des Weiteren steht definitiv fest, dass die visuelle Wahrnehmung in mehreren Schritten unter Verwendung unterschiedlicher Strategien abläuft.

Bei Menschen beginnt die Verarbeitung der visuellen Reize bereits in den Zellschichten der Retina. Zum Beispiel die Bipolarzellen in den Mittelschichten, die für die Verarbeitung essentielle Informationen in so genannte rezeptive Felder (RF) zusammenfassen, die jeweils aus einem Zentrum und einer Randzone bestehen. Die On-Bipolarzellen werden durch eine Licht-Reizung ihres Zentrums erregt und durch eine Licht-Reizung der Randzone gehemmt, die Off-Bipolarzellen verhalten sich genau umgekehrt.

Die Retina ist als in sich gegenseitig überlappende rezeptive Felder strukturiert. Damit werden schon auf dieser Ebene nicht homogene, monochrome Flächen detektiert, sondern Grenzen, Kanten und Kontraste. Durch konvergente Signalweiterleitung und laterale Inhibition wird die Kontrastinformation zusätzlich verstärkt.

Nach Schiller, Logothetis und Charles [SLC90] wird das visuelle Eingangssignal schon auf der Ebene des Corpus genicalatum laterale (CGL) aufgetrennt. Dabei wird die Retinotopie beibehalten; das bedeutet, dass Impulse aus benachbarten rezeptiven Feldern der Retina auch im CGL benachbart ankommen, damit wird die Nachbarschaftsbeziehung einbehalten. Die Auftrennung erfolgt nach visuellen Merkmalen:

- *Nach Bewegung:* M-Ganglienzellen, Schichten 1 und 2;
- *Nach Farbe, Textur, Muster und Tiefeninformationen:* P-Ganglienzellen, Schichten drei bis sechs.

In der primären Sehrinde werden danach die visuellen Impulse nach Merkmalen des Ortes, der Länge, der Orientierung, der Ortsfrequenz und der Bewegung analysiert. Durch die Arbeiten von Hubel und Wiesel [Hub95] konnten bahnbrechende Erkenntnisse über den visuellen Kortex des Säugetiers gewonnen werden. Es gelang ihnen, mit Mikroelektroden aus dem Katzengehirn einzelne Neuronenaktivitäten abzuleiten, während sie das Tier auf verschiedene Muster schauen ließen. Sie entdeckten dabei drei verschiedene Arten von Zelltypen im primären visuellen Kortex:

- Einfache kortikale Zellen (simple cells) reagieren am stärksten auf Linien an einem bestimmten Ort und in einer bestimmten Orientierung (senkrecht, waagerecht oder diagonal).

- Komplexe Zellen (complex cells) antworten am stärksten auf Linien mit einer bestimmten Bewegungsrichtung. Sie empfangen Eingangssignale von mehreren über die Retina verteilten kortikalen Zellen und reagieren, wenn diese Signale einen bestimmten zeitlichen Abstand aufweisen.
- „Endinhibierte Zellen“ (endstopped oder hypercomplex cells) antworten am stärksten auf Linien einer bestimmten Länge und Bewegungsrichtung. Hier liegt eine höhergradige Verschaltung aus einfachen und komplexen Zellen zugrunde, die diese Raum-Zeitliche Reizverteilung detektiert.

Besonderes interessant aus der Sicht der computergestützten Bildverarbeitung ist, dass Hubel und Wiesel die Neuronen im visuellen Cortex als Merkmalsdetektoren bezeichneten. Durch den Aufbau und die Verteilung von erregenden und hemmenden Bereichen im rezeptiven Feld der Zellen wird das orientierungsselektive Antwortverhalten dieser Zellen ermöglicht. Daraus resultiert ein Konzept der Äquivalenz des Neurons mit dem Stimulus, das seither als die Grundlage zur Beschreibung der neuronalen Repräsentation eines wahrgenommenen Gegenstandes dient.

Visueller Cortex ist wie das CGL retinotop organisiert. Den rezeptiven Feldern aus der Fovea wird allerdings in V1 etwa fünfmal so viel Raum eingeräumt wie denjenigen aus der Retinaperipherie. Hubel und Wiesel beschreiben in ihrer Arbeit auch die vertikale Organisation der primären Sehrinde in Säulen (columns), und zwar nach drei unterschiedlichen Reizeigenschaften: Zellen des gleichen Ortes auf der Retina (Positionssäulen), Zellen gleicher Merkmalsdetektion (Orientierungssäulen) und Zellen, die jeweils optimal auf eins der beiden Augen ansprechen (Augendominanzsäulen). Diese Säulen sind alle etwa 1 mm dick. Ein Kubikmillimeter-Block des Kortex bildet so, indem er jeweils eine Säule jeder Sorte zusammenfasst, ein Verarbeitungsmodul für Position und Orientierung. Eine solche Hyperkolumne bedient jeweils einen kleinen Teil der linken oder der rechten Netzhaut.

Auch der Tastsinn spielt bei der Wahrnehmung eine große Rolle. Nach Loomis und Lederman [LL86] wird der Tastsinn in drei Bereiche unterteilt, den kutanen, den kinästhetischen und den haptischen Sinn. Der kutane oder taktile Sinn vermittelt über die Hautrezeptoren Qualitäten wie Oberflächenbeschaffenheiten und Temperatur. Der kinästhetische Sinn liefert über die Rezeptoren in den Muskeln, Sehnen und Gelenken Information über die Stellung und Bewegung der eigenen Gliedmaßen. Der haptische Sinn benutzt beide Bereiche, die Hautrezeptoren und die kinästhetischen Rezeptoren, zur Erkundung von Objekten. Eine motorische Komponente ist dabei unentbehrlich, wie Lederman und Klatzky [LK98] in ihrer grundlegenden Forschungen zusammenfassen. Das Ertasten präziser Informationen und die Identifikation komplexer Objekte erfordern gewollte, explorative Bewegungen, die ihrerseits durch die Wahrnehmungsziele des Beobachters gelenkt werden. Dabei wird Handlung im Dienst von Wahrnehmung ausgeführt, abhängig von der Verarbeitung der kutanen und kinästhetischen Reize durch das haptische System.

B.3. Objekterkennung des Menschen

Werden die davor präsentierte Kenntnisse zusammengefasst, entsteht ein ungefähres Bild wie die Objekte durch die Menschen erkannt werden. Nach den neuesten Kenntnissen entsteht dabei eine mehrstufige Hierarchie. In der früheren Phase werden die visuelle Merkmale, wie Figuren, Form, Farbe, Größe, Textur, Bewegung, etc., extrahiert. Die Verarbeitung findet dabei nach dem bottom-up Prinzip statt. Die Verarbeitung in der späteren Phase läuft top-down und beinhaltet Mustererkennung, komplexe Figuren sowie die Verknüpfung mit semantischen Aspekten. Dabei beansprucht die visuelle Wahrnehmung eine enorme Gehirnleistung [GIM02]. Etwa die Hälfte eines Primatengehirns ist mit Sehen beschäftigt. Die Masse an anfallenden Informationen kann am besten am Beispiel der Sehbahn geschätzt werden. Die Sehbahn ist fast komplett kreuzweise über die Netzhaut verteilt und führt über Chisma Opticum und Corpus geniculate laterale zum prämeren visuellen Kortex [Gol07].

Nach dem die beiden Schritte abgeschlossen sind, müssen die extrahierten Merkmale zu Objekten zugeordnet werden. Die meisten Forscher auf diesem Gebiet sind davon überzeugt, dass dies unter Zuhilfenahme von Gestaltgesetzen [BGG03] passiert. Diese Gesetze funktionieren nach dem Prinzip, eine möglichst präzise Lösung zu finden und werden im Folgenden aufgelistet:

- Prägnanz,
- Nähe,
- Ähnlichkeit,
- Symmetrie,
- Gemeinsame Bewegung,
- Kontinuität,
- Geschlossenheit,
- Gemeinsame Region,
- Verbundene Elemente.

Diese oben genannten Gestaltprinzipien wirken, weil das visuelle System die Umwelt so konstruiert, wie sie am wahrscheinlichsten ist. Daher können folgende sinnvolle Annahmen über die Welt getroffen werden:

- Gleiche Oberflächen absorbieren das Licht ähnlich,
- Zusammengehöriges sieht ähnlich aus,

- Kohäsion; Benachbartes gehört zusammen,
- natürliche Bewegungen sind meist kontinuierlich,
- natürliche Objekte sind meist symmetrisch.

Basierend auf den oben vorgestellten Erkenntnissen präsentiert David Marr [Mar82] seine erweiterte Theorie der Wahrnehmung. Dabei zerlegt er die Wahrnehmung in verschiedene Phasen und beschreibt die Wahrnehmungs- sowie Kognitivenprozesse auf unterschiedlichen Ebenen. Die Theorie von Marr beinhaltet drei unterschiedliche Stufen, deren Komplexität absteigend wächst.

- Primärskizze (primal sketch): Kanten und Konturen werden identifiziert,
  - Raw primal sketch: Identifikation von Lichtintensitätsunterschieden,
    - Eckensegmente, Balken, Terminierungen, Klumpen („blobs“),
  - Full primal sketch: Identifikation der Anzahl von Objekten und Formen,
    - Principle of explizit naming,
    - Principle of least commitment,
- $2\frac{1}{2} - D$ Skizze: Orientierung und Tiefe des Gegenstandes, Oberflächenstruktur und Bewegung werden identifiziert (blickwinkelabhängig),
- $3 - D$ Skizze: Generierung einer Objektbeschreibung, Modells des Gegenstandes (blickwinkelabhängig).

Die Abbildung B.1 visualisiert die davor gehende Auflistung und ergänzt diese durch die Richtung des Informationsflusses.

Nach Marr und Nishihara [MN78] bestehen alle Objekte in der menschlichen Wahrnehmung aus Zylindern. Biederman [Bie87] erweitert diese Vorstellung und führt den Begriff “Geon” (geometrec icons) ein. Dabei sind die Geons 36 elementare 3-D-Formen, aus denen tausende von Objekten zusammengebaut werden. Sie sind leicht unterscheidbar und ermöglichen eine blickwinkelunabhängige Erkennung. Abbildung B.2 stellt einige Beispiele für die Geons und deren Zusammensetzung zur einem Objekt dar, das Bild wurde aus [Bie87] übernommen.

Die Konturen der Geons werden jeweils aus Schlüsselmerkmalen im Bild bestimmt, den so genannten „nonaccidental properties“ (NAPs). Das sind allgemeine Gesetze des Transfers einer zweidimensionalen Darstellung auf ein dreidimensionales Volumen: „collinearity“ (eine Gerade bleibt eine Gerade), „curvilinearity“ (eine Kurve bleibt eine Kurve), „symmetry“ (symmetrisches bleibt symmetrisch), „parallel edges“ (parallele Kanten bleiben parallel) und „coterminating edges“ (zugleich endende Kanten bleiben bestehen). Diese NAPs sind dabei weitgehend invariant gegenüber Blickrichtungsveränderungen. So

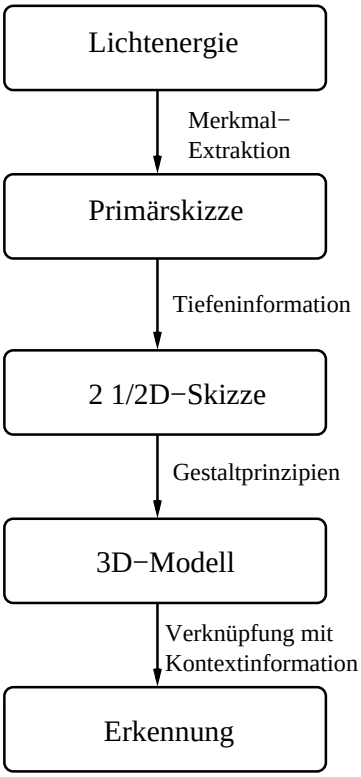


Abbildung B.1.: Zusammenfassung der Theorie der Wahrnehmung nach Marr.

können die 2-D-Konturen im Bild direkt in eine 3-D-Geon-Beschreibung übernommen werden.

Diese Theorie bleibt bis heute umstritten. So ist die Frage, wie die Geons selbst gelernt werden, immer noch unbeantwortet. Auch wie die konstituierenden Geons eines unbekannten Objekts gefunden werden ist unklar. Daraus ergibt sich, dass das Konzept nur für bereits gelernte Objekte effizient funktioniert. Außerdem fehlt bislang ein Existenznachweis für Geons überhaupt. Auch ihre mit 36 angegebene Anzahl ist nicht plausibel.

Das die räumliche Wahrnehmung bei der Objekterkennung eine wesentliche Rolle spielt ist unumstritten. Die Rekonstruktion der Tiefeninformation blieb in diesem Kapitel bislang unbeantwortet. Die Informationen über die Umgebung, die ein Mensch über seine Augen bekommt ist zweidimensional. Es dauerte mehrere Jahrhunderte, um die Leistung des Gehirns bei der Gewinnung der Tiefeninformation grob nachzuvollziehen. Die allgemeingültige Vorstellung ist mittlerweile, dass aus der Zusammenlegung der Ergebnisse mehrerer unterschiedlicher Verfahren die Tiefe geschätzt werden kann.

Dabei werden für unterschiedliche Entfernungen verschiedene Tiefenkriterien verwendet.

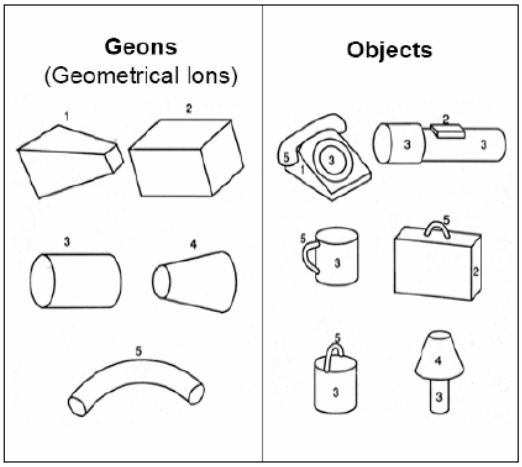


Abbildung B.2.: Beispiele für einige Geons und Objekte, die aus ihnen zusammengesetzt werden können. Durch relational unterschiedliche Konstruktion können verschiedene Objekte entstehen (aus Biederman, 1987).

Zum Beispiel ist die Disparität besonderes für die Auge-Hand-Koordination wichtig und daher im Nahbereich enorm relevant. Für die Objekte in weiterer Entfernung sind die monokularen Kriterien von Bedeutung. Die Visualisierung der unterschiedlichen Tiefenhinweise ist in der Abbildung B.3 wiedergegeben. Abbildung B.4 stellt die Gewichtung verschiedener Tiefenkriterien in Relation zu der Entfernung dar.

Mathematisch gesehen kann eine zweidimensionale Form als die Projektion einer unendlichen Mannigfaltigkeit verschiedener möglicher 3-D-Formen interpretiert werden. Aus diesem Grund besitzt der inverse Prozess der Ableitung einer dreidimensionalen Objektstruktur aus einem zweidimensionalen Netzhautbild während des Sehvorgangs, das so genannte „inverse Problem“ [Piz01], für sich genommen keine eindeutige Lösung. Zur Rekonstruktion der dritten Dimension ist daher immer Zusatzinformationen erforderlich.

Müsseler und Prinz untersuchten die Abhängigkeit zwischen Handlung und Wahrnehmung und stellen in ihrem Buch „Allgemeine Psychologie“ [MP02] folgende Hypothese auf: „Das, was man tut hat einen Einfluss auf das, was man sieht“. Somit beeinflusst nicht nur die Wahrnehmung die Motorik, sondern auch die Motorik die Wahrnehmung. Dieses wird auch durch die Arbeit von Pick und Salzmann [PS78] bestätigt, die die Wahrnehmung in Abhängigkeit von der aktuellen Aufgabe untersuchten.

Damit so eine Leistung vom Gehirn in Echtzeit erbracht werden kann, werden Informationen über das wahrgenommene Bild im Gehirn mehrfach repräsentiert und in unterschiedlichen Gehirnarealen verarbeitet. Die Verarbeitungspfade laufen häufig parallel, konvergieren aber nie in einem Punkt. Erst durch die synchronisierte Entladung

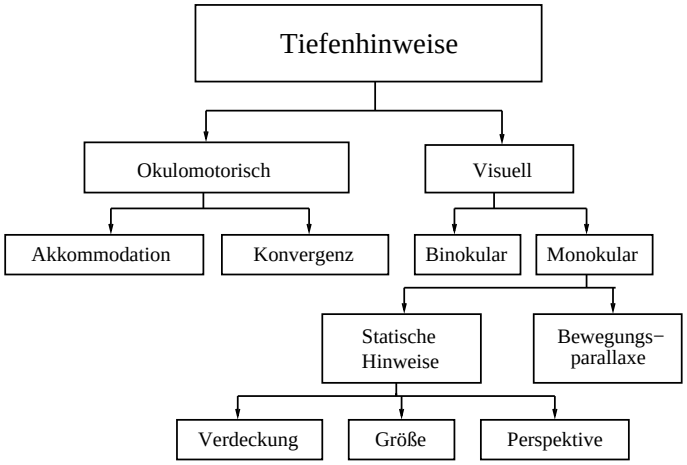


Abbildung B.3.: Verschiedene Tiefenhinweise, die für menschliche Tiefenwahrnehmung verantwortlich sind.

der Nervenzellen, die an der Verarbeitung verschiedener Eigenschaften eines und desselben Objekts beteiligt sind, wird die Wahrnehmung eines Objekts als eine geschlossene Einheit ermöglicht [NBAK05].

Das die visuelle Wahrnehmung fehleranfällig ist, wird durch optische Täuschungen auf die beeindruckende Weise bestätigt [ML89]. Dabei betreffen die optische Täuschungen nahezu alle Aspekte der visuellen Wahrnehmung. Es existieren mehrere unterschiedliche Illusionen, die durch Tiefe, Farbe, geometrische Anordnung oder Bewegung verursacht werden.

Abschließend bleibt nur festzustellen, dass es viele Theorien der Objekt- und Umgebungswahrnehmung existieren. Keine von diesen Theorien ist aber hinreichend, um die menschliche Wahrnehmung eindeutig und komplett zu beschreiben. Im Weiteren gibt es ein Problem, das alle genannten Theorien der Objekterkennung betrifft, insbesondere die Theorie von Biederman, die betont, dass wir Objekte weitgehend unabhängig von ihrer Größe, Orientierung, Farbe und Oberflächenbeschaffenheit erkennen können. Hinsichtlich Farbe und Oberflächenbeschaffenheit argumentiert Biederman, dass die Benennungszeiten für Objekte, die auf Strichzeichnungen (ohne Farbe und Textur) dargestellt sind, sich nicht signifikant von den Zeiten bei Fotografien unterscheiden. Aber Größe und Orientierung sind einflussreich, den in den Experimenten von Biederman und Cooper wurden Strichzeichnungen von gleichen oder verschiedenen Objekten schnell nacheinander gezeigt, und zwar teils in gleicher, teils in unterschiedlicher Größe. Es stellte sich heraus, dass die durchschnittliche Benennungszeit weitgehend unabhängig von der Größe der gezeigten Objekte war. Jedoch wurden die Reaktionszeiten und Fehlerraten für die Vergleichsurteile zwischen erstem und zweiten Bild systematisch länger, wenn dabei die

Depth information	0–2 meters	2–30 meters	above 30 meters
Occlusion	✓	✓	✓
Relative size	✓	✓	✓
Accommodation and convergence	✓		
Motion	✓	✓	
Disparity	✓	✓	
Height		✓	✓
Atmospheric perspective			✓

Abbildung B.4.: Gewichtung verschiedener Tiefenkriterien in Relation zur Entfernung.

Objektgröße variierte. Die Autoren sind der Meinung, dass beim Identifizieren (Benennung einer Objektklasse) und beim Wiedererkennen (Vergleich zwischen eben gesehenen Objekten) zwei verschiedene Bereiche des Gedächtnisses wirksam sind, und nur das Identifizieren sei weitgehend von der Objektgröße unabhängig.

Hinsichtlich der Orientierungen der Objekte ist aus verschiedenen Untersuchungen bekannt, dass ein Wiedererkennen um so schneller gelingt, je eher die Orientierung des gesehenen Objekts einer „kanonischen“ Standard-Ansicht entspricht, die meistens eine Ansicht „schräg von vorn“ bedeutet. Wenn Probanden ihre Vorstellungen von einem Gegenstand beschreiben sollen, dann berichten sie meist von einer „Standard“-Ansicht. Wenn sie entscheiden sollen, welches Foto sie als „beste“ Darstellung eines Objekts wählen würden, dann wurde überwiegend das Foto aus einer dieser „Standard“-Ansicht ausgesucht. Die kanonische Perspektive enthält fast immer Information über die 3-D-Verhältnisse zwischen Front-, Ober- und Längsseite. Plausibel ist, dass Identifikation und Wiedererkennen mehr Zeit brauchen und mit mehr Fehlern behaftet sind, wenn Objekte nicht aus der „kanonischen“ Perspektive gesehen werden.

B.4. Objekterkennung durch eine mobile Roboterplattform

Zuerst soll in diesem Abschnitt versucht werden die Aufgaben und die Funktionsweise der Wahrnehmung zusammenzufassen:

- Wahrnehmung dient der Erfüllung wichtiger Funktionen wie Navigation, Balance, sozialer Interaktion und Objektwahrnehmung;
- Um die Komplexität der Objekterkennung zu reduzieren werden des Öfteren die Muster verwendet, die eine reduzierte Repräsentation der Objekte darstellen;

- Nach Marr besteht jedes wahrgenommene Objekt aus elementaren zusammengesetzten geometrischen Primitiven (Geons);
- Objekterkennung in den Bilddaten kann mit und ohne eines Modells erfolgen. Für komplexe Strukturen ist es vorteilhaft kein Modell erstellen zu müssen. In solchen Fällen wird meist die Farbe und Reflektanz untersucht. Ein Modell erfordert dagegen Verfahren, die auf Kanten-, Flächen- und Eckenextraktion basieren.

Das biologisch relevante Wahrnehmen funktioniert nie in einer ausschließlich visuellen Welt; beim Sehvorgang wird eine Vielzahl von außerhalb der Netzhaut liegenden, zusätzlichen Faktoren mitintegriert. Aus eigenen Bewegungen, vorhandenem Wissen und präsenter Erfahrung, gelenkt durch unsere Aufmerksamkeit fließen die Zusatzinformationen mit ein. Alle diese Faktoren bilden den Kontext des Sehvorgangs und beeinflussen über das Wirkungsgefüge der visuellen Wahrnehmung das Erleben der Umwelt. Ohne den Kontext mit einzubeziehen ist visuelles Erkennen nicht zu verstehen. Jegliches Objektwissen, dass außerhalb des Objekts zur Verfügung steht, das zu einem anderen Zeitpunkt oder an einem anderen Ort gewonnen wurde, zählt zum „Kontextwissen“ [Alb95]. Bei der experimentellen Untersuchung der Objekterkennung fallen darunter im Wesentlichen drei verschiedene Aspekte:

- Das Vorwissen über die Objekte;
- Die relative Konfiguration der Objekte im gegebenen Szenario;
- Die Objektumgebung.

Dennoch bleibt die Frage offen, was tatsächlich in Bezug auf ein mobiles Robotersystem angewandt werden kann. Zuerst bietet eine mobile Plattform die Möglichkeit zur aktiven Wahrnehmung. Ein Roboter kann sich im Raum lokalisieren und bewegen. Durch die Bewegungen des Roboters und der Sensoren, die beispielsweise auf einer Schwenk-Neige Einheit montiert sind, besteht die Möglichkeit die Perspektive zu ändern. Werden die Daten, die aus der ersten Wahrnehmung stammen, gespeichert kann das Wahrgenommene ergänzt werden. Damit kann eine bestmögliche Objektorientierung gefunden oder ein komplettes 3-D-Objektmodell erschaffen werden. Durch die Verwendung unterschiedlicher Sensoren wird ermöglicht, verschiedene Merkmale zu detektieren und gemeinsam auszuwerten. Auch die Manipulatoren können genutzt werden, zum Beispiel um die Orientierung des Objekts oder sogar die Anordnung innerhalb der Szene zu verändern. So wird ein Roboter aktiv in die Wahrnehmung integriert. Damit wird ein Roboter zur einer aktiven Komponente innerhalb der Wahrnehmung, der abhängig von den Sensordaten auf die Umgebung reagiert und mit ihr interagiert.

Hier werden schon einige Vorteile gegenüber dem Menschen deutlich. So besteht beim Roboter kein „inverses Problem“, da die Tiefeninformationen durch die Sensoren immer

verfügbar ist. Durch die Verwendung einer verteilten Architektur, können genügend Ressourcen zur Verfügung gestellt werden. Die Objektdatenbank stellt die nötigen Daten für eine Kategorisierung bereit und kann durch die Modellierung nicht erkannter oder neuer Objekte zur Laufzeit ergänzt werden. Damit repräsentiert die Datenbank die Möglichkeit der Erinnerung. Zwar existieren mehrere Repräsentationen parallel, dennoch verhält es sich bei Menschen durchaus ähnlich.

Aus der Verwendung unterschiedlicher Sensoren und Sensorarten resultiert die Notwendigkeit einer mehrstufigen Verarbeitungshierarchie. Diese kann, abhängig von den gesuchten Merkmalen, top-down oder bottom-up ablaufen. Parallel zum Abgleich der gefundener Merkmale mit einer Datenbank, soll auch die funktionale Erkennung unterstützt werden. So wird eine parallel zum Boden verlaufende Fläche in ihrer Funktionalität als Tisch erkannt. Denn die Wahrscheinlichkeit Objekte auf solchen Oberflächen zu finden ist sehr groß. Auch die Verwendung des Kontextwissens, entweder die zum Boden verlaufende Fläche oder eine mögliche Untersegmentierung bei der Detektion mehrerer unterschiedlicher Farben, soll von dem realisierten Erkennungssystem unterstützt werden.

Engpässe sind bei der menschlichen Fähigkeit zur Interpretation, Einbindung in ein kontextuelles Umfeld sowie bei der Analyse und Globalisierung erkennbar.

In der vorliegenden Arbeit wird versucht einen kombinierten Ansatz zu implementieren und zu evaluieren. Einzelne Merkmaldetektoren werden nach der Stabilität ihrer Ergebnisse priorisiert und gewichtet. Danach wird durch eine gewichtete Abstimmung ein Konsens gebildet. Dieser Konsens, der auf gefundenen Objekteigenschaften basiert, kann durch Objektrelationen zusätzlich verbessert werden. Wird zum Beispiel ein separiertes Cluster als eine Gabel oder ein Whiteboard-Marker erkannt und sind weitere Objekte innerhalb des 3-D-Bilds vertreten, kann die Entscheidung optimiert werden. Handelt es sich um typische Objekte eines Büros, steigt die Wahrscheinlichkeit für den Marker. Sind es eher Küchenaccessoires, so steigt die Sicherheit für die Wahrnehmung einer Gabel. Abhängig davon wird durch ein regelbasiertes System eine Strategie für das weitere Vorgehen erarbeitet, die durch ein Roboter umgesetzt wird. Somit entsteht ein geschlossener Wahrnehmungskreis, der bei der Notwendigkeit mehrmals durchlaufen werden kann. Dadurch soll nicht nur die Objekterkennung verbessert, sondern auch die Auflösung partieller oder sogar totaler Verdeckung ermöglicht werden.

Anhang C

Kamerakalibrierung

Durch die Kamerakalibrierung werden die Linsenverzerrung sowie die intrinsischen und extrinsischen Parameter bestimmt [HZ04][Sch05]. Dabei beschreiben die intrinsischen Parameter einer Kamera die Projektion der Punkte einer 3-D Welt in das lokale Kamerakoordinatensystem, das Abbild. Die extrinsischen Parameter geben Auskunft über die Lage der Kamera bezüglich eines globalen Koordinatensystems.

In dieser Arbeit kam durchgehend die Kamerakaliebrierungsmethode nach Zhengyou Zhang zum Einsatz [Zha00]. Die Methode liefert die intrinsischen und extrinsischen Kameraparameter sowie die ersten beiden Koeffizienten der Linsenverzeichnung anhand mehrerer aufgenommener Bilder eines Schachbrettmusters, mit den davor bekannten Dimensionen der eingesetzten Kalibrierungsvorlage.

C.0.1. Kamerakalibrierungsverfahren

Nach Luhmann existieren drei grundlegende Verfahren zur Kamerakalibrierung, *Laborkalibrierung*, *Simultankalibrierung* und *Testfeldkalibrierung*. Diese werden anhand von Ort, Zeit oder eingesetzten Referenzkörpern unterschieden. In der vorliegenden Arbeit werden nur die gängigsten Methoden behandelt, da die vollständige Schilderung einzelner Kalibrierungsverfahren [Luh00] diesen Rahmen sprengen würde.

- **Laborkalibrierung**

  Bei der Laborkalibrierung wird mithilfe eines Goniometers und eines hochpräzisen Gitters die innere Orientierung der Kamera bestimmt. Das Verfahren liefert sehr genaue Ergebnisse, ist aber nur für Messkameras sinnvoll und sollte, wie die anderen Verfahren auch, in regelmäßigen Abständen wiederholt werden. Allerdings kann die Laborkalibrierung meistens nicht vom Anwender selbst durchgeführt werden, da das notwendige Equipment teuer ist und der Kalibrierungsprozess einer

entsprechenden Ausbildung bedarf.

- **Simultankalibrierung**

  Bei der Simultankalibrierung wird das zu vermessendes Objekt als Kalibrierungskörper verwendet. Die Kalibrierung findet meistens kurz vor dem geplanten Experiment statt. Dadurch liefert das Verfahren bessere Ergebnisse als die Testfeldkalibrierung, da die Parameter exakt zur Objektaufnahme bestimmt werden. Das Verfahren setzt aber voraus, dass die genaue Geometrie für jeden Kalibrierungskörper neu bestimmt werden soll. Falls die Simultankalibrierung nur mit fotogrammetrischen Beobachtungen durchgeführt wird, wird dieses Verfahren als Selbstkalibrierung bezeichnet.

- **Testfeldkalibrierung**

  Die Testfeldkalibrierung stellt das am weitesten verbreitete Verfahren zur Kamerakalibrierung dar. Dabei wird für die Kalibrierungszwecke immer derselbe Kalibrierungskörper verwendet, entsprechend werden dessen Abmessungen nur einmal bestimmt. Durch Erschütterungen oder Wechsel der Zoomeinstellungen kann sich die Geometrie der Kamera ändern. Um die Genauigkeit der Kamera konstant zu halten, sollte diese in regelmäßigen Abständen kalibriert werden. Der größte Vorteil der Testfeldkalibrierung resultiert aus den Tatsachen, dass das Verfahren einfach ist und permanent der gleiche Kalibrierungskörper eingesetzt wird. Dadurch nimmt die Kalibrierung nur wenig Zeit in Anspruch und kann auch von beliebigen Anwendern selbstständig durchgeführt werden.

Zhengyou Zhang unterscheidet seinerseits nur zwei Kalibrierungsmethoden, nämlich Kalibrierung mit und ohne Kalibrierungskörper. Diese Unterscheidung ähnelt der Klassifikation nach Luhmann, außer dass die Laborkalibrierung aufgrund der oben aufgeführten Nachteile nicht berücksichtigt wird [Zha00].

In dem Buch von Hornberg [Hor06] wird die Laborkalibrierung zwar als Kalibrierungsmethode beschrieben, dennoch werden die Nachteile dieser Methode explizit aufgeführt, was die praxisbezogene Relevanz infragestellt. Die Methode benötigt hochwertiges Equipment und kann in den meisten Fällen nicht vom Anwender selbstständig durchgeführt werden. Damit verursacht die Laborkalibrierung einen enormen zeitlichen und finanziellen Aufwand.

C.0.2. Intrinsische Kameraparameter

Die intrinsischen Parameter beschreiben die interne Geometrie der Kamera, dazu gehören die Auflösung des verwendeten Sensor-Chips und die Positionierung des Ursprungs des

Koordinatensystems auf dem Sensor-Chip sowie die Koeffizienten der Linsenverzerrung [FFH+92]. Da die intrinsischen Parameter die Zusammenhänge nur innerhalb der Kamera beschreiben, sind diese von der Lage und Ausrichtung der Kamera im Weltkoordinatensystem unabhängig.

Der Abbildungsprozess eines Weltpunkts auf die Bildebene kann vollständig über die perspektivische Projektion aus Gleichung (C.1) beschrieben werden.

$$P_c = A\ [R\ T] \cdot P_w \tag{C.1}$$

Die Rotation R und die Transformation T werden als extrinsische Kameraparameter zusammengefasst. Die Matrix A beinhaltet die intrinsischen Parameter der Kamera, die aus der Brennweite f und der u_0- und v_0-Koordinate des Hauptpunkts bestehen. Die Parameter fk_u und fk_v stellen dabei die horizontalen und vertikalen Skalierungsfaktoren dar. Der Parameter s, der so genannte *skew parameter* charakterisiert die bei der Projektion entstehende Verzerrung. Der Aufbau der Matrix sieht dann wie folgt aus:

$$A = \begin{bmatrix} fk_u & s & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.2}$$

Bei der Betrachtung der Matrix A wird ersichtlich, dass bekannte intrinsische Parameter für die Wiederherstellung des Zusammenhanges zwischen dem Kamera- und Bildkoordinatensystem genutzt werden können. Diese Zusammenhänge bilden gemeinsam mit den extrinsischen Parametern und Koeffizienten der Verzerrung eine notwendige Basis für die später folgende Tiefenrekonstruktion.

Im nächsten Unterkapitel werden weitere Eigenschaften der Kamera, die extrinsischen Parameter, theoretisch behandelt. Zuerst wird auf deren Bedeutung eingegangen. Anschließend wird die Bestimmung der Rotation und Transformation mathematisch beschrieben.

C.0.3. Extrinsische Parameter

Extrinsische Parameter beschreiben die Lage der Kamera bezüglich eines globalen Koordinatensystems [FFH+92]. Die Parameter bestehen aus der Rotation und Translation der Kamera. Somit lässt sich die Transformation vom Weltkoordinatensystem in das Kamerakoordinatensystem durch zwei Matrizen darstellen, einer 3×3 Rotationsmatrix und einem 3×1 Translationsvektor. Die mathematische Abbildung eines 3-D Punkts des Raums auf einen 2-D Punkt des Kamerakoordinatensystems C.3 sieht folgendermaßen aus:

$$P_c = P_w \cdot R + T \tag{C.3}$$

wobei P_c und P_w die Koordinaten eines Punkts im Kamerakoordinatensystem, beziehungsweise Weltkoordinatensystem beschreiben und R und T jeweils Rotationsmatrix und Translationsvektor darstellen. Die Rotation und die Translation können in einer Matrix zusammengefasst werden, siehe Gleichung (C.4).

$$P_c = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_w \tag{C.4}$$

Dabei werden beide Matrizen um jeweils eine Dimension erweitert, sodass ein 3-D-Punkt P(x, y, z), der auch als Vektor zwischen dem Ursprung und Punkt P aufgefasst werden kann, zu einem 4-D-Punkt P' (kx, ky, kz, k) wird. Die Komponente k stellt einen beliebigen von 0 unterschiedlichen Skalar dar.

In der Mathematik wird das oben beschriebene Verfahren als perspektivische Projektion bezeichnet. Dabei werden alle Punkte eines 3-D-Modells entlang einer Linie über das optische Zentrum auf eine Fläche der Bildebene projiziert. Dadurch entsteht ein zweidimensionales Abbild eines dreidimensionalen Modells.

Aus der perspektivischen Projektion lassen sich zwei grundlegenden Eigenschaften ableiten. Zum einen, um zu den ursprünglichen euklidischen Koordinaten eines Punkts zurückzukehren, reicht es, die ersten n-Elemente eines Vektors durch das $n + 1^{te}$-Element zu dividieren. Zum anderen werden zwei Punkte eines dreidimensionalen Modells genau dann auf einen zweidimensionalen Punkt projiziert, wenn sie durch die Skalierung ineinander überführt werden können.

C.0.4. Verzerrungen

Wie schon im vorangegangenen Abschnitt erwähnt, gibt es keine ideale Linse oder ideale Sensoren. Somit entstehen Verzerrungen in der Abbildung und Abweichungen zum Lochkamera-Modell. Diese sogenannten nicht linearen Effekte sollen unter Zuhilfenahme inverser Transformation behoben werden. Dafür werden bei der Kamerakalibrierung die Eigenschaften der Verzerrung bestimmt. Die Faktoren der Verzerrung gehören zu den intrinsischen Kameraparametern. Es wird zwischen zwei Arten der Linsen- beziehungsweise Sensor-Verzeichnung unterschieden, die tangentiale und die radiale Verzerrung.

In der Abbildung C.1 wird die tangentiale und radiale Linsenverzerrung grafisch dargestellt. Da die tangentiale Linsenverzerrung das Bild nur sehr geringfügig verfälscht, wird diese bei der hier verwendeten Kamerakalibrierungsmethode nach Zhengyou Zhang [Zha00] wie auch nach der Methode von Roger Tsai [Tsa87] nicht berücksichtigt.

Bei der radialen Linsenverzerrung wird wiederum zwischen zwei Arten unterschieden, der kissenförmigen und tonnenförmigen Verzerrung. Die Abbildung C.2 stellt die beiden Arten grafisch dar. Um das Modell der Lochkamera verwenden zu können, soll die radiale Verzerrung aus den Bildern herausgerechnet werden. Dafür werden mit bei der Kamerakalibrierung gewonnenen Daten die Bilder entzerrt und zur Gewinnung der intrinsischen und extrinsischen Kameraparameter weiterverwendet.

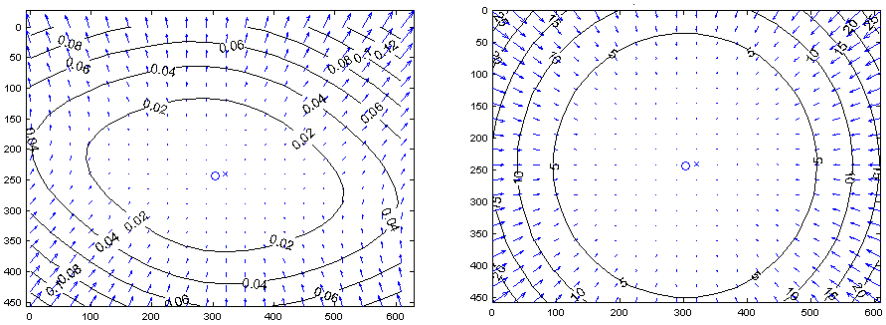


Abbildung C.1.: Beispiele für tangentiale (links) und radiale Linsenverzerrung (rechts). Wie aus den Bildern ersichtlich ist, wird der Punkt bei der tangentialen Verzerrung entlang der Tangente durch den Mittelpunkt verschoben. Bei der radialen Linsenverzerrung wird der Abstand des Punkts zum Mittelpunkt verändert.

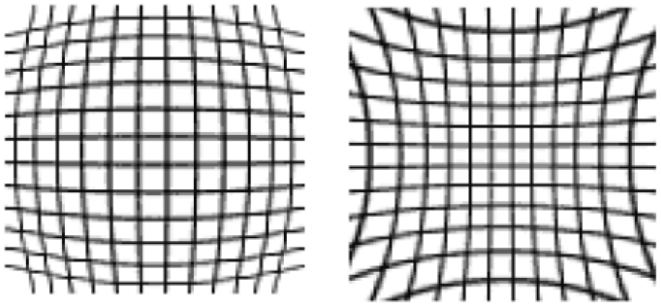


Abbildung C.2.: Darstellung der tonnenförmigen (links) und kissenförmigen (rechts) Verzerrungen, die aus der radialen und tangentialen Verzerrung resultieren.

Anhang D

Erfassung der Umgebung

Dieses Kapitel beschäftigt sich mit der Erfassung einer unbekannten und/oder einer dynamischen Veränderungen unterliegenden bekannten Umgebung. Hierbei steht die Auffindung von sogenannten ROIs (engl. Region Of Interest) im Vordergrund, dabei handelt es sich um planare, parallel zum Boden ausgerichtete Flächen wie zum Beispiel Tische, Fensterbänke und Ähnliches. In alltäglichen Situation sind das die Flächen, wo die Wahrscheinlichkeit, ein oder mehrere Objekte vorzufinden, extrem hoch ist. Damit vervollständigt dieses Kapitel die vorliegende Dissertation und beschreibt den oben dargestellten, mit den gestrichelten Linien begrenzten, Block in der Abbildung 1.3.

D.1. Entwicklung eines 3-D-simultanen, lokalisierenden und kartierenden Explorationssystems

Betrachtet wird zuerst die unbekannte Umgebung. Da für die in der vorliegenden Arbeit beschriebenen Szenarien eine komplette 3-D-Karte benötigt wird, muss besonderes auf die Vollständigkeit der Exploration geachtet werden.

Die meist bekannten Algorithmen sind zweidimensional, basierend auf Fernsteuerung, zufallsgenirierter Bewegung oder Wandverfolgung durch den Roboter, sogenannte „wall-follow“-Algorithmen. Für einen guten Überblick über die state-of-the-art-Verfahren sei der Leser auf das Buch von C. Stachniss [Sta09] verwiesen. Da bei den hier verwendeten Robotern eine 3-D-Kollisionsvermeidung notwendig ist und der Roboter sich autonom bewegen soll, können solche Algorithmen jedoch nicht verwendet werden. Die „wall-follow“-Algorithmen garantieren zusätzlich keine vollständige Exploration der Umgebung und neigen dazu, in eine ständig wiederkehrende Kreistrajektorie hinein zugeraten.

Abbildung D.1 visualisiert die Exploration mit dem sogenanntem „wall-follow“-Algoritmus, dabei folgt der Roboter den Wänden in einer bestimmten Richtung. Farbliche Bereiche veranschaulichen die durch die Sensoren des Roboters erfasste Umgebung:

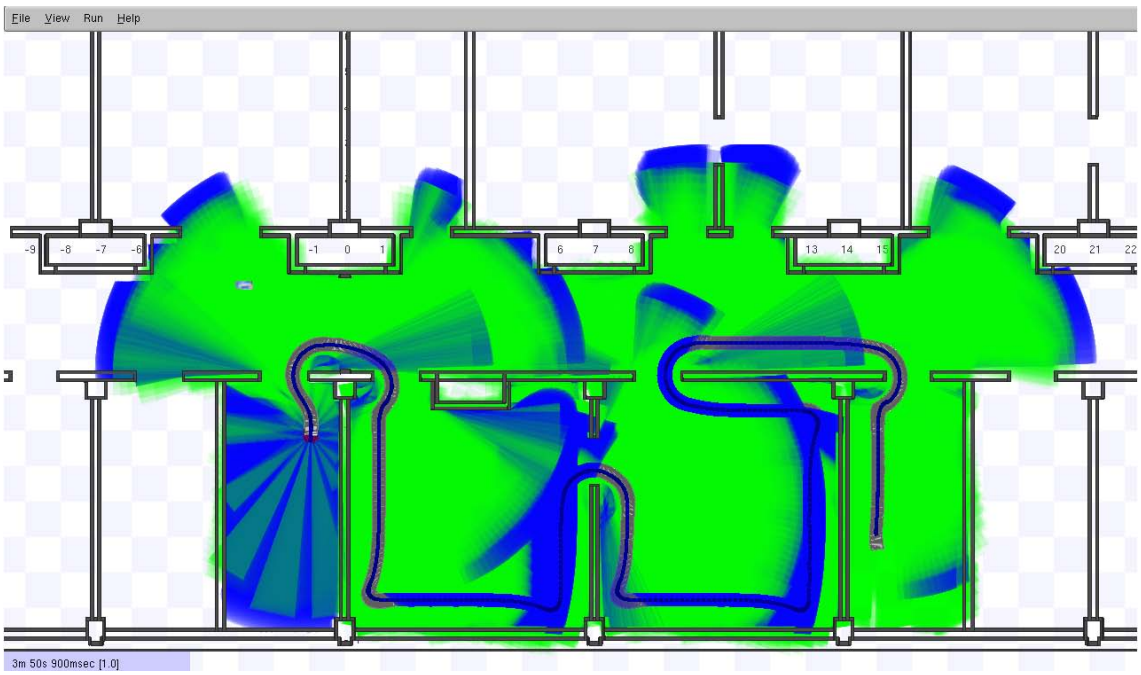


Abbildung D.1.: Umgebungsexploration mit dem „wall-follow„-Algorithmus, dabei folgt der Roboter den Wänden in eine bestimmte Richtung. Farbliche Bereiche visualisieren die erfasste Umgebung (grün durch die Laser- und Sonarsensoren; blau nur durch den Laserscanner).

grüne sind durch die Laser- und Sonarsensoren und blaue nur durch den Laserscanner erkannte Bereiche. Die Abbildung D.2 stellt den ungünstigsten Fall eines „wall-follow“-Algorithmen dar. Der Roboter bewegt sich in einer Kreistrajektorie, die Exploration der Umgebung kann in einer solchen Situationen ohne externen Eingriff nicht fortgesetzt werden.

Somit wurde keine passende und bereits implementierte Methode zur Erfassen einer Umgebung gefunden, und es wurde daher entschieden, einen eigenen Algorithmus für die Erfassung der unbekannten Umgebung zu entwickeln und zu implementieren, welches in der gemeinsamen Arbeit während und nach der Diplomarbeit mit Gregor Mechalicek verwirklicht wurde [MKZ11]. Ein weiterer Vorteil dieser Methode liegt darin, dass der gleiche Algorithmus unter Berücksichtigung der temporalen Kriterien auch für die bekannte Umgebung angewandt werden kann.

Der entwickelte Algorithmus besteht aus mehreren Komponenten und repräsentiert ein verteiltes System, wobei jede Komponente ein individuelles Programm ist und einzeln

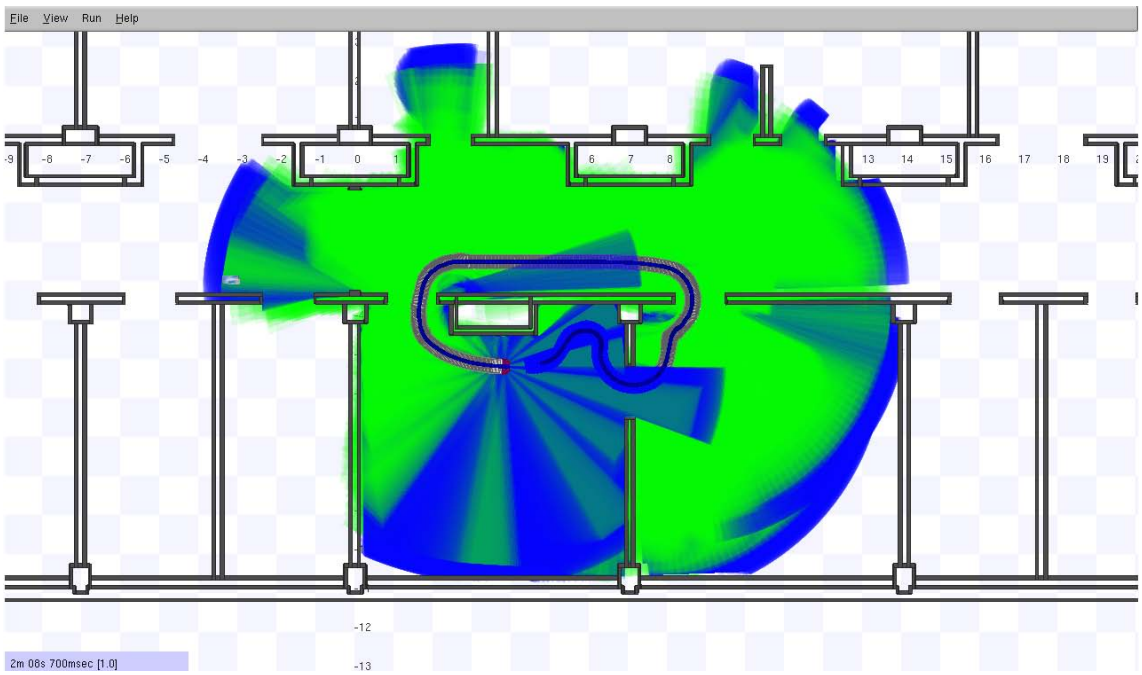


Abbildung D.2.: Visualisierung einer der Nachteile des „wall-follow“-Algorithmus. Der Roboter bewegt sich in einer Schleife, die Exploration der Umgebung kann ohne externen Eingriff nicht fortgesetzt werden. Farbliche Bereiche visualisieren die erfasste Umgebung (grün durch die Laser- und Sonarsensoren; blau nur durch den Laserscanner).

ausgeführt werden kann. Die Beziehung zwischen den einzelnen Komponenten ist in Abbildung D.3 grafisch dargestellt.

Die zentrale Komponente des verteilten Systems ist das Programm *RobotController*, dass die Kommunikation zwischen allen Komponenten kontrolliert. Die nächste wichtige Komponente ist *RobotConnector*. Die Bewegung des Roboters und die Datenakquisition mittels robotereigener Sensoren wird vom *RobotController* initiiert und an *RobotConnector* weietergegeben. *RobotConnector* sendet Scaner- und/oder Odometriedaten zurück. Der nächste Schritt ist die Konstruktion einer 3-D-Karte der Umgebung. Dafür sendet das Kontrollprogramm die von dem Roboter erhaltenen Daten an *MapGenerator*. Der seinerseits den “Simultaneous Localization And Mapping Algorithm” (SLAM) darauf anwendet. Der Algorithmus verwendet die Odometrie- und die extrinsischen Sensordaten für die genaue Lokalisierung und anschließende Kartenbildung. Die so bestimmte Information wird zurück an den *RobotController* gesendet. Der nächste Schritt ist die

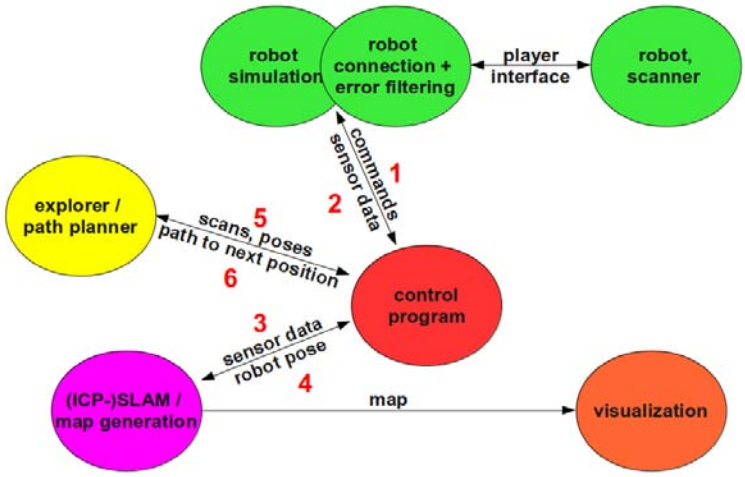


Abbildung D.3.: Visualisierung des verteilten Explorationssystems. Im Detail besteht das System aus der zentralen Komponente *RobotController* einem Kontrollprogramm. Dazu aus dem Verbindungsprogramm *RobotConnector*, dem SLAM realisierenden Programm *MapGenerator* und der Explorations- und Planungskomponente *Explorer*.

Bestimmung der nachfolgenden Scanposition des Roboters. Dafür wird die aktuelle Position des Roboters sowie die bis dahin erstellte Karte an die *Explorer*-Komponente, die die nächst beste Scanposition ermittelt, weitergereicht. Im Folgenden wird der Pfad zu der errechneten Position bestimmt und an das *RobotController* übergeben. Damit stehen dem *RobotController* genügend Information zur Verfügung für die Generierung einer Befehlskette zur Bewegung des Roboters. Diese Sequenz wird an den *RobotConnector* gereicht, was das Explorationszyklus abschließt.

Im Folgenden wird der Algorithmus ausführlich dargestellt und beschrieben.

D.2. Grundidee des Algorithmus

Der hier vorgestellter Algorithmus ist für die Innenräume entwickelt worden, daher setzen wir einen glatten Untergrund voraus, der aber in der Karte nicht direkt repräsentiert wird. Damit die Roboterbewegung geplant werden kann, soll der Flur aus der Karte extrahiert werden. Die Extraktion stellt den ersten Schritt des Algorithmus dar. Der nächste Schritt ist die Auswahl passender Landmarken auf dem extrahierten Flur. Der Roboter soll sich auf den geraden Bahnen zwischen diesen speziellen Punkten bewegen. Natürlich soll der nächste anzufahrende Punkt anhand der bis dahin akquirierten Karte bestimmt werden. Zu diesem Zweck werden zwei 2-D-Karte generiert. Die erste

Karte beinhaltet die Kosten für die Bewegung zu den davor bestimmten Positionen, die zweite spiegelt das bis dahin aus den Sensordaten gewonnene Wissen über die Umwelt wieder. Die Entscheidung, welche Position als nächstes angesteuert wird, ist ein Optimierungsproblem auf der Basis dieser beiden Karten. Der letzte Schritt des Explorationsalgorithmus ist die Pfadplanung zwischen dem Startpunkt, den Landmarken und der Zielposition.

Algorithm 3 The exploration algorithm

```
1: procedure EXPLORE(x_k, m)
2:     Extract the floor from the map data m.
3:     Erode the floor by the robot radius.
4:     Generate a homotopy preserving skeleton of the accessible floor.
5:     Select landmarks on the skeleton.
6:     Generate a graph of direct reachability between the landmarks.
7:     Generate the cost (distance) map for the accessible floor based on the basis of
       the current pose x_k and the landmarks.
8:     Generate the knowledge map on the basis of scan locations in map m.
9:     Select a point on the accessible floor with low knowledge and low cost.
10:    Plan the path to the selected point.
11:    return path
12: end procedure
```

Der Ansatz für jeden einzelnen Explorationsschritt ist in dem oben dargestellten Algorithmus 3 zusammengefasst. Nach der Flurextraktion wird die Umgebung mit dem Sicherheitsradius des Roboters erodiert, sodass der Roboter auf einen Punkt reduziert wird (vgl. Minkowski Differenz). Dafür wird zuerst eine Karte mit der Euclidean Distance Transformation (EDT) unter Berücksichtigung der Flurabgrenzungen für jeden Untergrundpunkt definiert. Danach werden alle Punkte, deren Distanz kleiner als der Robotersicherheitsradius ist, entfernt. Der nächste Schritt ist die Auffindung der passenden Landmarken auf dem Flur. Dafür werden zuerst die lokalen Maxima aus der EDT selektiert und als Anker für die Generierung eines Homotopie aufrechterhaltenden Skeletts genutzt. Zusätzlich zu den Ankerpunkten werden spezielle Punkte wie zum Beispiel Abzweigungen als Landmarken gewählt. Falls ein spezieller Punkt des Skeletts nicht direkt vom Nachbarpunkt erreicht werden kann, werden Zwischenpunkte selektiert. Zusammengefasst kann jeder erreichbarer Punkt des Flurs von einem der Ankerpunkte direkt erreicht werden. Natürlich könnte eine Situation konstruiert werden, wo dieses nicht zutrifft. Zum Beispiel hat der Flur eine perfekte Kreisform, es existieren keine lokalen Maxima in der EDT und damit keine Ankerpunkte. Somit ist es zweckmäßig, solche Situationen während der Exploration von vornherein auszuschließen. Unter diesen Voraussetzungen kann der Roboter von jedem Ankerpunkt jeden Flurpunkt erreichen. Die gewählten speziellen Punkte des Skeletts garantieren, dass jeder Ankerpunkt von

jedem anderen Ankerpunkt erreicht werden kann, soweit eine physikalische Verbindung besteht. Basierend auf den Landmarken des Skeletts und der aktuellen Position x_k kann die Bildung der Kostenkarte (Darstellung der Distanzen) fortgesetzt werden. Des Weiteren wird eine Wissenskarte anhand der früheren Scanpositionen generiert. Aufbauend auf den beiden Karten wird die nächste Scanposition ermittelt. Abschließend wird der Pfad von der Position x_k zur der neu bestimmten Scanposition, basierend auf den Verbindungen zwischen den Landmarken sowie Start- und Zielpunkten, errechnet.

D.3. ICP SLAM

Es existiert eine Menge Algorithmen zur Lösung des SLAM-Problems [TBF06]. Dennoch sind Algorithmen, die 3-D-Daten verarbeiten, bisher selten. Einige der Algorithmen wie der Extended-Kalman-Filter (EKF) SLAM sowie der Sparse Extended Information Filter (SEIF) SLAM arbeiten mit Merkmalsextraktion. Da wir unseren Ansatz so global wie möglich halten wollen, stellen diese Art von Algorithmen keine Option dar. Aus diesem Grund kommen FastSLAM in Kombination mit den Occupancy Grid Mapping und Iterative Closest Point (ICP) SLAM [BM92] in Betracht. Jeder der beiden Algorithmen hat seine Vor- und Nachteile. In der vorliegenden Arbeit wurde entschieden, den ICP-SLAM zu verwendet, da er bereits mit sechs Freiheitsgraden (DOF) für den 3-D-SLAM erprobt und evaluiert worden ist [NLHS07].

Die Karte m wird im ICP-SLAM durch eine Sequenz von Roboterpositionen mit korrespondierenden Punktwolken der erfassten Umgebung repräsentiert. Für jeden Schritt k des Roboters wird zuerst seine Position x_k geschätzt. Der Algorithmus vergleicht die gemessenen Oberflächenpunkte $\{d_{j,k}\}$ diesen Schritts mit den Punkten $\{m_i\}$ in der Karte und minimiert den quadratischen Fehler durch folgende Funktion

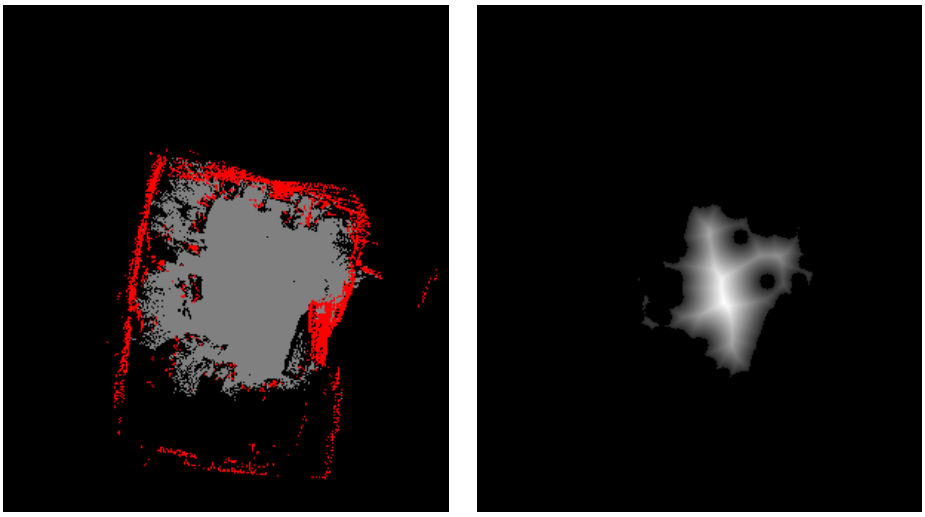
$$E(R,t) = \sum_i \sum_j w_{i,j} \left\| m_i - R d_{j,k} + t \right\|^2 \tag{D.1}$$

in Bezug auf die Rotationsmatrix R und den Translationsvektor t und angewandt auf die Roboterposition x_k oder als Konsequenz direkt auf die Messwerte $\{d_{j,k}\}$. Dabei repräsentiert $w_{i,j}$ die gewichtete Funktion der Korrespondenz der Punkte $\{m_i\}$ und $\{d_{j,k}\}$. Zusammengefasst stellt die Gleichung D.1 einen Ausdruck für die Summe der quadratischen Distanzen zwischen den korrespondierenden Punkten beider Punktwolken dar. Der ICP-SLAM minimiert iterativ diese Gleichung und korrigiert damit die initiale Position x_k des Roboters. Soweit alle Konvergenzkriterien erfüllt werden, terminiert der Algorithmus. Als letzter Schritt werden schließlich die korrigierten Messwerte der Karte hinzugefügt.

Abschließend soll erwähnt werden, dass in der vorliegenden Realisierung der ICP-SLAM-Algorithmus die obere Gleichung nur in Bezug auf die in der Ebene gegebenen drei Freicheitsgrade minimiert.

D.4. Der erreichbare Flur

Der Startpunkt der Flurextraktion ist die Karte m, die auf der Sequenz der Roboterpositionen und den dazugehörigen Messwerten, die 3-D-Punktwolken, für diese Positionen basiert. Zuerst wird der relevante Bereich in regelmäßige Gitterzellen unterteilt.



(a) (b)

Abbildung D.4.: (a) Extrahierter Flur. Die Abbildung visualisiert den Flur nach der Beendigung der davor beschriebenen Prozedur. Graue Pixel markieren freie Bereiche, rote die Hindernisse, die schwarzen Bereiche stellen die Positionen dar, von denen noch kein Wissen über den Zustand der Zellen vorhanden ist. (b) Distanztransformation (EDT) des Flurs. Nicht erreichbare Punkte sowie die Hindernisse sind im vorhergehenden Schritt entfernt worden.

Jeder einzelnen Zelle wird ein Zustand zugeordnet: Die Zelle ist frei, repräsentiert ein Hindernis, oder es existiert noch kein Wissen über den Zustand der Zelle. Dafür definieren wir eine Ebene d_{floor} über den Flur. Für jeden eingescannten Punkt wird die Zugehörigkeit zu dieser Ebene überprüft. Falls die Bedingung zutrifft, wird die korrespondierende Zelle in der Karte als Flurzelle markiert. Über die Flurebene wird eine Ebene d_{robot}, die die Höhe des Roboters hat, drübergelegt. Falls ein Messpunkt innerhalb dieser Ebene liegt, wird die korrespondierende Zelle als Hinderniszelle markiert. Nachdem diese beide Schritten abgeschlossen sind, wird den Zellen, die weder frei noch als Hindernis markiert sind, ein unbekannter Zustand zugeordnet. Da bei dem Scanprozess der Boden unter dem Roboter nicht gescannt werden kann, werden die Zellen rund um die Scanposition des Roboters mit dem Radius des Sicherheitspolygons als Flurzellen

markiert.

Abbildung D.4(a) visualisiert den Flur nach der Beendigung der oben beschriebenen Prozedur. Graue Pixel markieren freie Bereiche, rote die Hindernisse und die schwarzen Bereiche kennzeichnen die Positionen, wo noch kein Wissen über den Zustand der Zellen vorhanden ist. Das rechte Bild (D.4(b)) präsentiert die Distanztransformation (EDT) des Flurs. Nicht erreichbare Punkte und die Hindernisse sind davor entfernt worden. Der nächste Schritt ist die Determination der Distanztransformation für die Flurzellen in der Flurkarte in Bezug auf alle anderen Zelltypen. Dafür wurde der Algorithmus von Saito *et al.* [ST94] implementiert, der unter anderem sehr gute Ergebnisse in den Publikationen wie beispielsweise [FdFCTB08] erzielte.

Der Algorithmus nimmt als Parameter ein 2-D-Bild f mit Breite W und Höhe H entgegen, dass die Vordergrund- (1) und Hintergrundpixel (0) beinhaltet. In unserem Algorithmus stellen die Flurzellen die Vordergrundpixel dar. Die Methode berechnet die Quadratdistanz jedes Vordergrundpixels zur dem näheren Hintergrundpixel. Dafür wird zuerst ein Bild g nach der folgenden Gleichung bestimmt:

$$g_{ij} = \min_x \{(i-x)^2 | f_{xj} = 0, 1 < x < W\}. \tag{D.2}$$

Die Gleichung stellt die Transformation des Bilds in i-Richtung dar. Dabei beinhaltet die g_{ij} die Quadrate der Distanzen des Vordergrundes in Bezug zu dem Hintergrund in die angegebene Richtung. Die Berechnung kann einmal durch den Scan f von links nach rechts und einmal von rechts nach links geschehen, deswegen hat das Verfahren eine Komplexität von $\mathcal{O}(n)$ mit n als Anzahl der Pixel.

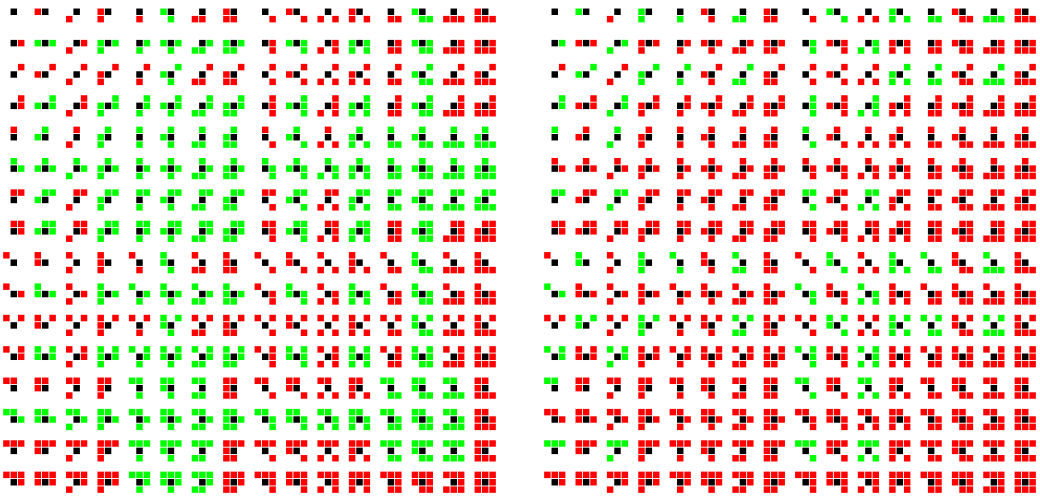
Der nächste Schritt ist die Transformation in die j-Richtung zur Berechnung des Bilds h nach der folgenden Gleichung:

$$h_{ij} = \min_y \{g_{iy} + (j-y)^2 | 1 < y < H\}. \tag{D.3}$$

Das neue Bild beinhaltet die Quadrate der nötigen EDT. Dieser letzter Schritt hat eine Komplexität von $\mathcal{O}(n \cdot \text{Average}\{\sqrt{g_{ij}}\})$. Es sei darauf hingewiesen, dass der Wert $\sqrt{g_{ij}}$ unabhängig von der Auflösung des Bilds ist. Damit hat der komplette Algorithmus eine Komplexität von $\mathcal{O}(n)$. Für die Anwendungen in Innenräumen stimmt diese Annahme für die ausgedehnten Karten, da normalerweise die Größe eines Raums unabhängig von der Größe der Karte ist. Nach der Kalkulation der EDT werden alle Punkte, deren Abstand zur Begrenzung kleiner als der Robotersicherheitsradius ist, gelöscht. Damit wird der Roboter auf einen Punkt reduziert (vgl. Minkowski-Differenz). Der übrig gebliebene Untergrund ist somit der für den auf einen Punkt reduzierten Roboter erreichbare Flur. Ein Beispiel ist in der Abbildung D.4(b) zu sehen. Die übrig gebliebene Flurpixel werden für die Bestimmung der Landmarken auf dem Flur genutzt.

D.5. Berechnung der Landmarken auf dem Flur und die Generierung des Skelettes

Damit die Landmarken bestimmt werden können, wird zuerst ein Homotopie erhaltendes Skelett des Flurs generiert. Es existiert eine Vielzahl solcher Algorithmen [SHB98], hier wird der Algorithmus von Vincent [Vin91] verwendet, da er neben der Homotopieerhaltung weitere positive Eigenschaften aufweist. Der Algorithmus arbeitet auf einem hexagonalen Gitter. Da unser Algorithmus eine rechteckige Gitterstruktur nutzt, wird der Algorithmus an diese angepasst. In der vorliegenden Arbeit wird eine Konnektivität von vier angenommen. Diese stellt eine wichtige Entscheidung dar, weil die Konnektivität die Homotopie beeinflusst.

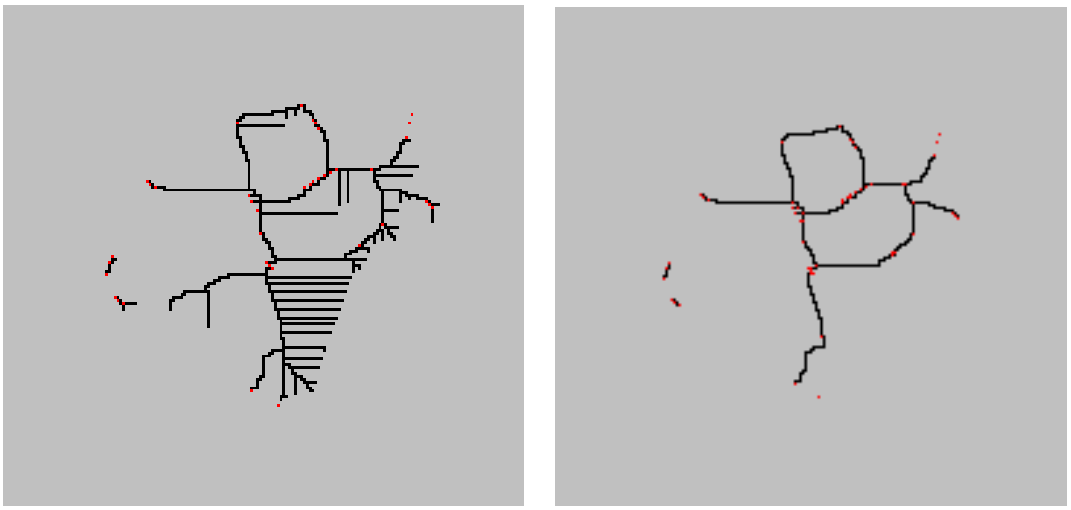


(a) (b)

Abbildung D.5.: (a) Homotopie-relevante Pixel für jede mögliche Kombination von gesetzten und nicht gesetzten Pixeln in der Nachbarschaft des schwarzen Referenzpixels, damit wird die Wichtigkeit des Referenzpixels für die Homotopie aufgezeigt. Das ist der Fall, wenn die Nachbarpunkte grün sind. Falls die Nachbarpixel rot sind, ist der Referenzpixel für die Homotopie nicht relevant. (b) Entfernung relevanter Pixeln für jede mögliche Kombination von gesetzten und nicht gesetzten Pixeln in der Nachbarschaft des schwarzen Referenzpixels. Damit wird visualisiert, welche Pixel aus dem Skelett entfernt werden können. Das ist der Fall, wenn die Nachbarpixel grün markiert sind.

Der Algorithmus von Vincent basiert auf drei Schritten. Der erste, initialisierende Schritt setzt die Ankerpunkte. Dieser Schritt beinhaltet die Erzeugung und das Befüllen

von Datenstrukturen, die der Algorithmus temporal nutzt. Der zweite Schritt, die sogenannte Ausbreitung (propagation step), generiert das ursprüngliche Skelett. Der letzte Schritt schneidet die Abzweigungen des initialen Skeletts ab und erzeugt das endgültige Skelett. Selbstverständlich soll der Algorithmus Wissen über die für die Homotopie des Skeletts wichtigen Pixel sowie über die zu entfernenden Pixels besitzen. Dieses Wissen wird in den Tabellen gespeichert. Für die hexagonalen Gitter können die Tabellen bei [Vin91] oder [Vin90] angeschaut werden. In der vorliegenden Arbeit werden die Tabellen auf die rechteckigen Gitter angepasst und sind in der Abbildungen D.5(a) und D.5(b) visualisiert. Die Abbildung D.5(a) verdeutlicht, dass wenn einer der Homotopie-relevanten Pixels entfernt würde, sich die Homotopie des Skeletts nachhaltig ändern würde. Die Abbildung D.5(b) zeigt die Pixels, die entfernt werden können. Als Startparameter nimmt der Algorithmus von Vincent ein Bild I mit Vordergrund- (1) und Hintergrundpixeln (0) entgegen. Ein weiteres Bild I_a beinhaltet Ankerpunkte des Skeletts. Diese Pixels werden in dem Bild mit 1 markiert, alle anderen werden auf 0 gesetzt. Als Ankerpunkte werden die lokalen Maxima der EDTs eingesetzt und das Ergebnis des Algorithmus in Bild I geschrieben.



(a) (b)

Abbildung D.6.: (a) Skelett nach dem zweiten Schritt. Das Skelett hat nach diesem Schritt einige unbrauchbare Abzweigungen, die, um das endgültige Skelett zu erhalten, entfernt werden müssen. Die roten Punkte stellen die Ankerpunkte des Skeletts dar. (b) Ein endgültiges Skelett mit hinzugefügten Landmarken. Die roten Punkte zeigen die genutzten Landmarken sowie die Ankerpunkte des Skeletts.

Der Algorithmus startet mit dem Initialisierungsschritt, dabei sind die Ankerpunkt

im i mit 1 markiert und jeder weitere Vordergrundpixel mit 2. Als nächstes wird eine FIFO-Queue mit den Randpixeln in I befüllt, die mit 3 markiert werden. Die Randpixel sind die Vordergrundpixel, welche Hintergrundpixel in ihrer direkten Nachbarschaft aufweisen. Nach der Initialisierung der FIFO-Queue kann mit dem zweiten Schritt begonnen werden. Dieser betrachtet jedes Pixel p der FIFO-Queue und durchläuft folgende Instanzen. Zuerst wird jedes Nachbarpixel p_n betrachtet: Falls dieser mit einer 2 markiert ist, also $I(p_n) = 2$, wird er in die FIFO-Queue aufgenommen und mit einer 3 markiert. Danach wird auf der Basis von Nachbarkonfiguration und der in der Abbildung D.5(a) visualisierten Tabelle die Relevanz für die Homotopie überprüft. Falls dieser Fall vorliegt, wird es zu der Datenstruktur namens TAB hinzugefügt, sonst wird $I(p)$ auf 0 gesetzt. Damit wird p aus dem Vordergrund und folglich aus dem Skelett entfernt. Das Ergebnis dieser Aktion ist ein Skelett mit einigen unbrauchbaren Abzweigungen. Ein Beispiel für ein solches Skelett kann in der Abbildung D.6(a) betrachtet werden.

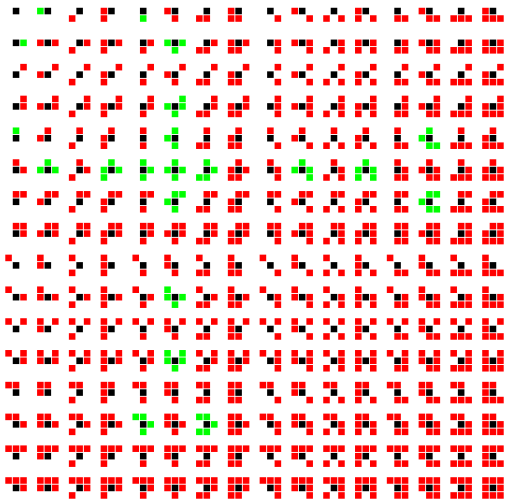


Abbildung D.7.: Spezielle Punkte des Skeletts. Für jede mögliche Kombination von gesetzten und nicht gesetzten Pixel in der Nachbarschaft des schwarzen Referenzpixels wird der Punkt als speziell markiert. Das ist der Fall wenn die Nachbarpunkte in Grün markiert sind.

Der dritte Handlung des Vincent-Algorithmus beginnt mit der Überprüfung aller Punkte in TAB. Das geschieht durch den Vergleich mit der in Abbildung D.5(b) dargestellten Tabelle. Falls der Pixel p entfernt werden kann, wird $I(p)$ mit 2 und damit als Randpixel markiert und in Folge zu der FIFO-Queue hinzugefügt. Als Nächstes werden die unnötigen Abzweigungen entfernt. Dafür wird jedes Pixel p der FIFO-Queue überprüft. Falls der Punkt entfernt werden kann, wird $I(p)$ auf 0 gesetzt und jedes Pixel p_n aus der Nachbarschaft des p für den $I(p_n) = 3$ in die FIFO-Queue hinzugefügt und

mit 2 markiert. Falls der Punkt nicht entfernt werden kann, wird $I(p)$ mit 3 markiert. Sobald der Vorgang abgeschlossen ist, werden alle Pixel in $I(p)$, die mit 3 markiert sind, auf 1 gesetzt. Damit werden die unnötigen Extremitäten des Skeletts entfernt und das endgültige Skelett ist gegeben als alle Punkte in I, die mit 1 markiert sind. Das Ergebnis dieser Aktion ist in der Abbildung D.6(b) dargestellt. Sie beinhaltet neben dem Skelett spezielle Punkte, die als Landmarken gewählt worden sind. Die Auswahl dieser Punkte basiert auf der in der Abbildung D.7 visualisierten Tabelle. Zusätzliche Punkte werden dem Skelett dann hinzugefügt, wenn der Pfad zwischen den benachbarten Punkten keine geraden Linie ist.

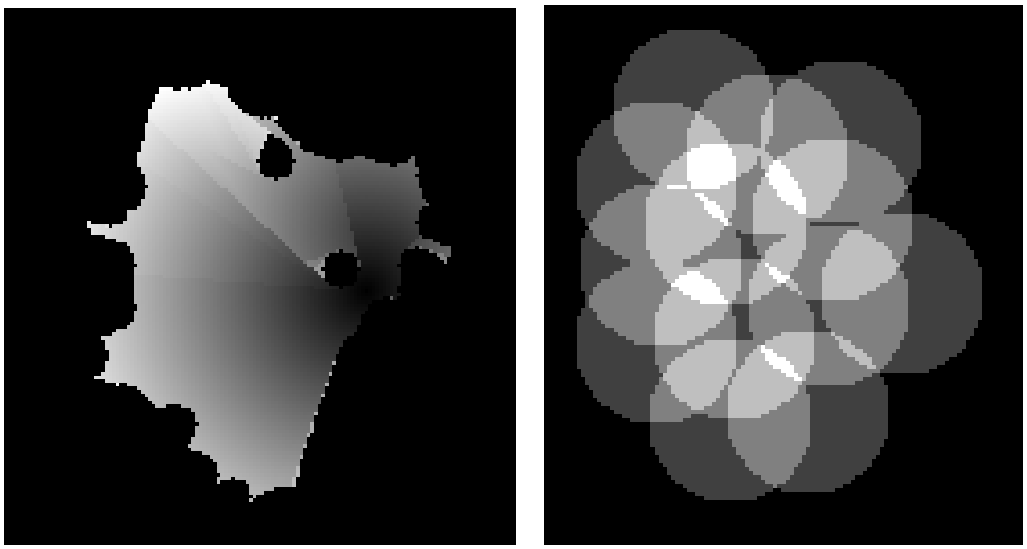
Nachdem das endgültige Skelett generiert worden ist, wird im nächsten Kapitel mit der Definition der Kosten- und Wissenskarte fortgefahren.

D.6. Kosten- und Wissenskarte

Die Kosten- und die Wissenskarte werden zur Bestimmung nächster anzufahrender Scanposition verwendet. Die Kostenkarte spiegelt die Kosten der Bewegung des Roboters zu einer Position wider. Die Wissenskarte dagegen repräsentiert das Wissen, das bereits über diesen Punkt vorhanden ist. Im Prinzip ist die Kombination dieser beiden Karten zur Bestimmung der nächsten Scanposition arbiträr. Im Wesentlichen ist nur relevant, dass ein sinnvolles Verhalten des Roboters daraus resultiert.

Die Kostenkarte basiert auf der Distanz zu einem gegebenen Punkt. Der Punkt kann entweder direkt oder über das Abfahren einer Sequenz von Landmarken und dann weiter zum Punkt erreicht werden. Zusätzlich wird die maximale Distanz, die der Roboter auf einer Linie zurücklegt, definiert. Die Kostenkarte wird somit folgendermaßen generiert: Zuerst wird direkte Distanz zu dem Punkt oder über das Abfahren einer Sequenz von Landmarken und dann zum Punkt berechnet. Danach wird überprüft, welche Landmarken von dem Startpunkt direkt erreicht werden können. Als Nächstes wird auf einem Graph, der die Konnektivität zwischen den Landmarken reflektiert, der *Dijkstra Algorithm* zur Bestimmung des kürzesten Pfades verwendet. Sobald die Distanzen kalkuliert sind, wird für jeden Punkt in der lokalen Nähe zu der jeweiligen Landmarke und der Startposition überprüft, ob der Punkt direkt erreicht werden kann. Ist das der Fall, wird geprüft, ob die kürzeste Distanz bereits bekannt ist. Falls nicht, wird der Punkt mit der gegebenen Landmarke oder der Startposition assoziiert. Damit ist der Punkt mit der kompletten Strecke, die zurückgelegt werden muss, assoziiert. Bei der Berechnung der Distanzen wird die lokale Umgebung durch die maximale direkte Bewegungsdistanz angegeben. Abbildung D.8(a) stellt die Kostenkarte für unser Beispiel dar.

Die Komplexität des Algorithmus zur Bestimmung der Kostenkarte ist proportional zum Quadrat der maximal direkt erreichbaren Distanz und ist konstant. Zusätzlich ist sie proportional zur der Landmarkenanzahl. Bemerkenswert, dass die Landmarkenanzahl linear zur der Größe der Karte wächst. In solchen Szenarien ist der Algorithmus sehr



(a) (b)

Abbildung D.8.: (a) Kostenkarte: Je heller der Punkt in der Karte, desto höher sind die Kosten für die Bewegung dorthin. Es ist sichtbar, dass einige Punkte nur über eine indirekte Bewegung über mehrere Landmarken erreicht werden können. (b) Wissenskarte: Je heller der Pixel desto höher ist das bereits vorhandene Wissen.

schnell. Die gute Performanz wird gebrochen, falls dem Roboter die Bewegung entlang einer Linie ohne Limit erlaubt wird. Dennoch stellt dieses kein wünschenswertes Verhalten des Roboters dar. Damit die Ergebnisse des SLAM-Algorithmus möglichst akkurat werden, soll die Position des Roboters anhand der Odometriedaten so genau wie es nur geht geschätzt werden. Dieses impliziert, dass der Roboter nur elementare und limitierte Bewegungen ausführt.

Die Wissenskarte ist so definiert, dass jeder Scan Wissen zur seiner Umgebung hinzufügt. Dafür wird der Radius r der Wissenserweiterung nach einem Scan festgelegt. Führt der Roboter einen Scan durch, wird das Wissen rund um den Roboter mit einem festen Radius r auf 1 gesetzt. Die Wissenskarte für unser Beispiel ist in der Abbildung D.8(b) zu sehen.

Abschließend soll noch die nächste Scanposition bestimmt werden. Dafür wird der Punkt gewählt, der das b_{xy} in der folgenden Gleichung maximiert:

$$b_{xy} = \frac{1}{k_{xy} + k_0} \cdot \frac{1}{c_{xy} + c_0} \tag{D.4}$$

wobei k_{xy} der Punkt in der Wissenskarte und c_{xy} der korrespondierender Punkt in der

Kostenkarte ist, k_0 und c_0 sind Konstanten und werden als Kosten- und Wissensoffset genutzt.

Die Auswahl der maximalen Bewegungsdistanz d_{max}, der Wissenserweiterungsradius k_r sowie k_0 und c_0 sind mehr oder minder arbiträr. Das Verhalten des Roboters kann durch diese Parameter maßgeblich beeinflusst werden. Die besten Ergebnisse wurden mit $d_{max} = 2.0\,m$, $k_r = 1.0\,m$, $k_0 = 1.0$ und $c_0 = 3.0$ erzielt. Dennoch, die Auswahl dieser Parameter ist von den Eigenschaften des Laserscanners, des Roboters und der Umgebung abhängig.

D.7. Evaluation

Für die Evaluation des Explorationsalgorithmus wurde eine Rotationsplattform entwickelt, die auf einem Pioneer-2-DX-Roboter montiert worden ist. Nach der Konzeptualisierung wurde die Rotationsplattform während der Diplomarbeit von Peter Breuer realisiert [Bre10]. Die Rotationsplattform besteht aus dem 2-D-Laserscanner, Schrittmotor sowie dem Microkontroller zur Kontrolle einzelner Komponenten. Der Pioneer-2DX-Roboter mit der darauf montierten Rotationsplattform ist in der Abbildung D.9 dargestellt.

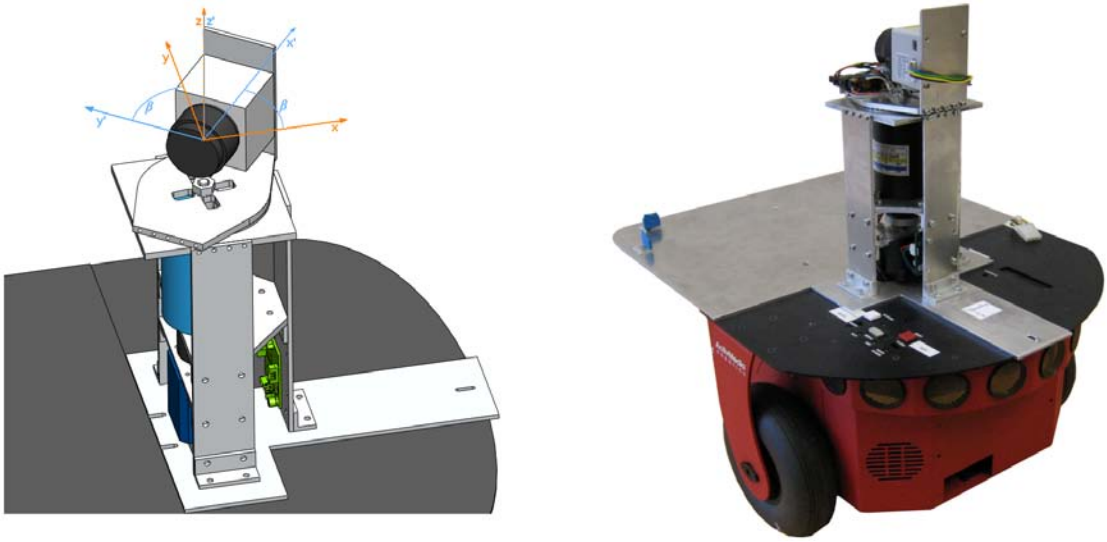


Abbildung D.9.: Die eingesetzte Plattform: Auf der linken Seite abgebildet ist die technische Zeichnung der entwickelten Explorationsplattform, rechts, die realisierte Plattform.

Der größte Vorteil des Systems sind die Möglichkeit zum permanenten Rotieren, gegeben durch ein Schleifring, sowie der Umstand, dass eine Rotation um 180° den kom-

pletten Scan der Umgebung beinhaltet.

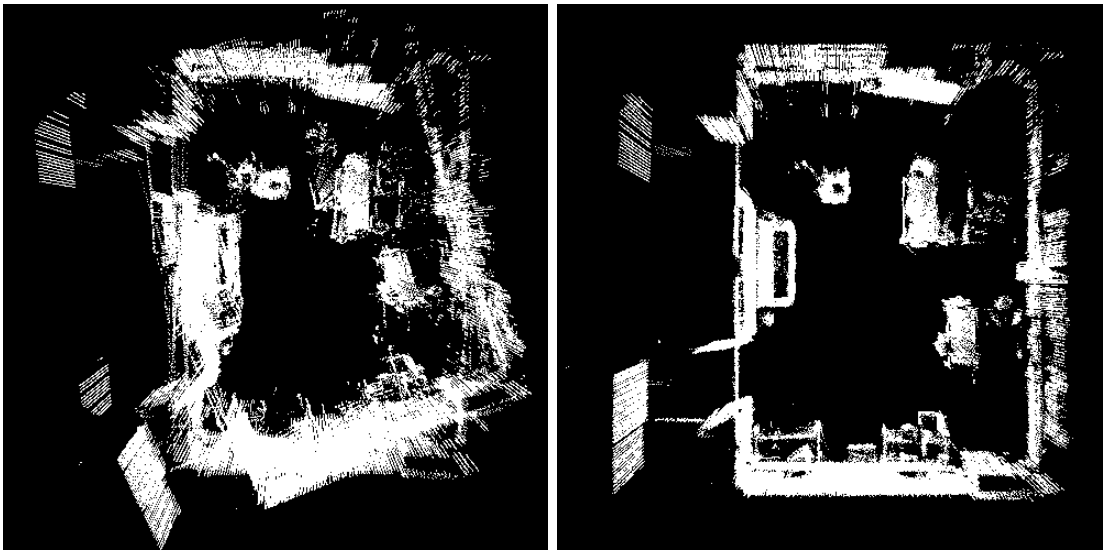


Abbildung D.10.: Ergebnis einer Karte nach 20 abgeschlossen Scanvorgängen. Das linke Bild zeigt die Karte, die durch das Zusammenfügen des Scans nur auf der Basis der Odometriewerte entstanden ist. Das rechte Bild zeigt die Ergebnisse des SLAM-Algorithmus.

Zuerst soll die Frage geklärt werden, inwieweit der Einsatz des SLAM-Algorithmus notwendig ist. Die Abbildung D.10 visualisiert die Karten mit und ohne SLAM.

Wie dem linken Bild der Abbildung D.10 entnommen werden kann, weist die Karte ohne SLAM enorme Ungenauigkeiten auf, ist von schlechter Qualität und reicht für die Exploration der Umgebung nicht aus. Dagegen weist die Karte mit SLAM-Unterstützung eine hohe Qualität auf. Die Scans passen mit höherer Genauigkeit zueinander und bilden damit eine gute Basis für die Umgebungsexploration. Ein weiterer Vorteil ist eine sehr geringerer Anzahl von falsch erkannten Hindernissen. Solche Hindernisse entstehen durch unakkurate Scanvorgänge und beschränken die Bewegungsfreiheit des Roboters. Wenn jeder neue Scan solche False-Positives in die Karte hineinbringen würde, würde diese sehr schnell unbrauchbar werden für die Exploration.

Abbildung D.11 visualisiert die endgültigen 2-D-Karten, die von dem Explorer-Programm des entwickelten Systems generiert worden sind. Dabei werden mehrere spezifische Charakteristiken des System offenbart. Erstens passen die Scans sehr gut zueinander: Die Anzahl der falsch erkannten Hindernisse sowie der Ausreißer, die durch den Laserscanner verursacht werden, ist sehr gering. Zweitens verdeutlicht die Wissenskarte, dass alle Scans an unterschiedlichen Positionen stattgefunden haben. Damit wurde immer ein Informationsgewinn gegenüber der vorherigen Situation erzielt. Die dritte Karte, die

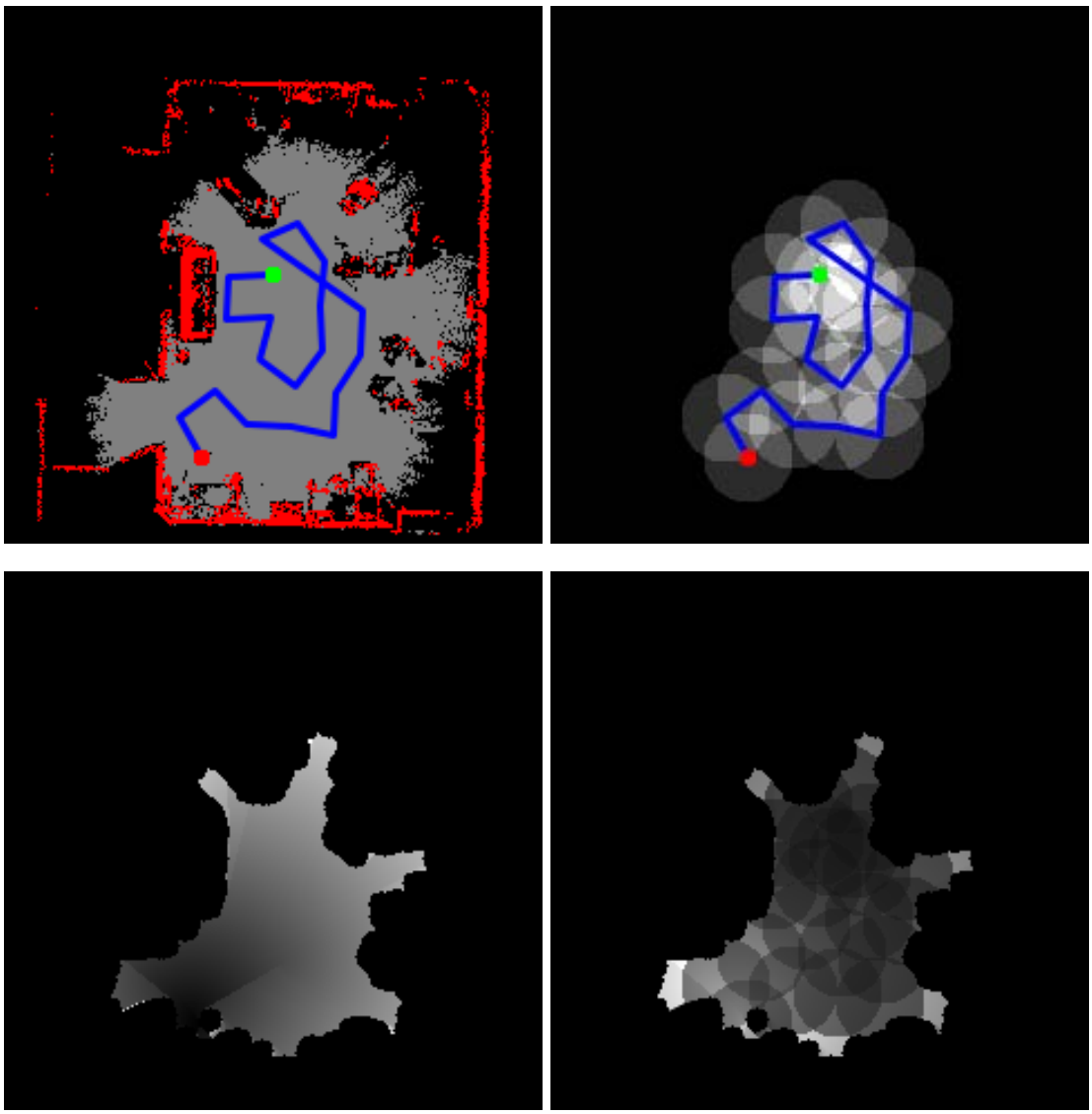


Abbildung D.11.: Explorationskarte nach den 20 Schritten. Das obere linke Bild zeigt die Flurkarte mit den eingetragenen Hindernissen, die in Rot dargestellt sind. Das obere rechte Bild visualisiert die Wissenskarte und das untere linke Bilde die Kosten, die mit dem Weg zum dedizierten Punkt der Karte assoziiert sind. Wissens- und Kostenkarte sind in dem unteren rechten Bild kombiniert, damit entsteht eine Karte, die für die zukünftigen Scanpositionen maßgeblich ist. Wobei je heller der Pixel, desto geeigneter ist die Scanposition. Die oberen Bilder zeigen zusätzlich den Pfad, der in Blau dargestellt ist, mit der grünen Start- und der roten Zielmarkierung.

diskutiert werden soll, ist die Kostenkarte. Die letzte Scanposition liegt im unteren Bereich des Bilds, aber befindet sich in dem zu explorierenden Raum. Einige Positionen im Raum können direkt und andere nur über weitere Landmarken erreicht werden. Das kann deutlich anhand der Diskontinuität in der Kostenkarte erkannt werden. Ein Vergleich der Kosten- mit der Flurkarte zeigt, dass der Roboter den Raum durch die linke Tür verlassen kann. Damit wird verdeutlicht, dass das Explorationssystem solche Bereiche, in denen der Roboter weiterfahren kann, erkennt und richtig bewertet. Abschließend zeigt die Evaluationskarte wie die Wissenskarte die Bereiche, wo das Wissen vorhanden ist oder blockiert, weil die Kostenkarte lokale Roboterbewegungen bevorzugt.

D.8. Mögliche Erweiterungen des Explorationsalgorithmus

Rückblickend bietet die verteilte Implementierung des Algorithmus einen weiteren wichtigen Vorteil. Zwar ist der Algorithmus für das Auskundschaften der unbekannten Umgebung ausgelegt, dennoch kann auch eine bekannte Umgebung in permanenten Abständen neu exploriert werden. Für weitere Informationen zum Algorithmus, dem theoretischen Hintergrund sowie Implementierungsdetails sei auf die Diplomarbeit von Gregor Michalicek [Mic10] sowie auf eine in gemeinsamen Arbeit entstandene weiterführende Publikation [MKZ11] verwiesen. Die erneute Exploration kann dadurch erreicht werden, dass in der einmal explorierten Umgebung das Wissen über die Umgebung mit der Zeit abnimmt. Dieses Konzept erinnert stark an die menschliche Fähigkeit des Vergessen. Durch die Implementierung einer temporal abhängigen Komponente kann die Anwendung in das bestehende System problemlos integriert werden. Damit veranlasst die Wissenskarte, dass auch schon explorierte Bereiche erneut angefahren werden. Damit entsteht in einer abgeschlossenen Umgebung ein optimaler Pfad, der immer wieder vom Roboter abgefahren wird. Dadurch wird die 3-D-Karte der Umgebung in permanenten Abständen aktualisiert, was der Auffindung der oben beschriebenen ROIs zugute kommt. Damit kann der Algorithmus sowohl für die Exploration der unbekannten als auch für die bekannten Umgebungen eingesetzt werden.

Die dynamischen Veränderungen in der Umgebung beeinflussen berechnete Pfade, stellen aber im Weiteren kein wesentliches Problem dar. Zusätzlich kann die explorierte Karte genutzt und immer wieder verbessert werden. Die meisten in Räumen auftretenden Hindernisse sind von statischer Natur und bleiben daher über einen längeren Zeitraum erhalten. Damit kann die Zusammenführung mehrerer Scans ein weiteres Mal verbessert und das gesamte System beschleunigt werden, dies wird auch als dynamisches Mapping bezeichnet.

D.9. Segmentierung von planaren Flächen

Nachdem der Prozess der Umgebungsexploration abgeschlossen ist, liegt eine in sich registrierte Punktwolke vor. Der nächste Schritt ist nun, die Umgebungswolke zu separieren. Damit soll die Punktwolke in voneinander unabhängige Regionen unterteilt werden, angelehnt an die Gleichung D.5. Dabei sind in erster Linie die Klassifikationen von Boden, Decke, Wänden und planaren, parallel zum Boden verlaufenden Flächen, wie zum Beispiel Tische, interessant. Da in dieser Arbeit die 3-D-Punktwolke als ein um eine Dimension erweitertes Bild erfasst werden kann, gilt nach [GW92] Folgendes:

Sei R die Gesamtregion eines Bilds und seien R_1, ..., R_n die segmentierten Teilregionen des Bilds, dann gilt:

$$R = \bigcup_{i=1}^{n} R_i, wobei\ R_i \cap R_j = \varnothing\ und\ \forall\ i, j,\ i \neq j \tag{D.5}$$

Zum Zeitpunkt der Realisierung existierten viele Methoden der Segmentierung, dennoch fanden nur drei davon innerhalb 3-D-Punktwolken am häufigsten Verwendung: nämlich RANSAC (engl. für RANdom SAmple Consensus) [FB81] sowie Region-Growing [GW02][Din07] und Hough-Transformation [Hou62]. Da in der vorliegenden Arbeit der auf dem RANSAC-basierender Algorithmus zum Einsatz kommt, wird dessen Funktionsweise später noch detailliert erläutert.

Der Region-Growing-Algorithmus ist eine einfache Bildsegmentierungsmethode, der Pixel-basiert arbeitet. Zuerst wird das Bild in elementare Zellen von vordefinierter Größe unterteilt. Dann wird mit einer dieser Zellen begonnen, dabei wird diese mit den benachbarten Zellen verglichen. Sind die Zellen homogen zueinander, werden sie zu einer Region zusammengefasst. Der Algorithmus terminiert, falls keine benachbarten homogenen Zellen vorhanden und damit alle Zellen einer Region zugeordnet sind. Des Öfteren wird der Mittelwert, angewandt auf Grauwerte oder einzelne Farbkanäle des Bilds, als Homogenitätskriterium verwendet. Abhängig von der Größe der initialen Zellen kann die Qualität beziehungsweise die Performanz der Methode gesteuert werden. Gleichzeitige Verbesserung der beiden Größen ist leider aufgrund der umgekehrten Proportionalität der Parameter zueinander nicht möglich. Die Vorteile des Algorithmus liegen in seiner Einfachheit, der korrekten Zuordnung einzelner Pixel sowie der Möglichkeit der gleichzeitigen Verwendung mehrerer Kriterien. Die Nachteile sind seine Empfindlichkeit gegenüber nicht gleichmäßig verteilter Rausch- sowie Ressourcen- und Zeitperformanz. Dennoch ist es mit der heutigen Technik denkbar, den Algorithmus zu parallelisieren, was seine Performanz massiv steigern könnte.

Dagegen ist die Grundidee der Hough-Transformation deutlich komplizierter. Der Algorithmus segmentiert einfache parametrisierbare geometrische Figuren, am Anfang nur Geraden, später wird die Methode durch andere geometrische Primitiven wie zum Beispiel Ellipse oder Kreise erweitert. Als Vorverarbeitungsschritt wird die Abbildung

erst in Schwarz-Weiß konvertiert mit anschließender Kantendetektion. Für jeden Kantenpixel des Originalbilds werden im sogenannten Hough-Raum spezifische Parameter eingetragen. Betrachten wir eine Gerade, so sind es die Geradensteigung und der y-Achsenabschnitt, die in den Dualraum übertragen werden. Abschließend wird der Hough-Raum ausgewertet, dabei werden die Häufungen des Dualraums lokalisiert und interpretiert. Die größten Nachteile dieses Verfahrens sind einerseits der enorme Speicherbedarf und die nötige Rechenkapazität sowie die oft notwendige Nachbearbeitung der Ergebnisse. Zum Beispiel gehen der Anfang und das Ende einer Kante bei der Transformation mit der Hough- Transformation für Geraden im Dualraum verloren. Da aber gerade diese Informationen in den meisten Fällen wichtig ist, müssen die Ergebnisse nachbearbeitet werden. Grundsätzlich ist der Algorithmus auf die Bearbeitung 2-D-Eingangsdaten ausgelegt, dennoch existieren mehrere Verfahren, die den Algorithmus auf die 3-D-Eingangsdaten erweitern. Ein guter Überblick über 3-D-Anwendung der Methode wird in [BELN11] vermittelt.

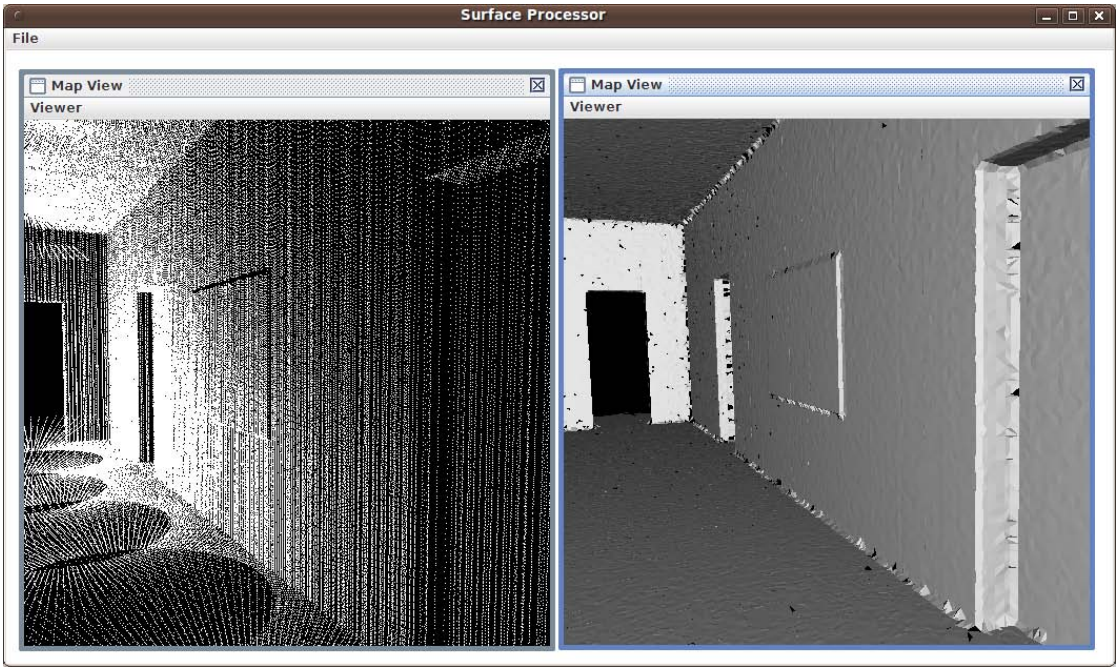


Abbildung D.12.: Ergebnisse der Umgebungserfassung in Form einer Punktwolke (links) und der darauf angewendete Ball-Pivotings-Algorithmus für die 3-D--Rekonstruktion (rechts).

Werden die bisherigen Ergebnisse zusammengefasst, wie schon oben erwähnt, entsteht eine in sich konsistente Punktwolke. Die Ergebnisse für die Punktwolke sowie der An-

wendung des Ball-Pivoting-Algorithmus [BMR+99] für die 3-D-Rekonstruktion sind in Abbildung D.12 dargestellt. Da die Größen des physikalischen Aufbaus bekannt sind, kann die Decke des Raums sofort separiert und aus der Punktwolke entfernt werden, der Fußboden, wie bereits oben erwähnt, wird in der Punktwolke nicht explizit dargestellt.

Damit die restliche Punktwolke weiter segmentiert werden kann, wird in der vorliegenden Arbeit ein Algorithmus genutzt, der auf RANSAC basiert und dessen Implementierung an die, in der Point Cloud Library (PCL [1]) integrierte Methode, basiert. Der Algorithmus ist in dem Bereich der Bildverarbeitung weit verbreitet und zeichnet sich durch seine Robustheit aus. Dabei schätzt die Methode ein Modell aus einer Messwertreihe, die durch Ausreißer und teilweise grobe Fehler gekennzeichnet sein kann. Die Robustheit wird durch die Berechnung einer bereinigten Datenmenge, einem sogenannten Consensus Set, erreicht. Einer der notwendigen Kriterien für den Einsatz von RANSAC ist ein überbestimmtes Modell, das heißt, dass mehr Daten, im vorliegendem Fall 3-D-Punkte, vorhanden sein müssen als es für die Parameterbestimmung notwendig ist. Durch die große Punktdichte ist dieses Kriterium in der vorliegender Arbeit definitiv erfüllt.

Gegenüber den bekannten Methoden, wie zum Beispiel der Methode der kleinsten Quadrate, verfolgt die RANSAC-basierte Methode einen iterativen Ansatz. Dazu wird eine kleine Menge, gerade ausreichend für die Parameterbestimmung, zufällig gewählt. Basierend auf diesem Datensatz werden unter Annahme, dass keine Ausreißer vorliegen, die Modellparameter berechnet. Danach wird das berechnete Modell mit allen vorhandenen Daten verifiziert. Diese Schritte werden mehrmals durchgeführt, dabei werden die zu einem bestimmten Modell passenden Punkte in einem sogenannten Consensus Set gespeichert. Abschließend wird aus dem größten Consensus Set mit einem traditionellen Ausgleichsverfahren die Lösung bestimmt. Die einzelne Schritte sind als Pseudo-Code im Algorithmus 4 zusammengefasst.

In dem vorliegenden Fall wird ein Modell der planaren Fläche aus einer zufällig gewählten Menge von Punkten gemäß der linearen Koordinatengleichung der Ebenendarstellung der Parameterform

$$ax + by + cz + d = 0 \tag{D.6}$$

kalkuliert. Dabei sind a, b, c, d feste und x, y, z variable Parameter. Ist die Anzahl der Punkte größer als der vordefinierte Schwellwert, werden die Punkte in einem Consensus Set gespeichert. Durch mehrmaliges Wiederholen und Vergleichen berechneter Consensus Sets wird die größtmögliche Fläche bestimmt. Die zum größten Consensus Set zugehörigen Punkte werden aus der Datenmenge entfernt, der Algorithmus wird auf die restliche Datenmenge erneut angewandt. Ist die Anzahl der verbleibenden Punkte zu niedrig oder der Abstand zwischen den Punkten zu groß, terminiert die Methode. Die übrig gebliebene Punkte werden als Ausreißer betrachtet und gelöscht.

[1] http://pointclouds.org/

Algorithm 4 The segmentation of planar surfaces

1: **procedure** THE PLANAR SURFACE SEGMENTATION
2: Extract randomly a specified set of points, enough for the model calculation.
3: Calculation of the model parameters.
4: Evaluate the calculated model with the rest of the input point cloud. If the number of points convenient with the calculated model is bigger then the seted threshold, than save all of the suitable points in a consensus set.
5: Repeat the steps 2 to 4 several times.
6: Find the maximum consensus set.
7: Calculation of the model parameters fo the maximum consensus set.
8: Remove the points of the maximum consensus set from the input point cloud.
9: If the number of the remain points is over and the distance between them under the threshold, repeat the algorithm for the rest of the input point cloud.
10: **return** all found plains (maximum consensus sets).
11: **end procedure**

Damit hängt die Qualität der Berechnung von folgenden Parametern ab:

- dem maximalen Abstand vom Punkt zum Modell, damit der Punkt noch als modellzugehörig betrachtet wird,
- der Anzahl der Iterationen,
- der Mindestgröße des Consensus Sets.

Die aufgeführten Parameter können empirisch angepasst werden. Die Nachteile des Algorithmus liegen in der nötigen Anpassung der oben genannten Parameter sowie der notwendigen Begrenzung der Iterationen. Sind die Parameter und die Anzahl der Iterationsschritten angepasst, liefert der Algorithmus schnelle und akkurate Ergebnisse.

Abbildung D.13 visualisiert die Ergebnisse der planaren Segmentierung für einen eher ungünstigen Fall: ein kleines Büro mit vielen durcheinanderliegenden Objekten, aufgenommen von unten mit dem in der Abbildung D.9 vorgestelltem System. Die Punktwolke umfasst ca. 230 000 Punkte. Dennoch wird der rechts stehende Tisch, aufgenommen mit einer sichtbar kleineren Punktdichte, richtig segmentiert. Dies ist erkennbar an den gelb gefärbten Punkten auf der Tischoberfläche.

Nachdem alle möglichen planaren Flächen bestimmt sind, kann mit der Klassifizierung begonnen werden. Dafür wird die Normalform der Ebene im dreidimensionalen Raum genutzt:

$$\vec{n} \cdot \vec{q} = \lambda, \tag{D.7}$$

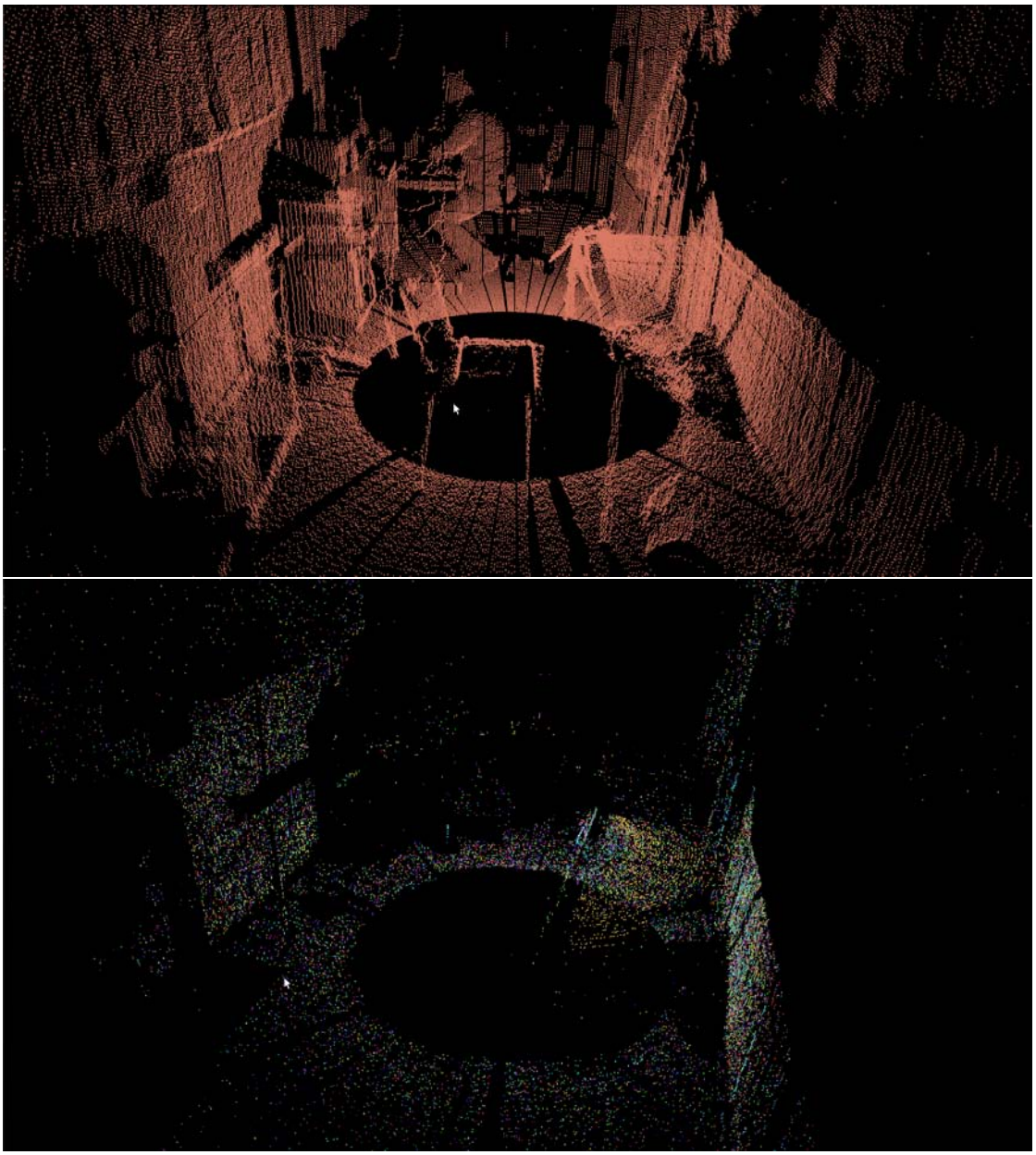


Abbildung D.13.: Oben sichtbar ein kompletter Laserscan (ca. 230 000 Punkte) einer Büroumgebung. Unten einzelne segmentierte Bereiche. Diese Abbildung verdeutlicht, dass trotz der vielen vorhandenen Objekte sowie kleinerer Punktdichte der Tisch, rechts im Bild, richtig segmentiert wird. Verdeutlicht wird dies durch die gelb gefärbten Punkte auf der Tischoberfläche

wobei $\vec{n}$ der Normalvektor der Ebene ist, λ bestimmt deren Position und der Ortsvektor $\vec{q}$ verbindet den Koordinatenursprung mit einem ihrer Punkte.

Durch den Vergleich der Ausrichtung der Normalen können die Flächen klassifiziert werden. Liegen die Normalen innerhalb eines Schwellwerts und sind damit senkrecht zum Boden und/oder Decke, werden die Flächen als Wände zusammengefasst. Sind die Normalen dagegen parallel zum Boden und/oder Decke, handelt es sich um für die vorliegende Arbeit interessante Bereiche (ROIs). Denn die Wahrscheinlichkeit, in einem Innenraum Objekte auf solchen Oberflächen zu finden, ist sehr hoch.

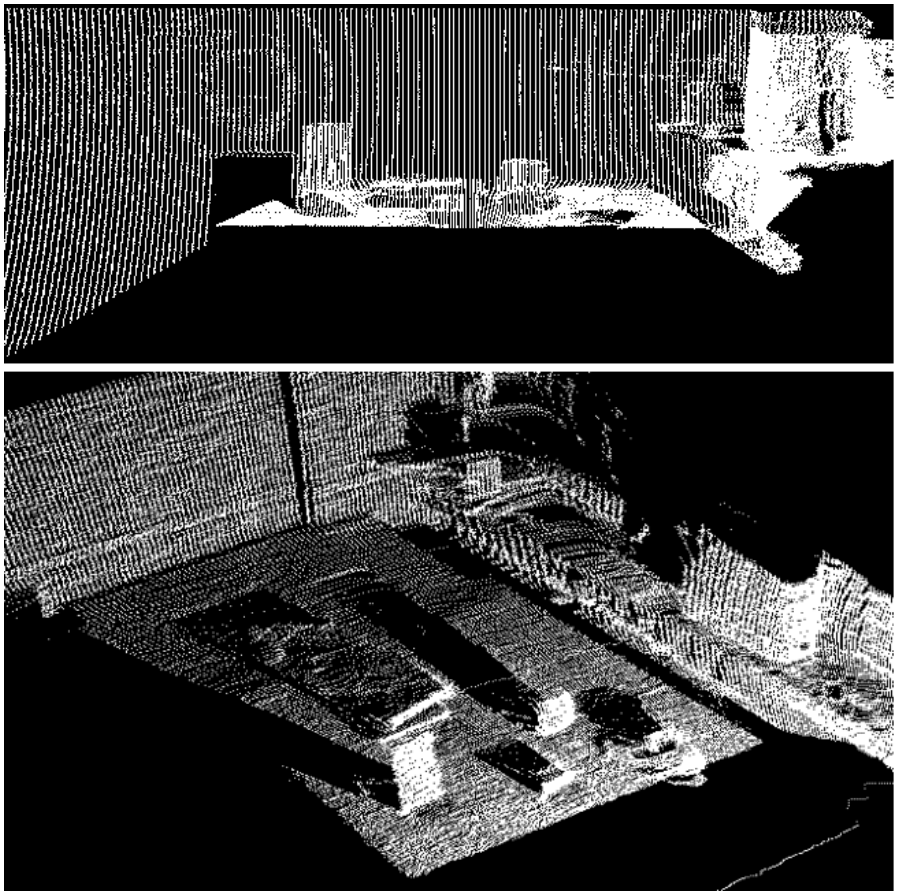


Abbildung D.14.: Mit einem bewegten 2-D-Laserscanner und einer größeren Punktdichte wahrgenommenen und zuvor segmentierte Tischszene dargestellt aus zwei Perspektiven.

Sind die ROIs berechnet, können diese von dem Roboter gezielt angesteuert werden und mit größerer Punktdichte, gegeben durch die Ausrichtung der Sensoren und dem

verkürzten Abstand zur Szene, erfasst und analysiert werden. Abbildung D.14 visualisiert aus zwei Perspektiven eine segmentierte Tischszene, die durch einen auf einer Schwenk-Neige-Einheit montierten 2-D-Laserscanner aufgenommen wurde. Dieses System wurde bereits im [KAZ09] vorgestellt.

Literaturverzeichnis

[AKJ02] S. Antani, R. Kasturi, and R. Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition 35, the journal of the pattern recognition society, pp. 945-965.*, 2002.

[Alb95] T. D. Albright. My most true mind thus makes mine eye untrue. *Trends in Neurosciences*, vol. 18:331–333, 1995.

[Alo93] Y. Aloimonos. *Active Perception.* Lawrence Erlbaum Associates, Inc, 1993.

[AMT+12] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, and S. Gedikli. Using the point cloud library for 3d object recognition and 6dof pose estimation. *Robotics & Automation Magazine, vol. September 2012*, page 12, 2012.

[APL07] H. Aziz, M. Peterson, and D. Leech. Combinatorical and computational aspects of multiple weighted voting games. *The third ACID Workshop, Warwick economic research papers, Department of Economics, University of Warwick*, 2007.

[Baj88] R. Bajcsy. Active perception. *Technical report, University of Pennsylvania, Proc. IEEE 76, 996-1005*, 1988.

[Bal91] D. H. Ballard. Animate vision. *Journal Artificial Intelligence archive Volume 48 Issue 1, Feb. 1991, Pages 57 - 86*, 1991.

[Bar83] L. W. Barsalou. Ad hoc categories. *Memory & Cognition, Springer-Verlag*, 11(3):211–227, 1983.

[BELN11] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter. The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2), 2011.

[BGG03] V. Bruce, M. A. Gergeson, and P. R. Green. *Visual Perception.* Psychology Press, 2003.

[BGM+11] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Rühr, M. Tenorth, and M. Beetz. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. *in IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2011.

[BH96] C. Brenner and M. Hahn. Object recognition using multi-sensor fusion and active exploration. *International Archives of Photogrammetry and Remote Sensing, vol. XXXI, Part B5*, 1996.

[Bha11] S. Bhattacharyya. A brief survey of color image preprocessing and segmentation techniques. *Journal of Pattern Recognition Research 1, pp. 120- 129*, 2011.

[Bie87] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review.*, Vol. 94.:115–147, 1987.

[BM92] P. J. Besl and N. D. McKay. Ieee trans. pattern anal. mach. intell. *Journal of Field Robotics*, 14(2):239–256, 1992.

[BM02] M. Bennamoun and G. J. Mamic. *Object Recognition.* Springer-Verlag London Berlin Heidelberg, 2002.

[BM06] A. C. Berg and J. Malik. Shape matching and object recognition. *J. Ponce et al. (Eds.): Toward Category-Level Object Recognition, LNCS 4170, pp. 483–507, Springer-Verlag Berlin Heidelberg*, 2006.

[BMR+99] F. Bernardini, J. Mittleman, H. Rushmeir, C. Silva, and G. Taubin. The ballpivoting algorithm for surface reconstruction. *IEEE Transaction on Visualization and Computer Graphics, 5(4), pp. 349-359*, 1999.

[Bow74] T. G. R. Bower. The evolution of sensory systems. *In R. B. MacLeod, & H. L. J. Pick (Eds.), Perception: Essays in honor of James J. Gibson, pp. 141-152.*, 1974.

[Bre10] P. Breuer. Entwicklung eines systems zur 3d-rekonstruktion der dynamischen umgebung mit einem rotierenden 2d-laserscanner. *Masterthesis at the University of Hamburg, MIN Faculty, Group TAMS*, 2010.

[Bro58] R. Brown. How should a thing be called? *Psychological Review.*, Vol. 65.:14–21, 1958.

[BT98] S. Birchfield and C. Thomasi. Depth discontinuities by pixel-to-pixel stereo. *Proceedings of the IEEE International Conference on Computer Vision, Bombay, India*, 1998.

[BTG06] H. Bay, T. Tuytelaars, and L. Van Gool. Speeded up robust features. *Proceedings of the 9th European Conference on Computer Vision*, 2006.

[BTM+12] B. Browatzki, V. Tikhanoff, G. Metta, H. H. Buelthoff, and C. Wallraven. Active object recognition on a humanoid robot. *IEEE International Conference on Robotics and Automation (ICRA), IEEE, Piscataway, NJ, USA, 2021-2028.*, 2012.

[Bur88] P. J. Burt. Smart sensing within a pyramid vision mashine. *Proceedings of the IEEE (Volume:76 , Issue: 8), pp. 1006-1015*, 1988.

[BZ06] T. Baier and J. Zhang. Reusability-based semantics for grasping evaluation in context of service robotics. *IEE International Conference on Robotics and Biomimetic, Kunming, China*, 2006.

[CBSF09] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. *IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, 2009.

[CF10] T. Cavallari and Tombari F. 3d object recognition based on correspondence grouping. *Fourth Pacific-Rim Symposium on Image and Video Technology*, 2010.

[Dau01] F. Daum. Book review on: Handbook of multisensor data fusion. *EEE Aerospace and Electronic Systems Magazine, 16(10):15–16.*, 2001.

[Din07] J.-J. Ding. The class of "time-frequency analysis and wavelet transform". *the Department of Electrical Engineering, National Taiwan University, Teipei, Taiwan*, 2007.

[EGS+12] R. Eidenberger, T. Grundmann, M. Schneider, W. Feiten, M. Fiegert, G. von Wichert, and G. Lawitzky. Scene analysis for service robots. *E. Passler et al. (Eds): Towards service robots for everyday environ., STAR 76, pp. 181-213, Springer-Verlag Berlin Heidelberg*, 2012.

[EH10] P. Eisentraut and B. Helmle. *PostgreSQL Administration.* O'Reilly, second edit., 2010.

[EKJ07] S. Ekvall, D. Kragic, and P. Jensfelt. Object detection and mapping for service robot tasks. *Journal Robotica archive, vol. 25 Issue 2, pp. 175-187, Cambridge University Press New York, NY, USA*, 2007.

[EKRZ13] L. Einig, D. Klimentjew, S. Rockel, and J. Zhang. Parallel plan execution and re-planning on a mobile robot using state machines with htn planning systems. *IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China*, 2013.

[Fau95] O. Faugeras. Stratification of 3-dimensional vision: Projective, affine and metric representations. *Journal of the Optical Society of America*, 19, 1995.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, Volume 24 Issue 6, pp. 381-395, ACM New York, NY, USA*, 1981.

[FCR04] K. Facelli, A.C.P.L.F. De Carvalho, and S.O. Rezende. Combining intelligent techniques for sensor fusion. *Applied Intelligence 20, pp. 199-213*, 2004.

[FdFCTB08] R. Fabbri, L. da Fontoura Costa, J. C. Torelli, and O. M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys (CSUR)*, 40, 2008.

[FFH+92] O. Faugeras, P. Fua, B. Hotz, R. Ma, L. Robert, M. Thonnat, and Z. Zhang. *Quantitative and Qualitative Comparison of some Area and Feature-Based Stereo Algorithms*. Wichmann, Karlsruhe, 1992.

[FSV04] D. Fofi, T. Sliwa, and Y. Voisin. A comparative survey on invisible structured light. *SPIE Electronic Imaging - Machine Vision Applications in Industrial Inspection XII*, 2004.

[FTV00] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications 12, Springer-Verlag, pp. 16-22*, 2000.

[GA11] C. Grosan and A. Abraham. Rule-based expert systems. *Intelligent Systems Reference Library Volume 17, Springer-Verlag Berlin Heidelberg*, pages 149–185, 2011.

[GH01] A. Gruen and T. S. Huang. *Calibration and Orientation of Cameras in Computer Vision*. Springer-Verlag Berlin Heidelberg, 2001.

[GIM02] M. S. Gazzaniga, R. Ivry, and G. R. Mangun. *Cognitive Neuroscience*. The Biology of the Mind. W.W. Norton, 2nd Edition, 2002.

[GL98] P. Gvozdjak and Z.-N. Li. From nomad to explorer: active object recognition on mobile robots. *Pattern Recognition vol. 31, no. 6, 773-790*, 1998.

[Gol07] E. B. Goldstein. *Wahrnehmungspsychologie.* Hans Irtel Spektrum-Akademischer Vlg., 2007.

[GS99] T. Gevers and A. W. M. Smeulders. Color-based object recognition. *Pattern Recognition 32, the journal of the pattern recognition society, pp. 453 – 464*, 1999.

[GW92] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley, 1992.

[GW02] R. C. Gonzalez and R. E. Woods. *Digital Image Processing 2nd Edition.* 2002.

[GWAH13] M. Guenther, T. Wiemann, S. Albrecht, and J. Hertzberg. Building semantic object maps from sparse and noisy 3d data. *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2013.

[HC11] A. Harris and J. M. Conrad. Survey of popular robotics simulators, frameworks, and toolkits. *In IEEE Southeastcon, pp. 243-249*, 2011.

[HKS97] T. C. H. Heng, Y. Kuno, and Y. Shirai. Active sensor fusion for collision avoidance. *in Proc of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 3, pp. 1244–1249*, 1997.

[HNL12] J. Hertzberg, A. Nüchtern, and K. Lingemann. *Mobile Roboter.* Springer Verlag Berlin Heidelberg, 2012.

[Hor84] B. K. P. Horn. Extended gaussian image. *Proc. of the IEEE, 72:1671-1686*, 1984.

[Hor06] A. Hornberg. *Handbook of Machine Vision.* WILEY-VCH Verlag GmbH & Co. KGaA, 2006.

[Hou62] P. Hough. Method and means for recognizing comples patterns. *In US Patent.*, 1962.

[Hub95] D. Hubel. *Eye, Brain and Vision.* W. H. Freeman and Company, New York., 1995.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[JFS+10] S. Jacobs, A. Ferrein, S. Schiffer, D. Beck, and G. Lakemeyer. Robust collision avoidance in unknown domestic environments. *Springer Berlin / Heidelberg, vol. 5949, pp. 116-127*, 2010.

[JKS95] R. Jain, R. Kastur, and B. G. Schunk. *Mashine Vision*. McGraw-Hill, Inc., 1995.

[Kan81] I. Kant. *Kritik der reinen Vernunft*. 1781.

[KAZ09] D. Klimentjew, M. Arli, and J. Zhang. 3d scene reconstruction based on a moving 2d laser range finder for service-robots. *In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, Guangxi, China, December 18-22, pp. 1129-1134*, 2009.

[KBA+14] L. Kunze, C. B., M. Alberti, A. Tippur, J. Folkesson, P. Jensfelt, and N. Hawes. Combining top-down spatial reasoning and bottom-up object class recognition for scene understanding. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 2910–2915, 2014.

[KDH14] L. Kunze, K. K. Doreswamy, and N. Hawes. Using qualitative spatial relations for indirect object search. *IEEE International Conference on Robotics and Automation, ICRA, Hong Kong, China*, pages 163–168, 2014.

[KF07] F. Krolupper and J. Flusser. Polygonal shape description for recognition of partially occluded objects. *ScienceDirect, Pattern Recognition Letters 28 1002–1011, Elsevier B.V.*, 2007.

[KFBZ10] D. Klimentjew, N. E. Flick, T. Bosselmann, and J. Zhang. 3d hypergraph-oriented air flow analysis based on ptv. *In Proceedings of the IEEE International Conference on Information and Automation (ICIA), China, pp. 424 - 429*, 2010.

[KFBZ11] D. Klimentjew, N. E. Flick, T. Bosselmann, and J. Zhang. Hypergraph-oriented 3d reconstruction, interpretation and analysis of air flows. *International Journal of Mechatronics and Automation (IJMA), Vol. 1, No. 1, pp. 9-18*, 2011.

[KH11] S. Kean and J. Hall. *Meet the Kinect: An Introduction to Programming Natural User Interface*. aPress, United States, 2011.

[KHZ10] D. Klimentjew, N. Hendrich, and J. Zhang. Multi sensor fusion of camera and 3d laser range finder for object recognition. *In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI) Fort Douglas, University of Utah, Salt Lake City, USA*, pages 236–241, 2010.

[KI93] S. B. Kang and K. Ikeuchi. The complex egi: A new representation for 3d pose determenation. *IEEE Trans. Pattern Anal. and Mach. Intell., 15(7): 707-721*, 1993.

[KKS96] R. Klette, A. Koschan, and K. Schluens. *Computer Vision.* Friedrich Vieweg & Sohn Verlaggesellschaft mbH, 1996.

[KMP+11] A. Kanezaki, Z.-C. Marton, D. Pangercic, T. Harada, Y. Kuniyoshi, and M. Beetz. Voxelized shape and color histograms for rgb-d. *IROS Workshop on Active Semantic Perception and Object Search in the Real World*, 2011.

[KOY+05] H. Kawata, A. Ohya, S. Yuta, W. Santosh, and T. Mori. Development of ultra-small lightweitght optical range sensor system. *IEEE Intelligent Robots and Systems (IROS)*, 2005.

[KRT06] A. Klausner, B. Rinner, and A. Tengg. I-sense: Intelligent embedded multi-sensor fusion. *In Proceedings of the 4th IEEE International Workshop on Intelligent Solutions in Embedded Systems (WISES).*, 2006.

[KRZ12] D. Klimentjew, S. Rockel, and J. Zhang. Towards scene analysis based on multi-sensor fusion, active perception and mixed reality in mobile robotics. *Proceedings of the IEEE First International Conference on Cognitive Systems and Information Processing (CSIP)*, 2012.

[KRZ13] D. Klimentjew, S. Rockel, and J. Zhang. *Active Scene Analysis Based on Multi-Sensor Fusion and Mixed Reality on Mobile Systems.* Advances in Intelligent Systems and Computing, Vol. 215, Chapter 69, pp. 795-810, 2013.

[KSJZ09a] D. Klimentjew, A. Stroh, S. Jockel, and J. Zhang. Real-time 3d environment perception: An application for small humanoid robots. *In Proceedings of the IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO)*, pages 354–359, 2009.

[KSJZ09b] D. Klimentjew, A. Stroh, S. Jockel, and J. Zhang. Real-time 3d environment perception: An application for small humanoid robots. *In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Bangkok, Thailand*, pages 354–359, 2009.

[KSZZ08] D. Klimentjew, A. Stroh, H. Zhang, and J. Zhang. 3d real-time stereo reconstruction for small humanoid robots. *In Proceedings of the Proceedings of the 3rd CAS Symposium on Robotics and Manufacturing Technology*, 2008.

[Kun04] L. I. Kuncheva. Combining pattern classifiers: Methods and algorithms. *Wiley Inter-science*, 2004.

[KYM05] H. Kawata, S. Yuta, and T. Mori. Design and realization of 2-dimensional optical range sensor for environment recognition in mobile robots. *Journal of Robotics and Mechatronics Vol.17 No.2*, 2005.

[KZ11] D. Klimentjew and J. Zhang. Adaptive sensor-fusion of depth and color information for cognitive robotics. *In Proceedings of the IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO)*, pages 957–962, 2011.

[Lak87] G. Lakoff. *Women, Fire and dangerous Things. ress.* Chicago, The University of Chocago Press., 1987.

[LE07] K.-H. Lee and R. Ehsani. Comparison of two 2d laser scanners for sensing object distances, shapes, and surface patterns. *computers and electronics in agriculture 60(2008), pp. 250–262, 2007 Elsevier B.V.*, 2007.

[Lee78] D. Lee. The functions of vision. *Modes of Perceiving and Processing Information, H. L. Pick Jr. and E. Saltzman, Eds. New York Wilev, vol. 1, pp. 73-90.*, 1978.

[LH91] R.J. Linn and D.L. Hall. A survey of multi-sensor data fusion systems. *n Proceedings of the SPIE - The International Society for Optical Engineering, volume 1470, pages 13–29, Orlando.*, 1991.

[LH98] J. Llinas and D. L. Hall. An introduction to multi-sensor data fusion. *In Proceedings of the IEEE International Symposium on Circuits and Systems, volume 6, pages 537–540.*, 1998.

[LK98] S. J. Lederman and R. L. Klatzky. The hand as a perceptual system. *The Pychobiology of the Hand. London, Mac Keith Press.*, pages 16–35, 1998.

[LL86] J. M. Loomis and S. J. Lederman. *Tactual perception.*, volume Vol. II: Chapt. 31. Handbook of Perception and Human Performance. New York, John Wiley & Sons., 1986.

[Lon98] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition, Volume 31, pp. 983 - 1001*, 1998.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[LSN08] Q.V. Le, A. Saxena, and A.Y. Ng. Active perception: Interactive manipulation for improving object detection. *Standford University Journal (2008)*, 2008.

[LSP05] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 27, No. 8, pp. 1265-1278*, 2005.

[LSS13] M. Lutz, D. Stampfer, and C. Schlegel. Probabilistic object recognition and pose estimation by fusing multiple algorithms. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4229–4234, 2013.

[LTP09] L. C. Lulio, M. L. Tronco, and A. J. V. Porto. Jseg-based image segmentation in computer vision for agricultural mobile robot navigation. *Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on*, 2009.

[Luh00] T. Luhmann. *Nachbereichsphotogrammetrie*. Wichmann, Heidelberg, 2000.

[Mar78] L. E. Marks. *The unity of the senses : interrelations among the modalities.* New York : Academic Press, 1978.

[Mar82] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Henry Holt and Co., Inc., New York, NY, USA, 1982.

[MBBM+14] Z.-C. Marton, F. Balint-Benczedi, O. M. Mozos, N. Blodow, A. Kanezaki, L. C. Goron, D. Pangercic, and M. Beetz. Part-based geometric categorization and object reconstruction in cluttered table-top scenes. *Journal of Intelligent & Robotic Systems*, pages 1–22, 2014.

[Mer99] B. Mertsching. Aktives sehen – eine kurze einführung. *Kuenstliche Intelligenz, arenDTaP Verlag, Bremen*, 1/99:5–6, 1999.

[Mic10] G. Michalicek. Development of a 3d simultaneous localization and mapping exploration system. *Diplomarbeit, Group TAMS, Department Informatics, MIN Faculty, University of Hamburg*, 2010.

[MKJ08] Y. Mingqiang, K. Kidiyo, and R. Joseph. A survey of shape feature extraction techniques. *Pattern Recognition, Peng-Yeng Yin (Ed.), pp. 43-90*, 2008.

[MKZ11] G. Michalicek, D. Klimentjew, and J. Zhang. A 3d simultaneous localization and mapping exploration system. *In Proceedings of the IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO)*, pages 1059–1065, 2011.

[ML89] F. C. Mueller-Lyer. Optische urtheilstäuschungen. *Archiv für Physiologie Supplement-Band. pp. 263–270, University of Illinois Press*, 1889.

[MN78] D. Marr and H. K. Nishihara. *Representation and recognition of the spatial organization of three-dimensional shapes.*, volume Vol. 200. Proceedings of the Royal Society of London: Biological Sciences., 1978.

[MNE00] A. Maki, P. Nordlund, and J.-O. Eklundh. Attentional scene segmentation: integrating depth and motion. *Comput. Vision Image Understanding. v78 i3. pp. 351-373*, 2000.

[MP02] J. Muesseler and W. Prinz. *Allgemeine Psychologie.* Spektrum-Akademischer Vlg., 2002.

[MSDC10] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury. A survey of decision fusion and feature fusion strategies for pattern classification. Technical report, 2010.

[MTM09] R. Madhavan, E. Tunstel, and E. Messina, editors. *Performance Evaluation and Benchmarking of Intelligent Systems.* Springer, 2009.

[Mur96] R. R. Murphy. Biological and cognitive foundations of intelligent sensor fusion. *IEEE Transactions on Systems, Man, and Cybernetics*, 26:42–51, 1996.

[Mur00] R. R. Murphy. *Introduction to AI Robotics.* A Bradford Book, MIT Press, Cambridge, Massachusetts, London, England, 2000.

[MWLS11] D. Meger, C. Wojek, J. J. Little, and B. Schiele. Explicit occlusion reasoning for 3d object detection. *BMVC*, pages 1–11, 2011.

[NBAK05] M. J. Naumer, C. Bledowski, C. F. Altmann, and J. Kaiser. Vom neuronalen einzelfahrschein zur kortikalen netzkarte. audio-visuelle objekterkennung in der großhirnrinde. 2005.

[NK03] M. Novotni and R. Klein. 3-d zernike descriptors for content based shape retrieval. *Proc. of the 8th ACM symposium on Solid modeling and applications (SM'03), pp. 216-225, New York , USA, ACM Press.*, 2003.

[NLHS07] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6d SLAM - 3d mapping outdoor environments. *J. Field Robotics*, 24(8-9):699–722, 2007.

[NP99] L. Nordmann and H. Pham. Weighted voting systems. *IEEE Transactions On Raliability, vol. 48, No. 1*, 1999.

[Piz01] Z. Pizlo. Perception viewed as an inverse problem. *Vision Research*, 41:3145–3164, 2001.

[PS78] H. L. Pick and E. Saltzmann. *Modes of perceiving an processing information.* In H.L. Pick & Saltzmann, E. (Hrsg.), 1978.

[RM75] E. Rosch and C. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology.*, Vol. 7.:573–605, 1975.

[RMGJ75] E. Rosch, C. Mervis, W. Gray, and D. Johnson. Basic objects in natural categories. *Cognitive Psychology.*, Vol. 8.:382–439, 1975.

[Sch05] O. Schreer. *Stereoanalyse und Bildsynthese.* Springer-Verlag Berlin Heidelberg, 2005.

[SF68] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. *In Pattern Classification and Scene Analysis, R. Duda and P. Hart, John Wiley and Sons*, 1968.

[SHB98] M. Sonka, V. Hlavac, and R. Boyle. Image processing, analysis, and machine vision. *Cengage Learning Services*, 1998.

[SLC90] P. H. Schiller, N. K. Logothetis, and E. R. Charles. Functions of the coloropponent and broad-band channels of the visual system. *Nature*, 343:68–70, 1990.

[SM93] B. E. Stein and M. A. Meredith. *The merging of the senses.* MIT Press, Cambridge, Mass, 1993. B. E. Stein and M. A. Meredithill.

[SM00] S. Schmalz and B. Mertsching. Object recognition with structural descriptions and deformable models. *Neurocomputing, Volume 31, Issues 1-4, pp. 143-151*, 2000.

[Squ92] L. Squire. *Encyclopedia of Learning and Memory. Company.* New York, Macmillan Publishing Company., 1992.

[ST94] T. Saito and J. Toriwaki. New algorithms for euclidean distance transformations of an n-dimensional digitized picture with applications. *Pattern Recognition*, 27:1551–1565, 1994.

[Sta09] C. Stachniss. *Robotic mapping and exploration.* Springer Verlag, 2009.

[Sto93] H. J. Stoering. *Kleine Weltgeschichte der Philosophie.* Fischer Taschenbuch Verlag GmbH Frankfurt a. M., 1993.

[SZ01] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. *Proc. Int'l Conf. Computer Vision, vol 2, pp. 636-643*, 2001.

[TBF06] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* The MIT Press, 2006.

[TBFP09] P. Tokekar, V. Bhatawadekar, D. Fehr, and N. Papanikolopoulos. Experiments in object reconstruction using a robot-mounted laser range-finder. *7th Mediterranean Conference on Control & Automation Makedonia Palace, Thessaloniki, Greece, June 24 - 26*, 2009.

[Tsa87] R. Tsai. A versatile camera calibration techniaue for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE JOURNAL OF ROBOTICS AND AUTOMATION*, 21, 1987.

[Tso92] J. K. Tsotsos. On the relative complexity of active vs. passive visual search. *Int. J. Comput. Vision 7, 127-141*, 1992.

[uHGM07] D. Schneider und H.-G. Maas. Integrated bundle adjustment with variance component estimation – fusion of terrestrial laser scanner data, panaramic and central perspective image data. *IAPRS Volume XXXVI, Part 3 W52*, 2007.

[vdGS05] M. van de Giessen and J. Schmidhuber. Fast color-based object recognition independent of position and orientation. *W. Dutch et al. (Eds.): ICANN, LNCS 3696, pp. 469-474, Springer-Verlag, Berlin, Heidelberg*, 2005.

[VHH+11] J. Velez, G. Hemann, A. S. Huang, I. Posner, and N. Roy. Planning to perceive: exploiting mobility for robust object detection. *Proceedings of the 21st International Conference on Automated Planning and Scheduling*, 2011.

[Vin90] L. Vincent. Algorithmes morphologiques a base de files d'attente et de lacets extension aux graphes. *l'Ecole Nationale Supérieure des Mines de Paris, These pour obtenir le titre de Docteur en Morphologie Mathematique*, 1990.

[Vin91] L. Vincent. Efficient computation of various types of skeletons. *Proceedings of SPIE*, 1445, 1991.

[Wit03] L. Wittgenstein. *Philosophische Untersuchungen.* Frankfurt, Suhrkamp., 2003.

[WLS14] Chenxia Wu, Ian Lenz, and Ashutosh Saxena. Hierarchical semantic labeling for task-relevant rgb-d perception. *Robotics: Science and Systems (RSS)*, 2014.

[WYP05] Y. Wang, J. Yang, and Ni. Peng. Unsupervised color-texture segmentation based on soft criterion with adaptive mean-shift clustering. *Pattern Recognition Letters*, 27, 2005.

[XTZ13] L. Xie, Q. Tian, and B. Zhang. Feature normalization for part-based image classifikation. *IEEE International Conference on Image Processing, Melbourne, Australia*, 2013.

[ZC01] C. Zhang and T. Chen. Efficient feature extraction for 2d/3d objects in mesh representation. *IEEE International conference on Image Processing (ICIP)*, 2001.

[ZdFF13] L. Zhang, M. J. da Fonseca, and A. Ferreira. A survey on 3d shape descriptors. *Computer Graphics, Imaging and Visualization (CGIV), 10th International Conference*, 2013.

[Zha98] Z. Zhang. A flexible new technique for calibration. *Microsoft Research, Technical Report MSR-TR-98-71, Microsoft Research, Microsoft Corporation*, 1998.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000.

[Zha12] J. Zhang. Introduction to robotics. *Lectures at the University of Hamburg, MIN Faculty, TAMS Group*, 2012.

[ZPBB11] C.-M. Zoltan, D. Pangercic, N. Blodow, and M. Beetz. Combined 2D-3D Categorization and Classification for Multimodal Perception Systems. *The International Journal of Robotics Research*, 30(11):1378–1402, September 2011.

[ZRP+13] L. Zhang, S. Rockel, F. Pecora, L. Hotz, Z. Lu, D. Klimentjew, and J. Zhang. Evaluation metrics for an experience-based mobile artificial cognitive system. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Workshop on Metrics of Embodied Learning Processes in Robots and Animals, Tokyo, Japan*, 2013.

[ZS10] S. Zhang and M. Sridharan. Vision-based scene analysis on mobile robots using layered pomdps. *ICAPS POMDP Practitioners Workshop, Toronto, Canada*, 2010.

[ZS12] S. Zhang and M. Sridharan. Active visual sensing and collaboration on mobile robots using hierarchical pomdps. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 2012.

Danksagung

An dieser Stelle möchte ich mich bei einigen Menschen bedanken, die an der Entstehung dieser Arbeit maßgeblich beteiligt waren.

Prof. Dr. Jianwei Zhang danke ich für das entgegengebrachte Vertrauen und die Unterstützung, die mich bereits während meines Studiums begleitete. Danke Jianwei!

Prof. Dr. Joachim Hertzberg danke ich dafür, dass er nicht nur die Zweitkorrektur übernahm. Joachim, mit deinen Vorschlägen, deinen Ideen und deiner konstruktiven Kritik hast Du es immer geschafft mich zu motivieren und erneut für das Thema meiner Dissertation zu begeistern. Danke!

Großes Dank an alle TAMSler, besonders an Tatjana (Lu) Tetsis, Bernd Schütz, Norman Hendrich, Andreas Mäder, Sascha Jockel, Hannes Bistry, Manfred Grove, Vlad Ciobanu, Lasse Einig, Wiebke Eggers, Martin Noeske, Florens Wasserfall, Markus Hüser, Tim Baer-Löwenstein, Daniel Westhoff. Es tut mir Leid, falls ich jemand vergessen habe. Bei einem meinen ehemaligen Kollegen, der über diese Zeit ein guter Freund wurde, möchte ich mich besonderes bedanken, Sebastian Rockel. Danke für die wundervolle und produktive Zusammenarbeit, ich hoffe, dass wir auch in der Zukunft an einigen Projekten gemeinsamen arbeiten werden.

Meinen Eltern danke ich für den permanenten Zuspruch und die Ermutigung während meiner Dissertation. Meiner Schwiegermutter gehört besonderer Dank für ihre ständige Hilfsbereitschaft und „Babysitting“unter allen Umständen. Ich bedanke mich bei Hans-Jürgen Warzecha für seine Korrekturen und konzeptionelle Vorschläge.

Ich danke allen unseren Freunden, die einerseits mich abgelenkt und damit den notwendigen Abstand zur vorliegenden Dissertation schufen. Andererseits sich aber permanent nach dem Stand der Arbeit erkundigten und mir nahe legten schnellstmöglich fertig zu werden. Ja, ich schulde euch eine Party!

Dennoch mein größter Dank geht an meiner bezaubernde Frau Svetlana, die mich nicht nur über die Tiefen und Höhen des Lebens begleitet, aufbaut und motiviert. Sonder auch mein größter Zusprecher und schlimmster Kritiker wurde. Danke für alles, besonderes aber für unsere Söhne, ein Geschenk, das nicht mehr übertroffen werden kann!

Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, 15.03.2015

_____

(Denis Klimentjew)