

Alexander Kasper

# Szenen- und Objektmodellierung für Serviceroboter

Generierung von Hintergrundwissen  
für Perzeption und Manipulation in  
Alltagsumgebungen

Karlsruher Institut für Technologie



# **Szenen- und Objektmodellierung für Serviceroboter**

zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik  
des Karlsruher Instituts für Technologie (KIT)

**genehmigte**

**Dissertation**

von

**Alexander Kasper**

aus Bad Säckingen

Tag der mündlichen Prüfung: 15. April 2013

Erster Gutachter: Prof. Dr.-Ing. Rüdiger Dillmann

Zweiter Gutachter: Prof. Dr.-Ing. Carsten Dachsbacher



---

## Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Anthropomatik des Karlsruher Instituts für Technologie (KIT). Herrn Prof. Dr.-Ing. Rüdiger Dillmann danke ich besonders für die Anregung zu dieser Arbeit, die wissenschaftliche Förderung, die stets vorhandene Diskussionsbereitschaft und für die Übernahme des Hauptreferates. Für die freundliche Übernahme des Korreferates gebührt mein ganz besonderer Dank Herrn Prof. Dr.-Ing. Carsten Dachsbacher vom Institut für Betriebs- und Dialogsysteme.

Die Entstehung und vor allem die Vollendung dieser Arbeit wäre ohne die Hilfe einer Vielzahl an Menschen sicher nicht möglich gewesen und ich möchte die Gelegenheit nutzen ihnen an dieser Stelle zu danken.

An erster Stelle stehen hier natürlich meine Eltern Elisabeth und Walter Kasper, die es mir durch ihre Unterstützung in allen Lebenslagen überhaupt erst möglich gemacht haben diesen Weg einzuschlagen und auch zu Ende zu gehen. Sie haben mir vorgelebt wie man Angefangenes er-

folgreich zu Ende bringt und mich stets bestärkt meine eigenen Ziele zu verfolgen - dafür vielen, vielen Dank!

Meine Partnerin Inga Boie hat zum Gelingen dieser Arbeit ebenfalls maßgeblich beigetragen. Sie hat die Höhen schöner gemacht und die Tiefen gelindert, vor allem aber meinen Horizont erweitert und mir stets das Gefühl gegeben das alles schaffen zu können. Ich danke ihr von ganzem Herzen dafür, dass sie mir in dieser Zeit immer zur Seite stand und viel Geduld mit mir hatte!

Meinem Bruder Christian Kasper möchte ich an dieser Stelle ebenfalls danken, er hat das Neuland Promotion als erster betreten und war stets mit kundigem Rat zur Stelle.

Ganz besonderen Dank möchte ich auch Peter Steinhaus aussprechen, der mich an die Robotik herangeführt, mein Promotionsvorhaben stets mit vollem Einsatz gefördert hat und Kollege, Freund und Mentor in einem war und ist. Ich danke meiner Vorgängerin und Kollegin Regine Becher, die mich als Student bereits in das Thema eingeführt hat und durch ihre Vorarbeiten eine ausgezeichnete Grundlage für meine Promotion geschaffen hat. Weiterhin möchte ich meinem ehemaligen Kollegen und guten Freund Tilo Gockel danken, der stets ein offenes Ohr für meine Sorgen und Probleme hatte und der mich immer wieder in hoch interessante Nebenprojekte verstrickt hat. Schließlich will ich meinen Kollegen Martin Lösch, Sven Schmidt-Rohr, Tobias Gindele, Rainer Jäkel, Sebastian Brechtel und Pascal Meißner für die gute Zusammenarbeit und die familiäre Atmosphäre danken. Die gemeinsamen Mittagspausen, der gute Espresso und die damit verbundenen angeregten Diskussionen haben mir stets Freude bereitet!

Ohne die Mitarbeit von engagierten Studenten wäre eine Promotion und die Bearbeitung von Forschungsprojekten natürlich nicht möglich und so gilt mein Dank ebenfalls: Stefan Kästle, Björn Schelker, Henning Renartz, Christian Wischniewski, Marius Elvert, Timo Schmidt, Qingqian Liu, Daniel Weckert, Martin Tillmann, Philipp Hassinger, Roman Prutkin, Yann Krehl und Alexey Kozlov.

Karlsruhe, im Mai 2013

*Alexander Kasper*





---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	15
1.1	Motivation .....	17
1.2	Zielsetzung .....	18
1.3	Beitrag .....	19
1.4	Überblick .....	20
<b>2</b>	<b>Stand der Forschung</b> .....	23
2.1	Sensorgestützte Erfassung und Modellierung von Objekten	24
2.1.1	Modelle im Bereich Objekterkennung und -lokalisierung .....	24
2.1.2	Greifplanung .....	30
2.1.3	Objektdigitalisierung .....	33
2.1.4	Fazit .....	49
2.2	Repräsentation und Modellierung von Szenen .....	50
2.2.1	Objekterkennung durch Kontext .....	52
2.2.2	Szenenerkennung durch globale Merkmale .....	54
2.2.3	Objektsuche durch Szenenkontext .....	57

10	Inhaltsverzeichnis	
	2.2.4	Semantische Karten zur Szenenrepräsentation . . . . 60
	2.2.5	Fazit . . . . . 62
<b>3</b>	<b>Grundlagen zur Abstandsmessung</b>	. . . . . 65
	3.1	Passive Verfahren zur Abstandsmessung . . . . . 66
	3.1.1	Kamerakalibrierung . . . . . 66
	3.1.2	Triangulierung . . . . . 70
	3.2	Aktive Verfahren zur Abstandsmessung . . . . . 73
	3.2.1	Laufzeitverfahren . . . . . 73
	3.2.2	Triangulationsverfahren . . . . . 75
	3.3	Zusammenfassung . . . . . 84
<b>4</b>	<b>Konzept</b>	. . . . . 85
	4.1	Motivation . . . . . 85
	4.2	Überblick . . . . . 86
	4.2.1	Objektmodelle . . . . . 86
	4.2.2	Szenenmodelle . . . . . 88
	4.3	Zusammenfassung . . . . . 90
<b>5</b>	<b>Sensorgestützte Modellierung von Einzelobjekten</b>	. . . . . 91
	5.1	Einleitung . . . . . 91
	5.2	Problemstellung . . . . . 93
	5.3	Lösungsansatz . . . . . 94
	5.4	Sensoraufbau . . . . . 95
	5.4.1	Mechanischer Aufbau . . . . . 97
	5.4.2	Sensorik . . . . . 100
	5.5	Kalibrierung . . . . . 104
	5.6	Datenaufnahme . . . . . 113
	5.6.1	3D Information . . . . . 114

5.6.2	Objektansichten .....	115
5.7	Software und Nachbearbeitung .....	119
5.7.1	Aufbereitung Dreiecksnetze .....	120
5.7.2	Texturierung .....	123
5.8	Objektdatenbank .....	131
5.8.1	Webdatenbank .....	131
5.9	Zusammenfassung .....	133
<b>6</b>	<b>Modellierung von Szenen auf Basis von räumlichen</b>	
	<b>Relationen</b> .....	135
6.1	Einleitung .....	135
6.2	Problemstellung .....	137
6.3	Lösungsansatz .....	138
6.4	Datenakquise .....	140
6.5	Annotierung .....	143
6.6	Relationen .....	146
6.6.1	„Ist Auf“-Relation .....	148
6.6.2	„Ist Neben“-Relation .....	153
6.7	Statistische Daten .....	157
6.8	Auswertung und mögliche Anwendung .....	158
6.9	Zusammenfassung .....	163
<b>7</b>	<b>Implementierte Softwarewerkzeuge</b> .....	165
7.1	Raptor (Rapid Textured Object Generator) - Applikation zur Aufnahme der 3D-Daten .....	165
7.2	RaptorTools - Applikation zur Aufnahme der Bilddaten ..	167
7.3	TextureMapping - Applikation zur automatischen Texturerzeugung .....	169

12	Inhaltsverzeichnis	
7.4	AnnotationLib - Bibliothek für die Relationsberechnung ..	170
7.5	OVISEAnnotation - Applikation zur Annotierung von 3D-Szenen .....	171
7.6	Zusammenfassung .....	173
<b>8</b>	<b>Experimente, Ergebnisse und Bewertung .....</b>	<b>175</b>
8.1	Evaluierung der Kalibrierung des Objektmodellierungsceters .....	175
8.1.1	Methodik .....	175
8.1.2	Ergebnisse .....	177
8.2	Ergebnisse sensorgestützte Modellierung von Einzelobjekten .....	179
8.2.1	Beispielobjekte .....	180
8.2.2	Nutzung der generierten Modelle .....	184
8.2.3	Verbreitung durch Webdatenbank .....	187
8.3	Ergebnisse Szenenmodellierung auf Basis von räumlichen Relationen .....	189
8.3.1	Evaluierung der Relation „Ist neben“ mittels Benutzerstudie .....	189
8.3.2	Datensatz .....	194
8.3.3	Auswertung .....	194
8.3.4	Schätzung der Klassenzugehörigkeit .....	202
8.4	Fazit .....	208
8.4.1	Objektmodelle .....	208
8.4.2	Szenenmodellierung .....	210
<b>9</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>215</b>
9.1	Ergebnisse der Arbeit und Erkenntnisse .....	215

9.2 Ausblick ..... 218

**A Konstruktion Modellierungscenter ..... 221**

**B Verwendete Softwareframeworks ..... 227**

B.1 Pointcloud Library (PCL) ..... 227

B.2 Rapidform.DLL SDK ..... 228

B.3 Object Oriented Rendering Engine (Ogre) ..... 229

B.4 wxWidgets ..... 229

B.5 Ogre Virtual Scene Environment (OVISE) ..... 230

B.6 Intergrating Vision Toolkit (IVT) ..... 231

B.7 Boost ..... 231

B.8 FreeImage ..... 232

**Literaturverzeichnis ..... 233**



## Einleitung

In der heutigen Zeit, die durch wirtschaftliche Globalisierung charakterisiert ist und mit steigenden Lohnkosten, erhöhter Nachfrage und komplexeren Herstellungsverfahren konfrontiert ist, spielt die Automatisierung eine immer wichtigere Rolle. In vielen Bereichen ist der Einsatz von Robotern zur Steigerung der Produktionseffizienz und Produktqualität bereits seit einigen Jahrzehnten Standard. Dazu zählen vor allem die Automobil- und Elektronikindustrie. Bei den dort auftretenden Automatisierungsaufgaben geht es in der Regel darum, genau beschriebene, sich jeweils zyklisch wiederholende Tätigkeiten einer Maschine zu übertragen, die diese effizienter und präziser durchführen kann als ein Mensch. Beispiele hierfür sind etwa das Setzen verschiedener Schweißpunkte an einer Karosserie oder die Platzierung von Kondensatoren und Prozessoren auf Platinen. Die Produktivität in zahlreichen Industrien konnte mit Hilfe der Automatisierungstechnik in den letzten Jahrzehnten drastisch gesteigert werden.

Industrieroboter arbeiten in der Regel in hoch strukturierten Umgebungen und sind in der Lage genau spezifizierte und a-priori definierte Aufgaben präzise zu erledigen. Es ist in diesem Kontext keine Reaktion oder Adaption auf sich ändernde Umwelteinflüsse vorgesehen. Ebenso ist das selbständige Lösen und Planen einer komplexen, ungenau beschriebenen Aufgabe in einer dynamischen und unstrukturierten Umgebung durch Industrierobotersysteme nicht vorgesehen. Angeregt durch die Diskussion der demografischen Entwicklung sollen Roboter jedoch in naher Zukunft auch in privaten Haushalten Einzug finden und dort Aufgaben übernehmen, die bisher durch Menschen ausgeführt wurden. Aufgrund dieser Anforderungen entstand der Begriff der *Servicerobotik*. Diese hat zum Ziel, weitgehend autonome Systeme zu realisieren, die in der Lage sind in einer sich dynamisch verändernden und unstrukturierten Umgebung eine vorgegebene Aufgabenklasse zu erledigen. Hierzu ist die Nutzung sensorischer Information, das selbständige Planen eigener Handlungsketten und das Lernen neuer Fähigkeiten und Zusammenhänge erforderlich.

Neben der Beobachtung einer dynamischen Umgebung wie z.B. eines privaten Haushalts, ist die Deutung und Interpretation der Interaktion zwischen Mensch und Maschine von zentraler Bedeutung für den Erfolg eines Serviceroboters. Hier geht es insbesondere darum, Kommunikationskanäle zu schaffen, die es dem Menschen erlauben auf möglichst einfache und natürliche Weise mit dem Robotersystem zu interagieren. Zusätzlich soll ein solches System in der Lage sein, die Intentionen des Menschen zu erkennen und entsprechend angepasst zu reagieren. Dies bedeutet salopp formuliert, dass die Maschinen sich den Präferenzen des Menschen anpassen.



Die zukünftigen Herausforderungen in der Servicerobotik liegen also im Bereich der sensorischen Umwelt- und Situationserkennung, der Aktorik, der Manipulation und Planung, der Mensch-Maschine-Interaktion sowie des Lernens und Verstehens der Umwelt.

## 1.1 Motivation

Wie bereits angesprochen, sind die Wahrnehmung von Objekten, Menschen und Situationen, sowie situationsgerechtes Handeln, zwei der großen Herausforderungen der Servicerobotik. Um erfolgreich mit der Umwelt interagieren zu können muss das System in der Lage sein, die aktuelle Umgebung zu erkennen, vor allem aber auch die einzelnen darin enthaltenen Gegenstände identifizieren, lokalisieren und manipulieren zu können. Des Weiteren sind für die Kommunikation mit dem Menschen gemeinsame Begrifflichkeiten unerlässlich. Genau wie der Mensch benötigt ein autonomes Robotersystem also eine geeignete Repräsentation der Umwelt. Erst ein geeignetes Modell ermöglicht die Interpretation der Sensormessungen und das Verstehen und Planen abstrakt beschriebener Handlungen.

Zum Zeitpunkt der Entstehung dieser Arbeit, entstanden die meisten Modelle für Perzeption und Manipulation durch aufwändige manuelle Programmierung und Modellierung. Der damit verbundene Aufwand bedeutete, dass die Menge und Komplexität der modellierbaren Gegenstände und Szenarien sehr eingeschränkt war, was in der Folge die Handlungsfähigkeiten der Systeme auf einzelne, einfache Testszenarien und spezielle Anwendungen reduzierte. Um diese Einschränkung zu reduzieren, war es

notwendig Methoden zu untersuchen, die den Modellierungsprozess vereinfachen, zeitlich beschleunigen und die zu erzeugenden Modelle komplexer und genauer automatisch generieren zu können.

Neben der individuellen Betrachtung von einzelnen Objekten innerhalb einer Szene, gibt auch die Zusammensetzung der Szene, also die räumliche Konfiguration der Objekte Aufschluss über die Szene und auch über die Objekte selbst. Gerade in anthropomorphen Umgebungen lässt sich eine gezielte Anordnung von Objekten hinsichtlich deren Funktionalität und anderer Modalitäten beobachten. Die Beziehungen der Objekte zueinander beinhalten weitere Informationen über die Objekte selbst. Die Erfassung und Beschreibung ganzer Szenen unter dem Aspekt räumlicher Objektbeziehungen ist ein weiterer Aspekt der vorliegenden Arbeit.

## **1.2 Zielsetzung**

Aus der skizzierten Motivation lassen sich zwei Ziele für diese Arbeit ableiten: Zum einen soll ein sensorgestütztes Modellersystem mit entsprechender Modellgenerierungs-Software entwickelt werden, dem es möglich ist nahezu beliebige Alltagsobjekte in rechnerinterne, präzise Modelle zu überführen, zum anderen soll eine Methode entwickelt werden, die eine Erfassung von Objekten im Szenenkontext ermöglicht.

Das Objektmodellierungssystem soll es ermöglichen eine Vielzahl von Alltagsgegenständen zu digitalisieren und in eine maschinelle Beschreibung zu überführen, die eine weitere Verarbeitung im Kontext der Servicerobotik ermöglicht.

Die Erfassung ganzer Szenen hat zum Ziel, Hintergrundwissen über einen oder mehrere Klassen von Alltagsszenen und den darin enthaltenen Objekten zu sammeln und ebenfalls in Form einer maschinellen Beschreibung den Teilsystemen eines Serviceroboters zur Verfügung zu stellen.

### **1.3 Beitrag**

Der Aufteilung der Arbeit in Objekt- und Szenenmodellierung folgend, lassen sich die Beiträge der Arbeit entsprechend in zwei Gruppen einteilen.

Im Bereich der Modellierung von Einzelobjekten leistet die Arbeit folgende Beiträge:

- Auslegung und Aufbau eines spezialisierten, sensorgestützten Modellierungssystems
- Entwicklung einer effizienten, exakten, semi-automatischen Prozesskette zur Objektmodellierung
- Implementierung der Softwarekomponenten zum Betrieb des Modellierungssystems und der Durchführung des Modellierungsprozesses
- Entwicklung und Evaluierung einer Methode zur Kalibrierung der Sensorik des Modellierungssystems
- Experimentelle Evaluierung der Kalibrierungs- und Modellierungsergebnisse
- Entwicklung und Betrieb einer Webplattform, die den Zugriff auf die generierten Modelle für Entwickler und Roboteranwender ermöglicht

Im Bereich der Modellierung von Alltagsszenen leistet die Arbeit folgende Beiträge:

- Anpassung und Implementierung eines Verfahrens zur sensoriiellen Erfassung von Alltagsszenen
- Entwicklung einer räumlichen Relation zur Beschreibung von Objektbeziehungen
- Entwicklung und Implementierung eines Annotierungsprozesses als Grundlage der Szenenmodellierung
- Entwicklung einer probabilistischen Szenenbeschreibung auf Basis räumlicher Relationen
- Experimentelle Evaluierung der vorgestellten Szenenbeschreibung

Insgesamt leistet die vorliegende Arbeit verschiedene Beiträge in den Gebieten der Perzeption und Manipulation für Robotersysteme. Durch die entwickelten Verfahren und Komponenten wird die Erzeugung von qualitativ hochwertigen Objekt- und Szenenmodellen wesentlich vereinfacht und der dafür notwendige Aufwand reduziert.

## **1.4 Überblick**

Die vorliegende Arbeit gliedert sich wie folgt. Im Anschluss an diese Einleitung folgt in Kapitel 2 die Beschreibung des aktuellen Stands der Forschung zu den Themen Objekt- und Szenenmodellierung. Neben einer Einordnung in den Kontext der Servicerobotik und der damit verbundenen

Herleitung der Motivation, werden verwandte Arbeiten einer kritischen Betrachtung unterzogen und bewertet.

Kapitel 3 stellt im Anschluss die Grundlagen vor, auf denen die Arbeit aufbaut. Dazu zählen vor allem die Beschreibung aktueller Verfahren im Bereich der optischen Messtechnik. Insbesondere die Tiefen- und Abstandsmessung in ihren unterschiedlichen Ausprägungen wird vorgestellt.

Den technischen und methodischen Grundlagen folgend, stellt Kapitel 4 die wesentlichen Konzepte vor, die im Rahmen der Arbeit entwickelt und umgesetzt wurden. Dazu zählt sowohl die Prozesskette zur Modellierung von Einzelobjekten, wie auch die Methode zur Modellierung von Szenen.

Kapitel 5 beschreibt schließlich die Modellierung der Einzelobjekte im Detail. Behandelt werden unter anderem die Komponenten des Modellierungssystems, das Verfahren zur Kalibrierung sowie die gesamte Prozesskette ausgehend vom realen Objekt bis zur Veröffentlichung des Modells innerhalb der Webplattform.

Analog widmet sich Kapitel 6 der detaillierten Beschreibung der verschiedenen Teilaspekte der Szenenmodellierung. Das verwendete Verfahren zur Digitalisierung von Alltagsszenen wird ebenso vorgestellt wie die Grundstruktur aus räumlichen Relationen, welches die Basis für die probabilistische Szenenbeschreibung bildet. Zusätzlich wird eine mögliche Anwendung dieser Beschreibung im Kontext der Servicerobotik vorgestellt.

Kapitel 7 widmet sich dann den verschiedenen Softwarekomponenten, die zur praktischen Umsetzung der Konzepte und Methoden entwickelt wurden.

Darauf folgend werden in Kapitel 8 die mit Hilfe dieser Implementierungen gewonnenen Ergebnisse vorgestellt und anhand verschiedener Experimente diskutiert. Es finden sich hier Visualisierungen von digitalisierten Objekten und Daten zu experimentell gewonnenen Szenenbeschreibungen.

Im letzten Kapitel werden die erzielten Ergebnisse schließlich zusammengefasst und reflektiert. Weiterhin werden mögliche Verbesserungen und Möglichkeiten zur Fortführung der Arbeit diskutiert.

## Stand der Forschung

Der Beitrag, den eine wissenschaftliche Arbeit leistet, kann nur dann sinnvoll bewertet werden, wenn sie in den Kontext ähnlicher Arbeiten eingeordnet wird. Die Betrachtung des Standes der Forschung ist aber auch relevant um die Entstehung der Arbeit nachvollziehbar zu machen und die Publikationen aufzuzeigen, die als Grundlage gedient haben und somit die Arbeit überhaupt erst möglich gemacht haben.

In diesem Kapitel werden zunächst Arbeiten aus dem Bereich der Perzeption und Manipulation vorgestellt um den Bedarf und die Erfordernisse der Erstellung geeigneter Objektrepräsentationen aufzuzeigen. Dabei wird auf existierende Systeme zur Digitalisierung eingegangen. Anschließend wird die Perspektive von einzelnen Objekten auf ganze Szenen erweitert und Arbeiten vorgestellt, die sich mit der Modellierung, Repräsentation und Erkennung von räumlichen Szenen beschäftigen.

## 2.1 Sensorgestützte Erfassung und Modellierung von Objekten

Im Bereich der Servicerobotik bezeichnet der Begriff *Objektmodell* zu meist die Art und Weise wie ein Teil der Umwelt (Gegenstand, Person, Roboter) rechnerintern beschrieben wird. Dazu wird mindestens ein Attribut des zu beschreibenden Objektes ausgewählt und dessen Ausprägung mathematisch formuliert. Die genaue Formulierung hängt dabei stark von der jeweiligen Anwendung ab. In den folgenden Abschnitten wird versucht einen möglichst repräsentativen Querschnitt über die im Bereich der Servicerobotik verwendeten Objektmodelle und Modellierungsverfahren zu geben. Dies dient zum einen dazu, die Anforderungen an einen automatisierten Modellierungsprozess aufzuzeigen und zum anderen, die von einem Objektmodell erwarteten Informationen, durch Betrachtung der einzelnen Ansätze zu ermitteln.

### 2.1.1 Modelle im Bereich Objekterkennung und -lokalisierung

In der Servicerobotik ist die Erkennung und Lokalisierung von Objekten in der Umwelt des Robotersystems eine besonders wichtige Fähigkeit, ist sie doch die Grundlage jeglicher Interaktion. Objekterkennung betrifft hierbei die Identifikation und Klassifikation eines Gegenstandes, was sowohl seine Segmentierung aus der umgebenden Szene beinhaltet, wie auch die Erkennung des Typs des Objektes oder anderer spezifischer Eigenschaften. Ziel der Objektlokalisierung ist die Feststellung der genauen Lage eines Objektes, entweder bezüglich des Sensorsystems oder eines entsprechend gewählten globalen Referenzsystems. Die Lage beinhaltet dabei üblicherweise drei bis sechs Freiheitsgrade (Position und Orientierung).



Die in der Robotik verwendeten Verfahren zur Objekterkennung und -lokalisierung basieren zum größten Teil auf visuellen Sensordaten (i.d.R. Intensitäts- und Tiefenbilder) und lassen sich dann grundsätzlich in zwei methodische Gruppen unterteilen: ansichtsbasierte Verfahren und modellbasierte Verfahren. Die folgende Übersicht basiert auf [Azad 08a] und [Kragic 09], dabei soll hier weniger auf die genaue Funktion der Erkennungs- und Lokalisierungsalgorithmen eingegangen werden, sondern viel mehr auf die benötigten Referenzdaten.

Die ansichtsbasierten Verfahren können in globale und lokale Ansätze unterteilt werden. Globale Verfahren verwenden a-priori Wissen in Form von Komplettansichten der zu erkennenden Objekte. Diese werden dann mit Hilfe verschiedenster Algorithmen mit dem zu untersuchenden Bild verglichen, z.B. Grauwertkorrelation, Verwendung von Momenten oder der Viola-Jones-Detektor [Viola 01] um nur ein paar zu nennen. Lokale Verfahren beschreiben ein Objekt als eine Menge lokaler Merkmale, die dann jeweils verglichen werden. Beispiele für bekannte Verfahren zur Berechnung von solchen Merkmalen sind der Harris-Kantendetektor [Harris 88] oder die Shi-Tomasi-Merkmale [Shi 94]. Für die Beschreibung dieser Merkmale zum Zweck der Korrespondenzfindung wurden unterschiedliche sog. Merkmalsdeskriptoren entwickelt. Besonders bekannt sind hier die SIFT<sup>1</sup>- und die SURF<sup>2</sup>-Deskriptoren. Eine Übersicht zu visuellen Erkennungsverfahren die auf solchen *Bag of Keypoints*-Methoden beruhen, findet sich in [Ramanan 11].

Die modellbasierten Verfahren im Bereich der Bildverarbeitung verwenden in der Regel geometrische Modelle um Objekte zu erkennen und zu

---

<sup>1</sup> Scale Invariant Feature Transform, [Lowe 04]

<sup>2</sup> Speeded Up Robust Feature, [Bay 08]

lokalisieren. Mögliche Methoden hierfür sind z.B. Verfahren die auf Kantedetektion aufbauen oder Verfahren, die lokale Merkmale auf der Oberfläche eines 3D-Modells verwenden. Ein bekannter Algorithmus in diesem Bereich ist der POSIT<sup>3</sup>-Algorithmus.

Im Folgenden sollen nun verschiedene Arbeiten im Bereich der Objekterkennung und -lokalisierung betrachtet werden um zu identifizieren welche Ausgangsdaten dort als Referenz- bzw. Trainingsdaten verwendet werden. Wenn sich auch das grundlegende Problem, Gegenstände einer Realszene zu erkennen und zu lokalisieren, einfach allgemein formulieren lässt, so ist eine ebenso allgemeine Lösung dieses Problems noch nicht erkennbar. Viel mehr finden sich spezielle Ausprägungen der bereits angedeuteten Verfahren, optimiert auf die jeweilige Szenerie und die darin vorkommenden Objekte.

Die Erkennung von Objekten unter Zuhilfenahme ihres strukturellen Aufbaus beschreibt [He 11]. In dieser Arbeit sollen in 2D-Bildern zusammengesetzte Objekte erkannt werden. Deren topologischer Aufbau ist a-priori bekannt und wird mit Hilfe eines Sternmodells codiert. Im Eingangsbild werden dann unter Verwendung eines HOG<sup>4</sup>-Klassifikators die einzelnen Teile segmentiert und versucht diese mit dem Strukturmodell in Übereinstimmung zu bringen. Für das Training des HOG-Klassifikators werden Serien von annotierten Beispielbildern mit möglichst verschiedenartiger Beleuchtung benötigt. Die in diesem Ansatz verwendete Objektbeschreibung besteht also aus einer Menge annotierter Beispielbilder und dem topologischen Sternmodell.

---

<sup>3</sup> Pose from Orthography and Scaling with Iterations, [Dementhon 95]

<sup>4</sup> Histograms of Oriented Gradients, [Dalal 05]

Ebenfalls auf reinen 2D-Bildern arbeitet das Verfahren von [Li 08], welches mit Hilfe von sog. *Algebraic Functions of Views (AFoVs)* die Erkennung von unterschiedlichsten Objekten (als Beispiel wird die Erkennung von Autos gezeigt) leistet. Auch hier gibt es wieder eine Unterteilung in Trainings- und Erkennungsphase. In der Trainingsphase werden einige wenige Referenzansichten der zu erkennenden Objekte verwendet um daraus Funktionen abzuleiten und zu parametrisieren, mit deren Hilfe sich beliebige weitere Ansichten des Objekts generieren lassen. Diese Funktionen codieren somit quasi alle möglichen Projektionen der Objekte. Für deren Erkennung werden im Eingangsbild lokale Merkmale berechnet aus denen sich schließlich eine Objekthypothese ergibt, die dann mit den vorhandenen Trainingsdaten abgeglichen werden kann. Die dabei verwendete Objektbeschreibung besteht also aus den Parametern für die algebraischen Funktionen zur Codierung der Ansichten. Ausgangsdaten für das Erzeugen der Referenzdaten ist auch hier eine Serie von annotierten Beispielbildern.

Die Erkennung von farbigen Objekten in 2D-Bildern alleine über den Vergleich von Ansichten beinhaltet die Arbeit von [Nayar 96]. Im Rahmen der Experimente wurden hier 100 verschiedene, farbige Objekte eintrainiert. Das Training bestand aus der Aufnahme von jeweils 48 Referenzansichten mit Hilfe eines Drehtellers. Die Farbinformation aus den Ansichten wird nun als Merkmalsvektor aufgefasst. Über eine Hauptkomponentenanalyse lassen sich die einzelnen Objekte dann unterscheiden und die Erkennung anschließend über die Suche nach dem Punkt der nächsten Mannigfaltigkeit in allen Eigenräumen realisieren. Die Objektbeschreibung in dieser Arbeit sind dementsprechend die jeweiligen Eigenräume, wobei die Ein-

gangsdaten für die Erzeugung der Referenzmodelle wiederum Serien von Beispielbildern sind.

Ein Vertreter der modellbasierten Erkennungs-, und hier auch Verfolgungsverfahren, ist die Arbeit von [Choi 10]. Hierbei sollen Objekte in monokularen 2D-Bildern erkannt und verfolgt werden. Als Ausgangsdaten dafür werden für jedes Objekt wiederum eine Reihe von Referenzbildern, aber auch ein CAD<sup>5</sup>-Modell benötigt. Aus den Referenzbildern werden lokale Merkmale extrahiert, die im ersten Erkennungsschritt eine initiale Positionsschätzung erlauben. Die Verfolgung des Objektes geschieht dann über die Projektion des CAD-Modells und den Abgleich dieser Projektion mit dem Ergebnis einer Kantendetektion im Eingabebild. Die zu Grunde liegende Objektbeschreibung besteht hier also sowohl aus 2D- wie auch 3D-Daten.

[Kragic 06] führen eine Objekterkennung auf Basis von Stereobildern durch. Die eigentliche Erkennung geschieht dabei mit Hilfe von Receptive Field Cooccurrence Histograms (RFCH), welche eine statistische Repräsentation des Auftretens von bestimmten Deskriptorwerten innerhalb eines Bildes bezeichnen. Eingabe für das Training der Objektmodelle sind hier wiederum Bildserien der zu erkennenden Objekte, mit deren Hilfe die Histogramme der Deskriptoren gebildet werden, die dann für die Erkennung verwendet werden. Die Objektbeschreibung ist in diesem Fall also ein Histogramm von bestimmten Merkmalsdeskriptoren.

Nicht nur auf 2D-Bilddaten kann eine Objekterkennung und -lokalisierung durchgeführt werden. Auch 3D-Punktwolken eignen sich hierfür, wie in der Arbeit von [Lai 10] vorgestellt. Hierbei wird in der Eingabepunkt-

---

<sup>5</sup> Computer Aided Design

wolke mit Hilfe des RANSAC<sup>6</sup>-Algorithmus zunächst die Bodenebene extrahiert. Anschließend werden die verbleibenden Punkte unter Verwendung des Mean-Shift-Verfahrens in Teilpunktwolken segmentiert. Für jeden Punkt dieser Teilpunktwolken werden nun Spin-Images berechnet, die eine rotationsinvariante Objektbeschreibung ergeben. Als Referenzdaten zur Erkennung dienen 3D-Modelle (Dreiecksnetze) aus unterschiedlichen, im Internet veröffentlichten Sammlungen, welche über ein Raycasting-Verfahren in Punktwolken verwandelt werden, worauf sich schließlich ebenfalls Spin-Images berechnen lassen. Die eigentliche Erkennung basiert dann auf der Distanz zwischen den jeweiligen Mengen von Spin-Images. Die Objektbeschreibung in diesem Verfahren ist also eine Menge von Spin-Images zu jedem Punkt einer Punktwolke. Grundlage für die Referenzdaten sind a-priori erstellte CAD-Modelle, die als Oberflächenmodelle vorliegen.

Nicht alle Objekterkennungsverfahren verwenden jedoch visuelle Sensordaten und a-priori Modellwissen. In [Sinapov 11] geschieht die Erzeugung des Modellwissens autonom durch das Robotersystem mittels haptischer Exploration. Dieses greift verschiedene Objekte selbständig und zeichnet die dabei entstehenden Kräfte und Momente in seinem Endeffektor auf. Diese Sensoreingaben stellen damit auch die Objektbeschreibung dar. Wird ein bekanntes Objekt nun erneut gegriffen, kann es über den Vergleich mit den zuvor ermittelten Sensordaten wiedererkannt werden.

Die Betrachtung dieses kleinen Querschnitts von Arbeiten zum Thema der Objekterkennung und -lokalisierung für Robotersysteme, zeigt also bereits, dass die meisten Ansätze ähnliche Ausgangsdaten für die Erzeugung

---

<sup>6</sup> Random Sample Consensus

der Referenzdaten - und damit der Objektmodelle - benötigen. Dies sind vielfach Serien von Beispielbildern, aber auch 3D-Oberflächenmodelle.

### **2.1.2 Greifplanung**

Nach der Erkennung von Objekten in der Umgebung des Roboters, soll das System mit diesen interagieren. Meist ist es dazu notwendig das betreffende Objekt mit Hilfe des Endeffektors zu greifen. Die Planung und Durchführung von stabilen und effizienten Griffen ist daher ein zentrales Problem der Robotik. Bei der Herangehensweise an dieses Problem können zwei grundlegende Ansätze unterschieden werden: Zum einen gibt es datengesteuerte Greifplanung, die versucht mit Hilfe eines a-priori bekannten Modells des Objekts geeignete Griffe zu berechnen und zu speichern. Zum Zeitpunkt der Ausführung muss dann zu dem aktuell vorhandenen Objekt, ein passender vorberechneter Griff gefunden werden und dieser dann ausgeführt werden. Der Vorteil dieses Ansatzes ist, dass er wenig Rechenzeit zur Laufzeit benötigt und durch die Perzeptionskomponenten oft schon der Abgleich zu einem a-priori-Modell geleistet wird. Zum anderen gibt es Greifplanungsansätze die ohne a-priori-Wissen arbeiten und versuchen auf Basis der aktuell vorhandenen Sensordaten direkt einen geeigneten Griff zu planen und durchzuführen. Der Vorteil dieses Ansatzes ist, dass kein a-priori-Modell vorhanden sein muss um erfolgreich greifen zu können, also auch in gänzlich unbekanntem Umgebungen Interaktion ermöglicht wird.

Im Folgenden sollen verschiedene Arbeiten im Bereich der Greifplanung kurz vorgestellt werden. Dabei wird hauptsächlich auf die verwendeten

Objektmodelle eingegangen um zu ermitteln welche Daten für diesen Teilbereich der Servicerobotik vorrangig benötigt werden.

Eine der frühen Arbeiten zu generischer Greifplanung für Serviceroboter stammt von [Miller 03]. Hier wird die Greifplanung auf Basis von 3D-Modellen der zu greifenden Objekte durchgeführt. Zunächst gilt es mögliche Anrückposen des Greifers zu bestimmen. Dazu wird aus dem detaillierten 3D-Modell eine approximierte Darstellung, zusammengesetzt aus Primitiven wie Kugeln, Kegeln und Quadern, erzeugt. Die verwendeten Primitive bestimmen dann die Auswahl der Anrückposen. Sind diese gefunden wird mit Hilfe des detaillierten Modells die Durchführbarkeit und Stabilität des Griffs evaluiert. Dieser Prozess wurde in der bekannten GraspIt!<sup>7</sup>-Software implementiert.

Vollständige 3D-Modelle der zu greifenden Objekte sind für viele Arbeiten der Ausgangspunkt. [Aleotti 11] verwenden diese beispielsweise als Basis für ihre Greifplanung im Kontext des Programmierens durch Vormachen (PdV). Zunächst wird das Objektmodell in Teile zerlegt und hierarchisiert, es entsteht ein sog. Reeb Graph. Schließlich führt der Benutzer die Handhabung des Objekts vor, wobei das System die relevanten Objektteile erlernt. Ein Planer kann dann die vorgeführten Handlungen mit gleichen oder ähnlichen Objekten imitieren. Hierbei liegt der Fokus auf dem Greifen von manipulierbaren Teilen eines Objektes wie etwa einem Verschluss. Weitere Arbeiten die detaillierte 3D-Modelle als Ausgangspunkt für die Greifplanung verwenden sind [Harada 08], [Saut 12] sowie [Berenson 07].

---

<sup>7</sup> <http://www.cs.columbia.edu/~cmatei/graspit/>

Die Arbeit von [Goldfeder 09b] basiert ebenfalls auf a-priori generierten Griffen mit Hilfe von bekannten 3D-Modellen. Interessant ist hier jedoch die Anwendung dieses Wissens in der konkreten Greifsituation. Das verwendete Robotersystem besitzt einen Laserscanner, der die Szene abtastet. Dadurch kann nicht davon ausgegangen werden, dass eine vollständige Objektansicht vorhanden ist, es liegen häufig lediglich Teils cans vor. Diese Teils cans müssen nun mit den a-priori Modellen in Einklang gebracht werden und die geeigneten vorgeplanten Griffe nur anhand dieses Ausschnitts ermittelt werden. Auf dieser Arbeit bauen [Brook 11] auf und erweitern den Ansatz um verschiedenste Objektrepräsentationen. So können haptische wie auch unterschiedliche visuelle Objektmodelle verwendet werden, wobei je nach vorhandenen Sensordaten und Objektmodellen ein entsprechender Planer für die Griffsynthese ausgewählt wird. In beiden Arbeiten spielen jedoch Trainingsdaten in Form von Bildern und Tiefendaten für die a-priori-Erzeugung von Griffen eine wichtige Rolle.

Zur Greifplanung werden jedoch nicht nur 3D-Daten genutzt, auch 2D-Bilder können als Grundlage dienen. [Glover 09] stellen eine Greifplanung für dynamische Objekte vor. Hierbei werden die jeweiligen Objekte probabilistisch modelliert, indem Konturen aus Trainingsbildern extrahiert werden und daraus Verteilungen für alle Objektklassen generiert werden. Das verwendete Objektmodell ist in diesem Fall also eine charakteristische Verteilung der Konturpunkte. Die Klassifizierung erfolgt dann über eine Maximum-Likelihood-Schätzung. Die Grundlage für die Auswahl und die Durchführung des Griffs sind die rekonstruierten Objektkonturen, was die Behandlung von teilweise verdeckten Objekten möglich macht.

Eine rein auf Bildverarbeitung basierende Greifplanung, die ebenfalls Konturen verwendet, wird in [Speth 08] vorgestellt. Das Objekt wird zu-



nächst aus mehreren Perspektiven betrachtet, die jeweilige Kontur extrahiert und daraus die Boundingbox und deren Position errechnet. Auf der Kontur werden dann mögliche Greifflächen erkannt und damit der bestmögliche Griff bestimmt. Dieses Verfahren kommt also völlig ohne a-priori-Wissen aus und benötigt dementsprechend keinerlei Modell oder Trainingsdaten.

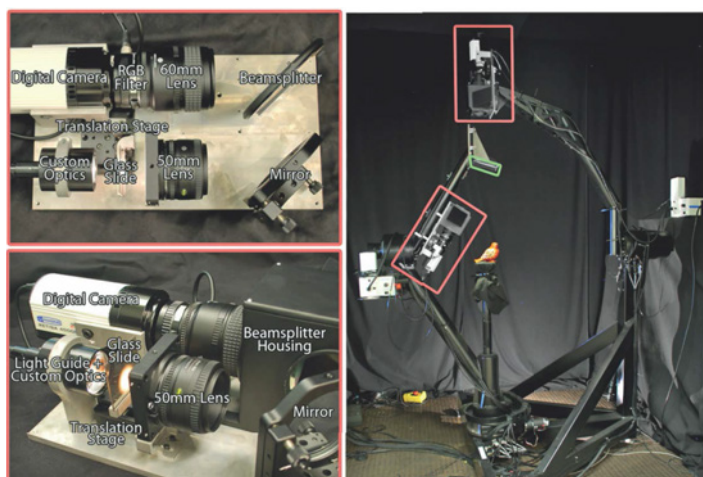
Diese Auswahl an Arbeiten zum Thema Greifplanung bzw. Objektmanipulation zeigt sehr deutlich, dass gerade 3D-Modelldaten eine wichtige Voraussetzung zur Lösung dieser Aufgabe sind. Vor allem Arbeiten, die a-priori Modelldaten verwenden, machen jedoch zumeist keine Angaben über die benötigte Qualität bzw. die wichtigen Eigenschaften der Modelldaten wie etwa Mannigfaltigkeit oder Geschlossenheit.

### **2.1.3 Objektdigitalisierung**

Nachdem nun deutlich geworden ist, dass für viele Methoden, die derzeit in der Servicerobotik eingesetzt werden, Objektmodelle in Form von 3D-Modellen oder 2D-Bilddaten benötigt werden, stellt sich die Frage welche Methoden bereits zur Erzeugung solcher Informationen existieren, bzw. welche Datensätze bereits erstellt wurden und verfügbar sind. Mittlerweile existiert eine Vielzahl von kommerziellen wie auch prototypischen Systemen mit unterschiedlichen Spezifikationen und Leistungsmerkmalen. Die Sensorsysteme zur Objektdigitalisierung lassen sich in zwei Gruppen unterteilen: einerseits stationäre Aufbauten, in deren Beobachtungsfeld das zu erfassende Objekt eingebracht wird und deren Erfassungsbereich damit beschränkt ist. Andererseits gibt es handgeführte mobile Systeme, die sich flexibel aufbauen und verwenden lassen.

## Stationäre Aufbauten

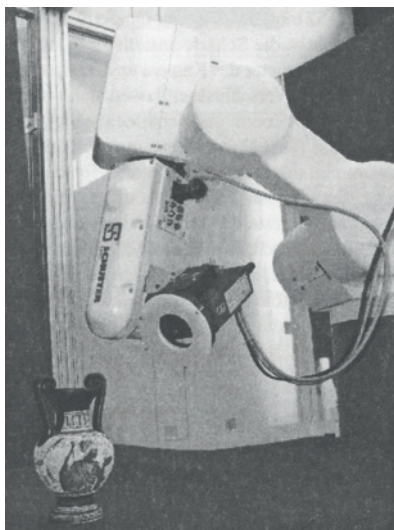
Stationäre Aufbauten zur Objektdigitalisierung zeichnen sich durch deren Spezialisierung auf bestimmte Objekttypen oder -größen aus, wobei unterschiedliche Anforderungen im Vordergrund stehen, wie beispielsweise maximale Automatisierung oder höchst mögliche Präzision.



**Abb. 2.1.** Aufbau zur Objektdigitalisierung nach [Holroyd 10].

Das *Stanford Spherical Gantry* ([Levoy 10]) ist ein spezialisierter Sensoraufbau, der ursprünglich zur Untersuchung von Lichtfeldern kleiner Objekte entwickelt wurde, ermöglicht jedoch auch die Verwendung unterschiedlichster Sensoren. Der Aufbau besteht aus zwei steuerbaren Armen, wovon einer zwei Rotationsfreiheitsgrade besitzt, der andere lediglich einen Rotationsfreiheitsgrad. Zusammen mit dem Drehteller, auf dem

die Objekte positioniert werden, ermöglicht dies den beiden Armspitzen, sich in konzentrischen Kugeln um das Objekt zu bewegen. Im ursprünglichen Aufbau befindet sich am inneren Arm eine CCD-Kamera, während am äußeren Arm eine Lichtquelle montiert ist. Der Arbeitsraum umfasst eine Kugel mit Radius 20,32 cm. Aufbauend auf diesem Design wurde von [Holroyd 10] ein Objektdigitalisierungssystem entwickelt, welches auf Basis von sinusförmig modulierter Beleuchtung arbeitet. Im Verlauf der Aufnahme eines Objektes werden aus verschiedenen Positionen einzelne *Scans* durchgeführt. Ein Scan besteht dabei aus mehreren Bildaufnahmen unter modulierter Beleuchtung. Für jedes dieser Bilder wird dann pro Pixel die Phasenlage berechnet. Anschließend wird für jeden Scan mit Hilfe eines merkmalsbasierten Algorithmus die Kamera (und wegen der koaxialen Anordnung auch der Lichtquelle) ermittelt. Weiterhin wird aus jedem Scan eine Tiefenkarte berechnet, die dann alle zu einem geschlossenen Netz fusioniert werden. Als letzter Schritt wird schließlich für jeden Knoten in diesem Netz eine Reihe von Messungen für die bidirektionale Reflektanzverteilungsfunktion ermittelt. Die bidirektionale Reflektanzverteilungsfunktion (BRDF) beschreibt als Funktion die Reflektionseigenschaften einer Oberfläche. Mit Hilfe der Werte dieser Funktion kann bei der künstlichen Bilderzeugung (Rendering) die zu erwartende Lichtintensität abhängig von der Stellung der Lichtquellen und der Kamera ermittelt werden. Die ermittelte Genauigkeit der 3D-Erfassung beträgt zwischen 40 und 50  $\mu\text{m}$ . Der gesamte Erfassungsprozess ist vollständig automatisiert, die Erfassungszeit pro Objekt liegt zwischen 6 und 7 Stunden. Systembedingt kann der Auflagebereich der Objekte nicht erfasst werden, da dies eine Repositionierung des Objekts erfordern würde.



**Abb. 2.2.** Der CaRo-Kameraroboter von Fautz. (Quelle: [Fautz 02]).

Einen Ansatz zur 3D-Erfassung von Objekten mit Hilfe einer frei geführten Kamera beschreibt [Fautz 02]. Der Aufbau besteht aus einem 6-achsigen Roboterarm an dessen Ende eine Kamera mit 1/3" CCD-Sensor mit einer Auflösung von 748x576 Bildpunkten montiert ist. Dies ermöglicht eine genaue Positionierung der Kamera in nahezu beliebigen Posen innerhalb des Arbeitsraums des Roboterarmes. Interessant an diesem Aufbau ist der Ansatz zur 3D-Rekonstruktion, basierend auf dem Volumenschnitt. Ausgangspunkt dieses Verfahrens ist die Segmentierung der Objektkontur in den aufgenommenen 2D-Bildern. Die Kontur wird dann durch Polygone angenähert. Mit Hilfe der Abbildungsparameter der Kamera lässt sich aus der Kontur eine Sichtpyramide erzeugen. Schneidet man die verschiedenen Sichtpyramiden, die aus unterschiedlichen Ansich-

ten erzeugt wurden, entsteht ein Polyeder. Durch sukzessives Schneiden mit weiteren Sichtpyramiden wird der Polyeder verfeinert und konvergiert dann gegen die visuelle Hülle des Objekts. Neben der Rekonstruktion der Objektgeometrie, ist die Erzeugung einer Objekttextur ein wichtiger Bestandteil dieses Ansatzes. Hierzu werden die einzelnen Kamerabilder anhand ihrer Position relativ zur Objektoberfläche bewertet und eine Untermenge aller Bilder als Ausgangsdaten für die Textur ausgewählt. Mit Hilfe einer Überblendungsfunktion entlang der Übergangskanten (Wechsel des Kamerabildes innerhalb der Textur) wird die Qualität der finalen Textur verbessert. Das verwendete Volumenschnittverfahren eignet sich zur Rekonstruktion der Geometrie verschiedenster Objekte, unterliegt jedoch gewissen Einschränkungen. So können Löcher in gekrümmten Oberflächen prinzipbedingt schlecht oder gar nicht rekonstruiert werden. Zudem ergeben sich Probleme bei filigranen Objektdetails, die durch Fehler in der Kamerapositionsbestimmung oder numerische Ungenauigkeiten verloren gehen können. Fautz gibt die Genauigkeit für das Verfahren im Bereich von 0,12 - 0,46 mm an. Die Messdauer wird maßgeblich von der Geschwindigkeit des Roboterarmes zur Positionierung der Kamera bestimmt. Für 100 Aufnahmen werden dafür ca. 11 Minuten benötigt, die weitere Nachverarbeitung zur eigentlichen Rekonstruktion schlägt dann noch einmal mit etwa 6 Minuten zu Buche, es ergibt sich also eine Gesamtzeit von etwa 17 Minuten, wobei sich keine eindeutigen Aussagen über die Genauigkeit und Auflösung für diesen Fall machen lassen.

Kommerzielle Systeme zur dreidimensionalen Rekonstruktion von Objekten und Personen sind mittlerweile sehr zahlreich am Markt vertreten. Die Firma Cyberware Inc. zählt zu den ersten Anbietern von kompletten Scan-Aufbauten und bietet mehrere verschiedene Systeme für unterschiedliche

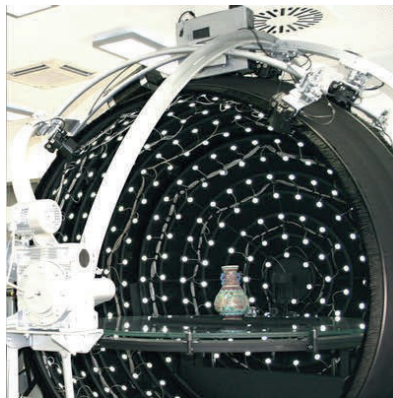


**Abb. 2.3.** Objekt-Scan-System der Firma Cyberware (Quelle: [Cyberware Inc. 11]).

Anforderungen an. Abbildung 2.3 zeigt das Modell „Model Shop Color 3D Scanner“ zur Erfassung von kleineren bis mittelgroßen, statischen Objekten. Dieses System arbeitet mit strukturiertem Licht, konkret wird eine Laserlinie auf das Objekt projiziert, welche von zwei Kameraeinheiten erfasst wird. Das Objekt befindet dabei auf einem Rotationsteller, der zusätzlich längs zur Scaneinheit bewegt werden kann. Die Position der Scaneinheit lässt sich in der Höhe manuell einstellen. Der Sensorkopf erfasst mit einer Aufnahme maximal 512x512 Tiefenwerte, zusätzlich kann eine Farbtextur mit maximal 2000x2000 Pixeln aufgenommen werden. Die Genauigkeit der 3D-Erfassung gibt der Hersteller für diesen Sensor im Bereich zwischen 75 und 500  $\mu\text{m}$  an. Das Sichtfeld umfasst einen Bereich

von ca. 300x340x300 mm. Weiterhin werden noch Komplettsysteme speziell zur Erfassung von Köpfen oder Personen angeboten.

Neben der Verwendung von Laserlinien zur aktiven Triangulation, werden in kommerziellen Systemen häufig auch Projektoren in Verbindung mit komplexeren Mustern (Streifen oder Rauschmuster) eingesetzt. Die Firma Steinbichler ([Steinbichler Optotechnik GmbH 12]) etwa bietet verschiedene Modelle auf Basis dieser Technik an. Hierbei variiert vor allem die Auflösung der eingesetzten Kamera (2-11 Megapixel) und damit auch die minimale Aufnahmezeit (0,6-4 Sekunden pro Einzelaufnahme). Die erzielbaren Genauigkeiten hängen ebenso von der eingesetzten Messoptik ab und bewegen sich im Bereich von 18-500  $\mu\text{m}$ . Um eine vollständige Objektvermessung zu erreichen, kann entweder das Objekt auf einen Rotationsteller gestellt werden oder der Sensorkopf mit Hilfe eines Roboterarmes automatisiert positioniert werden.



**Abb. 2.4.** Orbital Camera System der Firma NEK GmbH in Zusammenarbeit mit Technische Universität Kaiserslautern, AG Prof. Didier Stricker (Quelle: [Stricker 12]).

Ein weiteres, zum Zeitpunkt der Fertigstellung dieser Arbeit erst kurze Zeit auf dem Markt befindliches System zur automatisierten Digitalisierung von Objekten wird von der Firma NEK GmbH unter dem Namen *Orcam* (Orbital Camera System, [Stricker 12]) vertrieben. Wie in Abbildung 2.4 zu sehen, besteht der Aufbau aus einer großen Kugel, an deren Innenseite eine Vielzahl an steuerbaren Leuchtelementen angebracht ist. Das zu verarbeitende Objekt wird auf einer runden, rotierbaren Glasplatte platziert, wodurch auch Aufnahmen von der Unterseite ermöglicht werden. Zur Erfassung der Geometrie und der Textur wird ein Streifenprojektor zusammen mit 7 Kameras verwendet, die an einem Ring um das Objekt herum rotieren können. Zusammen mit dem Rotationsteller können so Aufnahmen aus allen Positionen rund um das Objekt herum gemacht werden. Es können Objekte bis zu 80 cm Größe und 100 kg Gewicht aufgenommen werden. Da es sich um ein kommerzielles Prototypsystem handelt, sind leider keine weiteren Angaben zur Genauigkeit, der Geschwindigkeit oder sonstigen Einschränkungen verfügbar.

### **Klein- und Handgeräte**

Neben den bereits diskutierten stationären Aufbauten zur Objektdigitalisierung gibt es auch mobile Systeme kleineren Formats. Diese Systeme werden meist zur Digitalisierung von großen Objekten wie bspw. Automobilen eingesetzt, da sie die stückweise Erfassung ermöglichen und sich leicht zum Einsatzort transportieren lassen. Technische Unterschiede gibt es hierbei nicht nur bei den eingesetzten Messverfahren sondern vor allem bei der Rekombination der Einzelaufnahmen zum finalen Ergebnisdatensatz. Letzteres kann dabei vollständig durch Software geschehen, also z.B. durch Verfolgung von speziellen Markern oder Merkmalen, aber auch



durch zusätzliche Positionssensorik wie etwa eine passive kinematische Kette, erfolgen. Im Folgenden sollen exemplarisch einige besonders interessante Vertreter dieser Systemklasse vorgestellt werden, für eine vollständigere Auflistung von Herstellern und Vertreibern solcher Systeme wird auf [Wohlers Associates 10] verwiesen.



**Abb. 2.5.** 3DMo-System des DLR (Quelle: [http://www.dlr.de/rm/en/portaldata/52/Moduledata/7983/sv\\_503\\_1\\_1.jpg](http://www.dlr.de/rm/en/portaldata/52/Moduledata/7983/sv_503_1_1.jpg)).

Ein prototypisches System aus der Robotikforschung, welches gleich mehrere Sensorsysteme zur 3D-Erfassung in sich vereint, ist das *3DMo*-System der DLR<sup>8</sup> (vgl. [Suppa 07] und Abb. 2.5). Dieser Sensorkopf, der sowohl an einem Robotersystem, wie auch handgeführt betrieben werden kann, verwendet neben einem Stereokamera paar einen Streifenprojektor und ein Abstandsmesssystem auf Basis von Lasertriangulation. Die Ab-

<sup>8</sup> Deutsches Zentrum für Luft- und Raumfahrt

standsmessung mittels Stereokamera und Streifenprojektor wird in Abschnitt 3 näher erläutert. Der Lasertriangulationssensor sendet einen Laserstrahl aus, dessen Reflektion von einem lichtempfindlichen Sensorelement mit Positionsauflösung registriert wird. Die Position des reflektierten Lichtpunkts ist dabei vom Abstand des Reflektionspunktes zum Sensor abhängig, wodurch mit einer Messung genau ein Tiefenwert rekonstruiert werden kann. Durch Rotation des Sensors und laterale Verschiebung kann über die Zeit eine räumliche Abtastung realisiert werden. Der Arbeitsbereich des Sensors liegt zwischen 53 und 300 mm, die Abtastrate bei max. 10 kHz. Wird das System handgeführt betrieben, kann die Position des Sensorkopfs durch an den Seiten angebrachte passive Marker erfasst werden.

Eine umfassende Produktpalette an frei geführten Scansystemen bietet die Firma Creafom (vgl. [Creafom 3D 12]) an. Die Produktreihe der Handyscan-Systeme basiert dabei auf dem Verfahren der aktiven Triangulation mittels Laserlicht. Die hierbei erzielbare Auflösung wird mit 0,1 mm angegeben, die maximale Genauigkeit zwischen 0,04 und 0,08 mm. Um die Position des frei geführten Sensors zu ermitteln muss das zu vermessende Objekt zunächst mit künstlichen Landmarken versehen werden. Diese werden dann von einem integrierten Kamerasystem verfolgt und auf deren Basis die Position des Sensors ermittelt. Dazu muss natürlich für jede Messung sichergestellt sein, dass sich eine Mindestanzahl dieser Landmarken im Sichtbereich des Systems befindet. Gleichzeitig wird durch die künstlichen Landmarken die natürliche Oberfläche des Objekts verändert, was vor allem die Erfassung der Farbinformationen beeinflusst.

Als Vertreter der mobilen Kleingeräte, die nicht handgeführt arbeiten, kann der *QTSculptor* der Firma Polygon Technology GmbH

([Polygon Technology GmbH 12]) angeführt werden. Dieses System arbeitet mit einem Streifenprojektor und zwei im Abstand zum Projektor und zueinander verstellbaren Kameras. Zusätzlich können hier die Linsen der Kameras ausgetauscht werden, was auch Aufnahmen im Makrobereich (Abbildungsmaßstab 1:1) ermöglicht. Insgesamt liegen die möglichen Objektgrößen im Bereich von 25x18 mm bis 1500x1125 mm, die dabei erzielbare Genauigkeit ist zwischen 0,0027 und 0,1 mm (Tiefe) bzw. 0,016 bis 0,78 mm (lateral) angegeben.

Neben den kommerziell vermarkteten und speziell auf die Anforderungen der 3D-Rekonstruktion ausgelegten Sensorsystemen, hat das Kinect-Sensorsystem, vertrieben von Microsoft<sup>®</sup> und entwickelt von PrimeSense - ursprünglich als alternative Eingabemethode für die Spielkonsole Xbox 360<sup>®</sup> - eine ganze Reihe von Digitalisierungsapplikationen begründet. Da dieser Sensor von sich aus keine spezialisierte Lösung für die Positionserkennung bietet, wird dies von den meisten Ansätzen mittels einer markerlosen Merkmalsverfolgung in der Farbbildsequenz durchgeführt. Zwei Ansätze, die auf Basis solch einer markerlosen Kameraposenverfolgung arbeiten sind die Software *ReconstructMe* ([Profactor GmbH 12]) und *KinectFusion* ([Izadi 11]). Über die genaue Funktionsweise von *ReconstructMe* sind keine Angaben zu finden, prinzipbedingt ist aber davon auszugehen, dass der Ansatz ähnlich zur Arbeit von Izadi et al. funktioniert. Dort wird die Verfolgung der Kamera rein auf den Tiefenbildern des Sensors durchgeführt und die aufgenommenen Daten werden in einer Voxelrepräsentation schrittweise fusioniert. Dies bedeutet, dass keine direkte Oberflächenrekonstruktion durchgeführt wird, sondern der Raum in kleine Einheiten unterteilt wird, die Informationen über das Vorhandensein eventueller Oberflächen (Abstand und Normale) enthalten. Dies

erlaubt zusammen mit einer effizienten, GPU<sup>9</sup>-basierten Implementierung eine echtzeitfähige Rekonstruktion von Räumen und Objekten. Der Nachteil dieser Methode besteht darin, dass die Größe bzw. die Auflösung des rekonstruierten Bereichs von der Menge an zur Verfügung stehendem Grafikspeicher limitiert ist. Zusätzliche Einschränkungen in Genauigkeit und Materialabhängigkeiten ergeben sich aus den Spezifikationen des verwendeten Kinect-Sensors, der in Abschnitt 3.2.2 genauer beschrieben wird. Weitere Arbeiten zur Objekt- bzw. Szenenrekonstruktion mit Hilfe des Kinect-Sensors sind z.B. [Jeromin 12], [Engelhard 11] und [Schulze 12].

Systeme zur 3D-Rekonstruktion von Objekten lassen sich aber auch mit einfachen Mitteln herstellen. Besonders das Verfahren der aktiven Triangulation mit Lichtschnitt (vgl. Abschnitt 3.2.2) bietet sich hierfür an. Mit Hilfe einer Kamera und einem Linienlaser lassen sich schon gute Ergebnisse erzielen. Anleitungen und Software für solche Heimbausätze finden sich bei [Gockel 06b] und [DAVID Vison Systems GmbH 12].

## **Datensätze**

Neben dem Bedarf an Modelldaten und den vorhandenen Systemen zur Erzeugung solcher Daten, gilt es noch zu untersuchen welche Datensätze bereits vorhanden sind und wie genau die darin zur Verfügung gestellten Daten sind. Die für den Bereich der Servicerobotik relevanten Projekte können in „reale“ und „künstliche“ Datensätze unterteilt werden. Erstere werden mit Hilfe von Sensoren erzeugt, deren Daten dann entweder aufbereitet oder möglichst sensornah zur Verfügung gestellt werden. Die künstlichen Datensätze finden sich überwiegend bei den 3D-Daten, die manuell

---

<sup>9</sup> Graphics Processing Unit, dedizierter Prozessor zur Verarbeitung von Grafikoperationen

mit Hilfe entsprechender Software, entweder nach einer realen Vorlage oder völlig frei, erstellt werden. Häufig sind die Datensätze mit der Absicht erzeugt worden, eine Testdatenmenge (Benchmark) zu generieren, die einen möglichst objektiven Vergleich verschiedener Ansätze ermöglicht.

Im Bereich der Bildverarbeitung, also z.B. der bildbasierten Erkennung oder Lokalisierung von Objekten findet sich eine Vielzahl an Testdatensätzen. In diesen Datensätzen sind in der Regel viele möglichst unterschiedliche Objekte vorhanden, wobei sowohl die Perspektive wie auch das Vorhandensein weiterer Objekte im Bild von Fall zu Fall mitunter stark variiert. Auch kann unterschieden werden zwischen annotierten Datensätzen und Datensätzen ohne „ground truth“. Ein Vertreter der annotierten Datensätze ist *ImageNet* ([Deng 09]). Der Datensatz ist entsprechend der Hierarchie des *WordNet* ([Stark 98]) organisiert, also in sog. „Synsets“ (synonym set) aufgeteilt. Ein Synset beinhaltet dabei Worte bzw. Objekte mit gleicher oder ähnlicher Bedeutung. *ImageNet* hält nun für jedes dieser Synsets durchschnittlich 1000 handverlesene Bilder vor, die Kommentierungen in Form von Boundingboxen enthalten bzw. für die bereits die in [Lowe 04] beschriebenen SIFT-Merkmalsskriptoren vorberechnet wurden. Insgesamt finden sich über 1 Mio. Bilder in der Datenbank.

Im Rahmen der „PASCAL Visual Object Classes Challenge“ ([PASCAL 12]), einer Serie von Wettbewerben zum Thema Musteranalyse und maschinelles Lernen sind eine ganze Reihe verschiedener Datensätze entstanden. Diese enthalten mehrheitlich thematisch gruppierte Bilder, wobei sowohl kommentierte, teilweise kommentierte und kommentarfreie Datensätze existieren. [Leibe 04] beschreiben einen Objekterkennungsansatz, der auf einem kommentierten Bilddatensatz

bestehend aus 326 Seitenansichten von Motorrädern, Autos und Kühen besteht. Ebenfalls mit motorisierten Fahrzeugen befasst sich der Datensatz aus [Agarwal 02], der 1328 kommentierte Bilder enthält, wobei hierin auch Negativbeispiele vorkommen. Weitere Datensätze im Rahmen dieser Wettbewerbe finden sich in [Everingham 06b], [Everingham 06a], [Fergus 03], [Torralba 04], [Opelt 04] und [Fei-Fei 04].

Neben diesen großen und durch die Literatur sehr bekannten Datensätzen findet sich eine Vielzahl an kleineren Datensätzen in nahezu allen Bereichen der Objekterkennung und -lokalisierung. Ein Beispiel für einen solchen Datensatz stellt das NORB-Dataset ([Lecun 04]) dar, welches Bildserien von 50 Spielzeugen enthält, speziell auf 3D-Objekterkennung mittels Form ausgerichtet. Eine nicht mehr ganz aktuelle Liste an kleineren und größeren Datensätzen findet sich z.B. bei [Computer Vision Department, CMU 12] und [Leibe 12]. Verweise auf weitere Datensätze, auch speziellere wie z.B. unterschiedlich beleuchtete Oberflächentexturen, listet [The Ponce Group 12] auf.

Im Gegensatz zur Bildverarbeitung existiert im Bereich des Greifens neben der in dieser Arbeit vorgestellten Datenbank, nur ein weiterer Datensatz, die Columbia Grasp Database ([Goldfeder 09a]). Diese enthält Griffe, die mit Hilfe der GraspIt!-Software ([Miller 03]) auf Basis der im Princeton Shape Benchmark ([Shilane 04]) gesammelten 3D-Modelle berechnet wurden. Die Griffe berücksichtigen verschiedene Aktoren, wie Backengreifer und menschliche Fünf-Finger-Hand. Der Grund für die geringe Verfügbarkeit von vorberechneten Griffen in Form von Datenbanken ist die Abhängigkeit der Griffe von der jeweiligen Kinematik des Aktors, was zum aktuellen Zeitpunkt die Wiederverwendbarkeit solcher Daten stark einschränkt.

Nicht nur im 2D-Bereich, auch im 3D-Bereich finden sich zahlreiche Datensätze und Projekte zur Verbreitung von Geometriedaten. Ein Vertreter dieser Kategorie, der Princeton Shape Benchmark ([Shilane 04]), wurde bereits angesprochen. Dieser besteht aus ca. 1800 im Internet zusammengetragener Modelle verschiedenster Ausprägungen. Die Modelle liegen im sog. Object File Format (.off) vor, zusätzlich wird eine beschreibende Textdatei mitgeliefert, die verschiedene Charakteristika wie die Anzahl an Polygonen im Modell beinhaltet. Die Qualität der 3D-Modelle variiert sehr stark und reicht von sehr abstrakt bis sehr detailliert. Einheitliche Materialinformationen oder Texturen sind nicht vorhanden.

Eine ähnlich inhomogene, allerdings noch größere Sammlung an 3D-Modellen bietet das 3D-Warehouse von Google ([Google 12]). Hierbei handelt es sich um eine Menge von Modelldaten, die durch Zusammenarbeit vieler tausend Internetnutzer entstanden ist. Mit Hilfe der Modellierungssoftware SketchUp kann prinzipiell jeder zu dieser Kollektion beitragen. Der Datenbank liegt keine übergeordnete Struktur zu Grunde, die Nutzer können selbst Sammlungen anlegen, die allerdings allen möglichen Kriterien unterliegen können. Die einzelnen Modelle sind also auch keinen einheitlichen Kategorien zugeordnet und können beliebige Namen haben. Auch die Qualität, was Detaillierungsgrad und Präzision betrifft, variiert dementsprechend stark von Modell zu Modell, da die Modelle überwiegend von Hand erzeugt werden. Es existieren auch weitere Webplattformen zum Austausch von manuell erstellten 3D-Modellen in verschiedenen Formaten, sowohl frei zugänglich [Muldoon 12], [3DModelFree.com 12] wie auch kommerziell [TurboSquid 12] und [CGTrader 12]. Hier gelten allerdings jeweils ähnliche Einschränkungen wie für das 3D-Warehouse.

Auch im Bereich der Erkennung und Lokalisierung auf Basis von 3D-Daten gibt es Wettbewerbe und entsprechende vereinheitlichte Datensätze. Zu nennen sind hier bspw. die jeweiligen Sammlungen der „Shape REtrieval Contests (SHERC)“ [Bronstein 10], die insgesamt ca. 500 Modelle aus verschiedenen Bereichen und in unterschiedlichen Konfigurationen enthalten. Die Modelle enthalten jedoch keinerlei Material- oder Farbinformationen, da die Anwendung der gestaltbasierten Objekterkennung dies nicht erfordert.

Schließlich gibt es noch Datensätze, die auf Sensordaten beruhen und somit der Anwendung in einem Robotersystem besonders nahe stehen. [Lai 11] haben mit Hilfe des bereits angesprochenen Kinect-Sensors 300 Alltagsgegenstände mit Hilfe eines einfachen Rotationstellers aus verschiedenen Perspektiven aufgenommen. Die so erzeugten Daten bestehen aus jeweils drei Videosequenzen, die Farb- und Tiefeninformation enthalten. Die Objekte sind mit Hilfe von Hypernym-Hyponym-Relationen aus WordNet in 51 Kategorien gruppiert worden. Einzelansichten oder 3D-Rekonstruktionen der Objekte müssen aus diesen Daten vom Nutzer selbst generiert werden. Die erzeugten Daten sind sehr homogen, unterliegen allerdings den qualitativen Einschränkungen des verwendeten Kinect-Sensors. Neben den Sequenzen der Einzelobjekte werden zusätzlich 8 Videosequenzen von Alltagsszenen, bestehend aus den Objekten, zur Verfügung gestellt.

Als weitere Objektdatenbank basierend auf Sensordaten soll die Sammlung an Haushaltsgegenständen des ROS-Projekts vorgestellt werden ([Willow Garage 12]). Für diese Datensammlung wurden eine Vielzahl an handelsüblichen Haushaltsgegenständen (Gläser, etc.) mit Hilfe von Kamerabildern dreidimensional rekonstruiert. Bei rotationssymmetrischen



Objekten mit Hilfe der Objektkontur, bei allen anderen Klassen aus verschiedenen Ansichten unter Verwendung einer kommerziellen Rekonstruktionssoftware. Zusätzlich zur 3D-Geometrie wurden mit der bereits erwähnten GraspIt!-Software Griffe zu jedem Objekt vorberechnet. Die Datenbank speichert die 3D-Geometrie als einfache Punkt- und Kantenlisten innerhalb eines SQL<sup>10</sup>-Schemas. Auch hier liegen einheitliche Daten vor, die jedoch qualitativ zum Teil unter der Rekonstruktion aus Ansichten leiden, da hierbei keine konkaven Flächen rekonstruiert werden können.

#### 2.1.4 Fazit

Der erste Teil dieser Übersicht widmete sich der Frage, wo in der Servicerobotik Objektmodelle verwendet werden und wie diese aussehen. Im Bereich der visuellen Objekterkennung und -lokalisierung zeigt sich, dass für viele aktuelle Verfahren Trainingsdaten in Form von Objektansichten benötigt werden. Die Mehrheit an Greifplanungsmethoden basiert hingegen überwiegend auf 3D-Geometriedaten. Gleichzeitig wird klar, dass die Erzeugung dieser Ausgangsdaten, seien es Bilder oder Geometrie, nicht im Vordergrund dieser Arbeiten steht. Ein geeigneter Prozess hierfür wird also gesucht um den Bedarf an Trainingsdaten decken zu können. Dies führt direkt zu der Frage, welche Verfahren und Sensoren bereits existieren um geeignete Modelldaten zu erzeugen. Hier zeigt sich, dass die dreidimensionale Erfassung von Gegenständen und Personen intensiv untersucht wurde und viele kommerzielle wie auch prototypische Systeme, basierend auf unterschiedlichen Sensoren verfügbar sind. Allerdings wird

---

<sup>10</sup> Structured Query Language, eine Sprache zur Definition von Datenstrukturen in relationalen Datenbanken

die Kombination aus Bildansichten, 3D-Geometrie und möglichst automatisierter Erfassung speziell für die Anforderungen der Robotik, von keinem der aufgeführten Systeme hinreichend erfüllt. Beschränkungen sind hier entweder die Qualität der Daten in Bezug auf Präzision oder Vollständigkeit, oder aber der Digitalisierungsprozess benötigt so viel Zeit, dass die Erfassung großer Zahlen von Objekten nicht praktikabel ist. Diese Beobachtung wiederholt sich bei der Betrachtung der bis dato verfügbaren Datensätze im Bereich Objektmodelle. Zwar existiert eine Vielzahl an einzelnen Sammlungen von Bildern bzw. 3D-Daten, die jedoch in der Mehrheit große Inhomogenitäten aufweisen. Gerade die Kombination aus Geometrie und zugehörigen Objektansichten wurde bis dato noch nicht berücksichtigt. Zusammenfassend lässt sich feststellen dass, speziell im Rahmen der Servicerobotik, weder ein optimierter Digitalisierungsprozess für Objekte zur kombinierten Erfassung von Bild- und Geometriedaten existiert, noch ein ausreichend großer und homogener Datensatz verfügbar ist.

## **2.2 Repräsentation und Modellierung von Szenen**

Der erste Teil dieser Literaturübersicht ist rein objektzentriert, d.h. die aufgezeigten Modelle und Repräsentationen beschränken sich auf einzelne Objekte und deren Attribute. Gerade in Umgebungen, in denen Serviceroboter eingesetzt werden sollen, wie z.B. Haushalt oder öffentliche Einrichtungen (Bahnhöfe, Flughäfen, Supermärkte, etc.), kann jedoch aus der Umgebung selbst noch weitere Information gewonnen werden. In vom Menschen geschaffenen und für Menschen bestimmten Umgebungen sind die einzelnen Gegenstände fast immer einer bestimmten Ordnung unterworfen und stehen miteinander in Beziehung.

Schon in den 1970er Jahren haben sich Kognitionspsychologen mit Studien beschäftigt, die ergründen sollten wie der Mensch seine Umgebung wahrnimmt. [Biederman 82] führte bspw. Experimente zur Objekterkennung in Szenen und den Einfluss von Kontext auf die Erkennung beim Menschen durch. Probanden wurden dazu Fotografien von Alltagsszenen vorgelegt, in denen sie bestimmte, vorgegebene Objekte identifizieren und lokalisieren sollten. Die Abbildungen wurden im Verlauf des Experiments sowohl im Originalzustand, wie auch in einer räumlich veränderten Anordnung vorgelegt. Damit sollte untersucht werden welchen Einfluss die folgenden Klassen von Relationen auf die Erkennungsleistung haben:

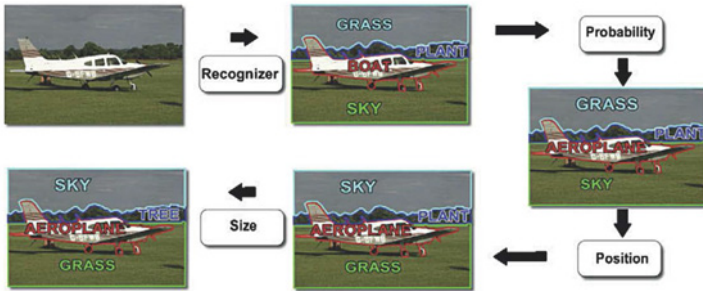
1. Interposition - Objekte unterbrechen den Hintergrund
2. Auflage - Objekte befinden sich typischerweise auf stützenden Oberflächen
3. Wahrscheinlichkeit - Objekte kommen in bestimmten Szenen häufiger vor als in anderen
4. Position - Objekte befinden sich oft an gleichen/ähnlichen Orten
5. Größe - Objekte haben in der Regel eine bestimmte Größe im Verhältnis zu anderen Objekten

Es zeigte sich, dass die Verletzung dieser Relationen (bis auf die 1.) deutliche Auswirkungen auf die Erkennungsrate beim Menschen hat. Die letzten drei werden dabei als semantische Relationen bezeichnet, weil sie vom Kontext abhängen, die ersten beiden sind physikalische Relationen. Diese Einteilung wurde im Laufe der Zeit von vielen Arbeiten im Bereich der Szenenerkennung und -klassifikation übernommen.

Die Arbeiten im Umfeld der Servicerobotik, befassen sich überwiegend mit der Klassifikation von Szenen und dem Erzeugen von entsprechenden Repräsentationen. Dabei können zwei unterschiedliche Ansätze identifiziert werden: zum einen die Erkennung auf Basis der in der Szene auftretenden Objekten und deren Konstellationen, zum anderen ganzheitliche Ansätze die das gesamte Szenenbild analysieren. Einige wenige Arbeiten beschreiben auch Mischformen, die z.B. die Szenenerkennung einsetzen, um eine lokale Objekterkennung zu verbessern.

### **2.2.1 Objekterkennung durch Kontext**

Inspiziert durch die Ergebnisse der Kognitionspsychologie wurden zahlreiche Ansätze entwickelt um das Problem der Objekterkennung und -lokalisierung durch zusätzliches Kontextwissen zu erleichtern bzw. die Ergebnisse zu verbessern und die Verfahren robuster zu gestalten. [Galleguillos 10] zeigen in ihrer Übersicht über die Verwendung von kontextuellem Wissen (u.a. räumliche Relationen) auf diesem Gebiet, welche verschiedenen Strömungen und Ansätze hier existieren. Die bereits angeführte Einteilung der Objektrelationen von Biedermann wird hier weiter unterteilt und es wird unterschieden zwischen semantischem Kontext (funktional zueinander passende Objekte in der selben Szene), räumlichem Kontext (in welchen relativen Posen treten Objekte in der Szene auf) und Skalierungskontext (Objekte haben typische Größen - absolut aber vor allem im Verhältnis zu anderen Objekten). Schließlich kommen [Galleguillos 10] zu dem Schluss, dass die Integration des Kontextwissens in die Objekterkennung über Verfahren des maschinellen Lernens geschieht.



**Abb. 2.6.** Idealisiertes System zur Objektkategorisierung. (Quelle: [Galleguillos 10]).

In die Kategorie der Verfahren, die mit Hilfe von lokalen Merkmalen Objekterkennung betreiben und diese durch Kontextwissen anreichern, zählt die Arbeit von [Santosh K. Divvala 09]. Die Arbeit untersucht den Beitrag des Kontextwissens um zu einer quantitativen Aussage zu gelangen, wie stark aktuelle Verfahren zur Objekterkennung von solchem zusätzlichen Wissen profitieren können. Als Grundlage dient der Datensatz der PASCAL VOC 2008<sup>11</sup>, um einen möglichst aussagekräftigen Vergleich der beiden Methodiken zu erlauben. Kontext wird in dieser Arbeit vor allem als Umfeld auf verschiedenen Ebenen verstanden, lässt sich aber bei der eigentlichen Anwendung wieder auf die Relationen von Biedermann zurückführen. Die Autoren kommen zu dem Ergebnis, dass die Verwendung von Kontextwissen auf allen Ebenen deutliche Verbesserungen der Objekterkennung mit sich bringt. Dies deutet darauf hin, dass was für den Menschen gilt, sich offenbar also auch auf die maschinelle Perzeption übertragen lässt. Zu ähnlichen Ergebnissen gelangen [Costea 11], deren Arbeit sich mit der Kommentierung von Bildern mittels Szenen- und Objekter-

<sup>11</sup> <http://pascal.in.ecs.soton.ac.uk/challenges/VOC/voc2008/>

kennung befasst. Zunächst wird unabhängig voneinander eine Objekt- und eine Szenenerkennung basierend auf lokalen Merkmalen durchgeführt. Die Ergebnisse werden schließlich kombiniert, indem szenenspezifische Auftretenswahrscheinlichkeiten für die Objekte ermittelt werden um so Unsicherheiten in der Objekterkennung aufzulösen. Die experimentellen Ergebnisse zeigen, dass sich die Erkennungsrate für Objekte um ca. 10 % durch das hinzugenommene Szenenwissen verbessert. Auch [Kollar 09] nutzen Auftretenswahrscheinlichkeiten von Objekten in bestimmten Szenen um die Erkennung der einzelnen Objekte zu verbessern. Hierbei bewegt sich der Roboter durch die Szene, erstellt dabei eine Karte und lokalisiert und erkennt dabei gleichzeitig Objekte. Deren Posen werden in die entstehende Karte eingetragen. Die einzelnen Objektbeobachtungen werden dabei durch ein probabilistisches Modell dargestellt, welches es ermöglicht Hintergrundwissen in Form von Wahrscheinlichkeitsverteilungen zu integrieren. Die benötigten Verteilungen über das gemeinsame Auftreten von Gegenständen entstehen durch Evaluierung von kommentierten Bildern aus der Flickr<sup>12</sup>-Datenbank. Dabei wird die Annahme getroffen, dass zwei Objekte in räumlicher Nähe zueinander auftreten, sobald sie gemeinsam in einem Bild in diesem Datensatz gefunden werden. Abgesehen von den Auftretenswahrscheinlichkeiten werden keine weiteren Relationen definiert.

### **2.2.2 Szenenerkennung durch globale Merkmale**

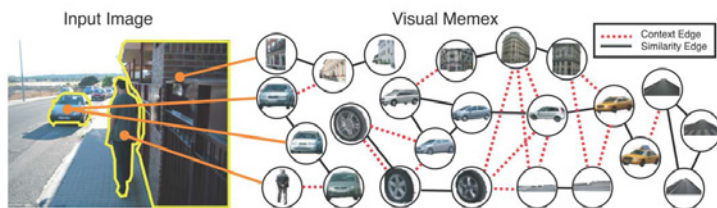
Neben der Verwendung des Szenenkontextes zur Verbesserung einer Objekterkennung, dient dieser bzw. ein Szenenmodell natürlich auch dazu,

---

<sup>12</sup> <http://www.flickr.com>

bestimmte Szenen wieder zuerkennen bzw. allgemein zu klassifizieren. [Oliva 01] führen zu diesem Zweck den sog. *Spatial Envelope* ein, eine Repräsentation von Szenen auf Basis von 2D-Bilddaten. Hierfür wurden zunächst mit Hilfe von Versuchspersonen empirisch fünf unterschiedliche Kriterien zur Beschreibung von Szenen entwickelt: Naturalness (Natürlichkeit), Openness (Weitläufigkeit), Roughness (Körnigkeit), Expansion (Ausdehnung) sowie Ruggedness (Rauheit). Für ein gegebenes Bild wird die Ausprägung dieser Kriterien nun mit Hilfe einer Hauptkomponentenanalyse auf der Fouriertransformation des Eingabebildes berechnet. Ein Klassifizierungssystem kann dann auf Basis klassifizierter Trainingsdaten eingelernt werden um in diesem mehrdimensionalen Merkmalsraum neue Datensätze zu erkennen. Bei diesem Ansatz findet also keine explizite Segmentierung bzw. Erkennung von Einzelobjekten statt, sondern die Szene wird in ihrer Gesamtheit evaluiert. Ein Mischansatz zwischen Szenen- und Objekterkennung findet sich bei [Torralba 03]. Hier wird ebenfalls versucht die Objekterkennung durch Szenenerkennung zu verbessern. Allerdings betrachten und klassifizieren die Autoren die Szene global mit einem, dem *Spatial Envelope* sehr ähnlichen Ansatz. Anschließend wird dann eine Objekterkennung auf Basis von lokalen Merkmalen durchgeführt. Der Szenenkontext, der durch die vorangegangene Szenenerkennung zur Verfügung steht, wird nun genutzt um zum einen nur nach bestimmten Objekten zu suchen (Suchraumreduktion durch Verwendung von Auftretenswahrscheinlichkeiten), bzw. nur in bestimmten Regionen nach Objekten zu suchen (Suchraumreduktion durch Verwendung von szenenspezifischen Auftretensregionen pro Objektklasse). Die Generierung des dazu notwendigen Hintergrundwissens in Form von Wahrscheinlichkeitsverteilungen geschieht über die Auswertung von kommentierten Trainingsbildern.

[Quelhas 07] stellen eine weitere Arbeit zur Szenenerkennung und -klassifikation auf Basis einer globalen Szenenbetrachtung vor. In einem Eingangsbild werden lokale Merkmalspunkte bestimmt und aus diesen Deskriptoren (z.B. SIFT<sup>13</sup>) berechnet. Es folgt eine Quantifizierung der Deskriptoren mit anschließender Histogrammbildung (mittels K-means Clustering). Die einzelnen Szenenkategorien werden dann durch Anwendung von Probabilistic Latent Semantic Analysis (PLSA) und Expectation Maximization (EM) auf vorklassifizierten Trainingsdaten eingelesen. Auch hier wird also keine Segmentierung in Einzelobjekte durchgeführt, sondern die Szene global über die extrahierten lokalen Merkmale beschrieben.



**Abb. 2.7.** Der Visual Memex als beispielbasierte Repräsentation von Szenen. (Quelle: [Malisiewicz 09]).

Alle bisherigen Ansätze basieren auf expliziten Kategorien für Objekte und Szenen. Die Verwendung von expliziter Kategorisierung begründet sich in den Annahmen, dass Objekte bestimmten Kategorien zugeordnet werden können und diese Zuordnung gleichzeitig eine vorteilhafte Abstraktion mit sich bringt. Es ist jedoch auch denkbar auf explizite Kate-

<sup>13</sup> Scale Invariant Feature Transform, [Lowe 04]

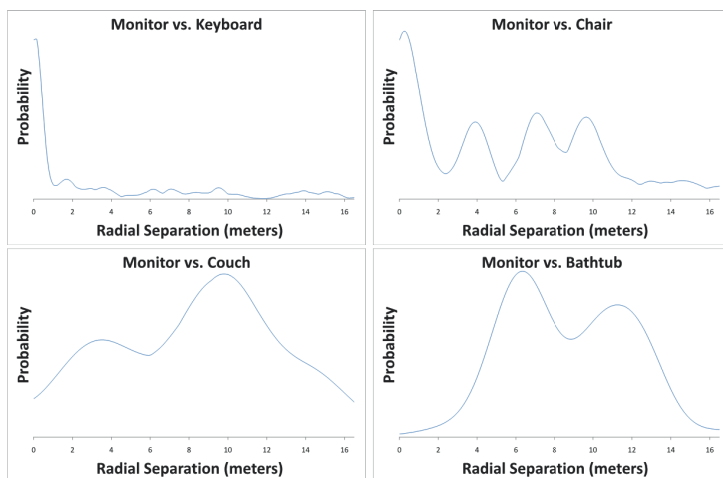


gorien zu verzichten und stattdessen alle beobachteten Beispielinstanzen anhand von Ähnlichkeiten bzw. gemeinsamem Auftreten zu verknüpfen, wie es [Malisiewicz 09] mit dem sog. *Visual Memex* vorschlagen. Dieses Modell ist formal ein Graph  $G = (V, E_S, E_C, \{D\}, \{f\})$ , wobei  $V$  die Knoten,  $E_S$  die Ähnlichkeitskanten,  $E_C$  die kontextuellen Kanten,  $\{D\}$  eine Menge von Ähnlichkeitsfunktionen und  $\{f\}$  die räumlichen Merkmale der Objekte bezeichnen. Das gesamte Szenen- und Objektmodell ist also beispielbasiert und verknüpft lediglich einzelne Beobachtungen ohne explizit zu kategorisieren. Bei einem neuen Beispiel kann dann diesen Verknüpfungspfad gefolgt werden um entsprechende (in irgendeiner Form ähnliche) Objekte zu finden, was wesentlich näher an der Vorgehensweise des Menschen orientiert ist.

### 2.2.3 Objektsuche durch Szenenkontext

Neben der Erkennung von Objekten und der Kategorisierung von Szenen, kann der räumliche Kontext von Objekten innerhalb einer abgegrenzten Umgebung, auch zur zielgerichteten Suche nach bestimmten Objekten genutzt werden. [Aydemir 10] implementieren eine solche Objektsuche mit Hilfe von räumlichen Relationen. Vorgestellt wird im Besonderen die Relation „ist auf“, deren Formulierung auch im weiteren Verlauf dieser Arbeit verwendet wird (vgl. 6.6.1). Der vorgestellte Algorithmus bedient sich einer probabilistischen Modellierung der Erkennung eines Objekts in einer zu untersuchenden Region. Mit Hilfe dieses Modells wird anschließend die beste, nächste Pose des Roboters ermittelt um das Zielobjekt zu lokalisieren. Dies bedeutet, dass hier eine *indirekte Suche* durchgeführt wird, d.h. das Auffinden eines bestimmten Objekts erfolgt mit Hilfe anderer Objekte, die sich leicht finden lassen und von denen bekannt ist, dass sie sich

häufig in der näheren Umgebung des Zielobjekts befinden. Im Speziellen wird hier die bereits erwähnte „ist-auf“-Relation verwendet um den Suchraum günstig einzuschränken.



**Abb. 2.8.** Dichteschätzungen für räumliche Relationen zwischen Objektklassen in künstlich erstellten 3D-Szenen. (Quelle: [Fisher 10]).

Während Aydemir et al. die szenenspezifischen räumlichen Relationen zwischen Objekten ausnutzen, um in realen Szenen mit Hilfe eines Bildverarbeitungssystems nach vorgegebenen Zielobjekten zu suchen, nutzen [Fisher 10] ähnliche Relationen als zusätzliche Information beim Problem der 3D-Objektsuche. Ihr Ziel ist es, in einer vorgegebenen, künstlich modellierten Szene, das für einen bestimmten Raumbereich passende Objekt zu finden. Dazu wurden die räumlichen Relationen in einer großen Zahl (4876) von künstlich erstellten 3D-Szenen von Räumen (aus Google Ware-

house<sup>14</sup>), zwischen den Objekten untersucht, u.a. unter Zuhilfenahme der vorhandenen Kategorisierungen der Szenen und Objekte. Die grundlegende Annahme ist der bereits erwähnte *Visual Memex* von [Malisiewicz 09]: werden zwei Objekte  $f$  und  $g$  in Szene  $A$  beobachtet, und ein zu  $f$  ähnliches Objekt  $f'$  in Szene  $A'$ , dann ist ein Objekt  $g'$  in der Szene an der Stelle passend, an der die Relation zu  $f'$  ähnlich der Relation zwischen  $f$  und  $g$  ist. Das Auftreten von Objektpaaren wird als Beobachtung modelliert, wobei diese die räumliche Relation zwischen den Objekten, die Größe, die Form und die Textur beider Objekte beinhaltet. Als räumliche Relationen werden lediglich die Höhenunterschied (in  $z$ -Richtung), sowie die Distanz in der  $x$ - $y$ -Ebene zwischen den Zentren der Objektboundingboxen betrachtet. Die Ähnlichkeit zwischen solchen Distanzen wird mit Hilfe eines Gauß'schen Kernels modelliert. Zusätzlich wird die geometrische Ähnlichkeit zwischen zwei Objekten ermittelt. Aus den Relationen lassen sich mit Hilfe der Ausgangsdaten Wahrscheinlichkeiten für das Auftreten von Objekten in spezifischen Kombinationen berechnen. Diese werden dann zur Beantwortung einer Suchanfrage für ein bestimmtes Raumvolumen verwendet. Auch in dieser Arbeit finden also die Relationen von [Biederman 82] Anwendung, wobei jedoch lediglich sehr einfache Abstandsberechnungen als Relationen eingesetzt wurden und keine komplexere Funktionen wie bspw. „ist auf“ oder „ist neben“.

Schließlich kann Objektsuche in einer Alltagsumgebung auch auf Basis abstrakten Hintergrundwissens erfolgen. [Saito 11] nutzen Allgemeinwissen über solche Umgebungen, welches von menschlichen Nutzern über die OMICS (Open Mind Indoor Common Sense Projekt<sup>15</sup>) Internetplattform gesammelt wurde. Dieses stellt u.a. die in dieser Arbeit benötigten

<sup>14</sup> <http://sketchup.google.com/3dwarehouse/>

<sup>15</sup> <http://openmind.hri-us.com>



bung, aus dem zunächst planare Ebenen extrahiert werden, die anschließend klassifiziert werden. Ziel der Klassifikation ist die Erkennung von Türen und Schubläden. Mögliche Kandidaten hierfür werden durch Betrachtung von 3D- und 2D-Merkmalen der segmentierten Ebenen ermittelt und anschließend durch das Robotersystem manipuliert. Aus der Interaktion lassen sich Bewegungsfreiheitsgrade extrahieren, die Rückschlüsse auf die Objektkategorie zulassen. Erkannte Möbel werden dann in die semantische Karte eingetragen, die zwar einen hierarchischen Aufbau besitzt, jedoch keine weiteren Relationen zwischen den Objekten modelliert. Ähnlich zu diesem Ansatz, jedoch ohne Interaktion mit der Szene, ist die Arbeit von [Koppula 11] einzuordnen. Ausgangspunkt ist ebenfalls eine 3D-Punktwolke der betrachteten Szene, die mit einem gängigen SLAM<sup>16</sup>-Verfahren erzeugt wird. Auch hier wird diese in planare Teile segmentiert, die es dann zu Klassifizieren gilt. Neben den dafür hauptsächlich verwendeten objektzentrischen Merkmalen der Punktwolke (wie etwa Normalen), werden auch räumliche Relationen zwischen den segmentierten Teilen betrachtet:

- Horizontaler Abstand zwischen Mittelpunkten
- Vertikaler Abstand zwischen Mittelpunkten
- Winkel zwischen Normalen
- Winkelunterschied zur Vertikalen
- Abstand zwischen den nächsten Punkten
- Relative Position zur Kamera

---

<sup>16</sup> Simultaneous Localization and Mapping

Die experimentelle Validierung ergibt auch hier eine Verbesserung der Klassifikationsrate durch Hinzunahme dieses kontextuellen Wissens, die vorgestellten räumlichen Relationen sind jedoch auch hier sehr einfach gehalten und abstrahieren die geometrischen Verhältnisse dementsprechend wenig. Eine weitere Arbeit, die sich mit dem Aufbau einer kategorisierten 3D-Karte aus entsprechenden Sensordaten befasst, beschreiben [Tenorth 10]. Das darin vorgestellte System KnowRob Map, dient zur Erstellung von Umgebungskarten durch Verknüpfung von räumlichen Informationen über Objekte mit enzyklopädischem Wissen. Aus den gegebenen, segmentierten Sensordaten wird eine Menge von Instanzen zu den im Hintergrundwissen vorhandenen Objektklassen erzeugt. Für diese Abbildung existiert ein Satz spezieller Regeln. Zusätzlich werden auch probabilistisch modellierte Relationen in die semantische Karte integriert, die Quelle dieser Verteilungen ist das bereits erwähnte OMICS-Projekt (s. [Kochenderfer 03]). Dort wird Allgemeinwissen über Haushaltsszenarien von menschlichen Benutzern gewonnen, die dazu vorgefertigte und automatisch generierte Sätze vervollständigen, die dann in Relationen zerlegt werden. Ein Beispielsatz ist etwa: „You generally find a frying pan in the same room as a ...“. Allerdings werden räumliche Relationen wie „neben“ oder „in der Nähe von“ nicht spezifiziert, können also nicht direkt von einem Sensorsystem evaluiert werden.

### **2.2.5 Fazit**

Die Betrachtung der verschiedenen Arbeiten zur Modellierung von Szenen im Kontext der Servicerobotik zeigt, dass Wissen über die typischen Konstellationen in denen Objekte im Alltag auftreten, für Robotersysteme in unterschiedlichen Anwendungen von Nutzen sein kann. Als rele-

vante Informationen in Szenenmodellen können die Wahrscheinlichkeit, mit der sich Objekte in bestimmten Szenen finden und die räumlich-geometrischen Relationen zwischen den einzelnen Objekten identifiziert werden. Die Gewinnung dieser Informationen geschieht dabei entweder durch die Auswertung manuell vorklassifizierter Trainingsdaten, oder die Ausnutzung explizit formulierten Expertenwissens. Räumliche Relationen und ähnliches Kontextwissen werden jedoch nicht exklusiv genutzt, sondern nur unterstützend neben anderen Merkmalen. Eine Untersuchung, welche Ergebnisse bei der ausschließlichen Nutzung von Objekt-Objekt-Relationen erzielt werden können liegt derzeit noch nicht vor.





## Grundlagen zur Abstandsmessung

Dieses Kapitel erläutert die technischen Grundlagen und Verfahren der im weiteren Verlauf der Arbeit verwendeten Sensoren. Von Interesse sind daher Methoden zur Abstandsmessung, die eine dreidimensionale Abtastung von Gegenständen und Szenen ermöglichen. Zum besseren Verständnis und zur Einordnung werden neben den genutzten Triangulationsverfahren noch weitere Möglichkeiten der Abstandsmessung grundlegend vorgestellt.

Grundsätzlich kann im Bereich der Abstandsmessung zwischen aktiven und passiven Verfahren unterschieden werden. Aktive Verfahren bringen aktiv ein Signal in die Umwelt ein und messen die Veränderung dieses Signals in der Umwelt. Passive Verfahren nutzen lediglich die natürlich vorhandenen Größen und werten diese aus. Beide Verfahren liefern als Ergebnis den Abstand von einem oder mehreren Punkten der Umwelt zum Sensor. Sie unterscheiden sich in der erzielbaren Genauigkeit, den möglichen Auflösungen sowie der benötigten Zeit für die Messung.

### 3.1 Passive Verfahren zur Abstandsmessung

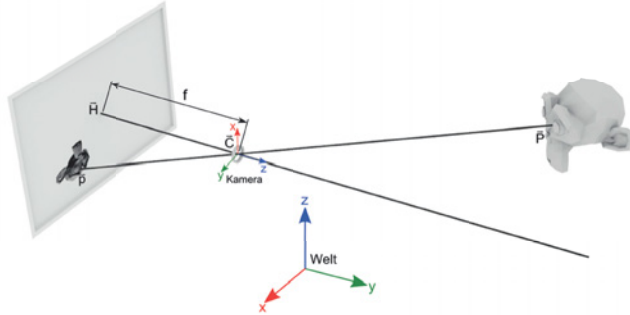
Wie bereits erwähnt bedienen sich passive Verfahren zur Abstandsmessung der natürlichen, in der Umwelt vorhandenen physikalischen Größen. Stellvertretend soll an dieser Stelle die passive Triangulierung auf Basis von Kamerabildern vorgestellt werden.

Grundlage jeder Triangulierung ist das Anpeilen des selben Punktes im Raum von verschiedenen Positionen aus. Ist die räumliche Beziehung der Messpunkte zueinander sowie die Abbildungseigenschaften der jeweiligen Sensoren bekannt, kann die Position des angepeilten Punktes im Raum berechnet werden. Um die räumlichen Verhältnisse der Messpunkte und die Abbildungseigenschaften zu ermitteln muss zunächst eine Kalibrierung durchgeführt werden. Im weiteren folgt eine kurze Zusammenfassung der Kalibrierung für ein Stereokamerasystem bestehend aus zwei Kameras (vgl. [Zhang 99], [Tsai 87], [Azad 08b] und [Willow Garage 11]).

#### 3.1.1 Kamerakalibrierung

Um die Abbildungseigenschaften eines Kamerasensors zu ermitteln muss zunächst ein Kameramodell erstellt werden. In der Literatur hat sich das sog. *Lochkameramodell* als erste Näherung etabliert.

Dieses Modell beschreibt die Projektion eines Punktes im dreidimensionalen Raum auf eine zweidimensionale Bildebene:



**Abb. 3.1.** Einfaches Lochkameramodell mit Brennweite  $f$ , Bildhauptpunkt  $\mathbf{H}$ , Projektionszentrum  $\mathbf{C}$ , projiziertem Punkt ( $\mathbf{P}$  und  $\mathbf{p}$ ) sowie Welt- und Kamerakoordinatensystem.

$$\mathbf{p} = A \cdot T \cdot \mathbf{P}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R \\ t \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.1)$$

Dabei ist  $\mathbf{P}$  der Punkt im Raum in Weltkoordinaten und  $\mathbf{p}$  der projizierte Punkt in Bildkoordinaten. Bei dieser Projektion wird  $\mathbf{P}$  zunächst durch die Transformationsmatrix  $W$ , bestehend aus einer Rotationsmatrix  $R$  und einem Translationsteil  $t$  in Kamerakoordinaten transformiert. Anschließend erfolgt die Abbildung auf die Bildebene durch die Projektionsmatrix  $A$ . Die Transformationsmatrix enthält 12 Parameter, die die Kamerapose im Raum beschreiben, während die Projektionsmatrix 4 Parameter enthält, die die Brennweite in  $x$ - und  $y$ -Richtung ( $f_x, f_y$ ) sowie den Bildhauptpunkt

$(c_x, c_y)$  beschreiben. Insgesamt ergeben sich also  $12 + 4 = 16$  zu bestimmende Parameter.

Leider ist dieses Kameramodell nicht geeignet um reale Kamerasensoren zu beschreiben, da hier üblicherweise Verzerrungen durch die Verwendung von Linsen in den Objektiven entstehen. Um diese Gegebenheiten zu berücksichtigen müssen weitere Parameter eingeführt werden. Gleichung 3.1 lässt sich für Weltpunkte mit  $z \neq 0$  dann folgendermaßen umschreiben:

$$\begin{aligned}\bar{x} &= \frac{x}{z}, \bar{y} = \frac{y}{z} \\ u &= f_x \cdot \bar{x} + c_x, v = f_y \cdot \bar{y} + c_y\end{aligned}\tag{3.2}$$

Sollen nun radiale und tangentielle Linsenverzerrungen berücksichtigt werden, ergeben sich folgende Gleichungen:

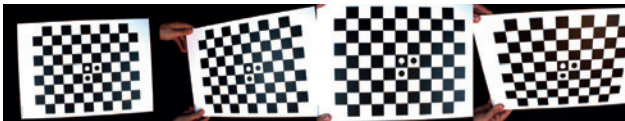
$$\begin{aligned}\bar{x} &= \frac{x}{z}, \bar{y} = \frac{y}{z} \\ \check{x} &= \bar{x} \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 \bar{x} \bar{y} + p_2 (r^2 + 2\bar{x}^2) \\ \check{y} &= \bar{y} \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 \bar{x} \bar{y} + p_1 (r^2 + 2\bar{y}^2) \\ &\text{jeweils mit } r^2 = \bar{x}^2 + \bar{y}^2 \\ u &= f_x \cdot \check{x} + c_x, v = f_y \cdot \check{y} + c_y\end{aligned}\tag{3.3}$$

Hierbei bezeichnen  $k_1, k_2, k_3$  die radialen Verzerrungskoeffizienten und  $p_1, p_2$  die tangentialen Verzerrungskoeffizienten.

Es kann in diesem erweiterten Lochkameramodell zwischen sog. *intrinsischen* und *extrinsischen* Parametern unterschieden werden. Brennweite, Bildhauptpunkt und Verzerrungskoeffizienten sind Parameter, die die interne Beschaffenheit der jeweiligen Kamera und des Objektivs beschrei-

ben, sind also intrinsisch. Die Pose der Kamera und damit die Koeffizienten der Transformationsmatrix beschreiben die externe Lage der Kamera in der Welt, diese Parameter sind also extrinsisch.

Um nun die verschiedenen Parameter des Modells bestimmen zu können muss eine Kalibrierung durchgeführt werden. Dabei gilt es ein bekanntes, also vermessenes Objekt zu betrachten und die durch den Kamerasensor durchgeführte Abbildung auszuwerten. Üblicherweise wird dazu ein planares Objekt verwendet auf das ein Schachbrettmuster aufgebracht ist. Dieses wird in verschiedenen Entfernungen und Winkeln zur Kamerapositioniert und jeweils ein Bild aufgenommen, s. Abb. 3.2. In den Bildern wird dann nach den Kreuzungspunkten des Schachbrettmusters gesucht. Mit dem Wissen über die realen Abstände, sowie die Anordnung der Eckpunkte des Musters und den so gewonnenen Bildpunkten, können dann verschiedene Gleichungssysteme aufgestellt werden, deren Auflösung dann die gesuchten Parameter ergibt. Wird solch eine Kalibrierung

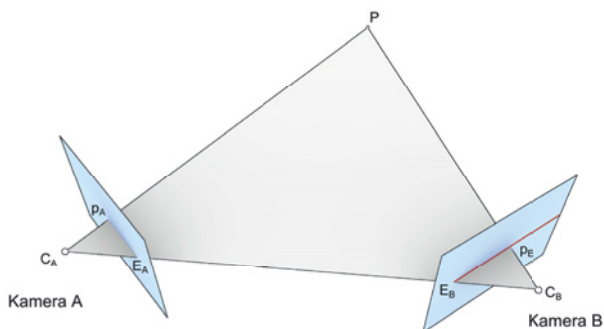


**Abb. 3.2.** Auszug aus einer Serie von Aufnahmen zur Kalibrierung mit Hilfe eines Schachbrettmusters. Quelle: [Kasper 12b]

mit zwei oder mehr Kameras in einer Stereoanordnung durchgeführt, kann damit dann auch die relative Pose der einzelnen Kameras zueinander bestimmt werden.

### 3.1.2 Triangulierung

Nach der Kalibrierung sollte die relative Position der einzelnen Sensoren zueinander bekannt sein. Um nun eine Triangulierung durchführen zu können, die es ermöglicht die Raumposition eines bestimmten Punktes bestimmen zu können, muss dieser Raumpunkt in allen Sensorbildern gefunden werden. Dies bedeutet, dass zunächst ein Korrespondenzproblem gelöst werden muss. Ausgehend von einem Bildpunkt im ersten Sensorbild müssen nun die anderen Bilder nach einem passenden Korrespondenzbildpunkt durchsucht werden. Um den Suchraum einzuzugrenzen kann man sich der sog. *Epipolargeometrie* bedienen. Interessante Ergebnisse



**Abb. 3.3.** Darstellung der Epipolargeometrie von zwei Kameras A und B. Die Epipole  $E_A$  und  $E_B$  und der Weltpunkt  $P$  bestimmen die Lage der Epipolarebene und damit der Epipolarlinie im Bild der Kamera B (rot).

der Epipolargeometrie für die Einschränkung des Suchraums sind die Epipole und die Epipolarlinien (vgl. Abb. 3.3). Die Epipole ( $E_A$  und  $E_B$ ) entstehen dabei durch die Abbildung des Abbildungszentrums ( $C_A$  bzw.  $C_B$ )

einer Kamera auf die Bildebene der anderen Kameras. Der Epipol kann dabei außerhalb der Bildebene liegen oder bei absolut parallelen Bildebenen im Unendlichen und ist nur abhängig von der relativen Position der Kameras.

Die Epipolarlinie entsteht nun abhängig vom betrachteten Bildpunkt  $p_A$  in Kamera A. Bei der Abbildung eines Raumpunktes in die Bildebene geht zwar die Tiefeninformation verloren, es kann jedoch die Richtung in der der Punkt liegen muss rekonstruiert werden. Bildet man nun diese Halbgerade in die andere Kamera ab, erhält man die Epipolarlinie dieses Punktes. Dabei gilt, dass alle Epipolarlinien durch den Epipol gehen und der gesuchte Korrespondenzpunkt daher auf der Epipolarlinie zu finden sein muss.

Auch wenn der Suchraum durch Nutzung der Epipolargeometrie stark eingeschränkt werden kann, so muss für die verbleibenden Kandidaten immer noch eine Korrespondenzfindung durchgeführt werden. Um nun zu entscheiden ob ein Bildpunkt in Bild B mit einem Punkt in Bild A korrespondiert, wird eine Kostenfunktion aufgestellt und für jeden Pixelkandidaten berechnet. Der Pixel mit den geringsten Kosten wird dann als Korrespondenzpunkt angenommen. Eine Kostenfunktion ist die sog. Summe der quadrierten Differenzen (SSD):

$$\sum_{(i,j) \in W} (I_A(i, j) - I_B(x + i, y + j))^2 \quad (3.4)$$

Die Berechnung dieser Differenzen erfolgt innerhalb eines Fensters um den Pixelkandidaten (oft 3x3 oder 5x5 Pixel). In obiger Formel bezeichnet  $(x, y)$  den Versatz vom Referenzbild A zum Zielbild B. Eine Über-

sicht und ein qualitativer Vergleich von Korrelationsverfahren findet sich in [Scharstein 02].

Wurden nun zwei korrespondierende Punkte gefunden, kann damit die Tiefeninformation des Ursprungspunkt rekonstruiert werden. Dazu werden mit Hilfe des Kameramodells die Geraden der Korrespondenzpunkte durch die Projektionszentren gebildet und diese geschnitten. Da die Korrespondenzfindung aber meist fehlerbehaftet und durch weitere Ungenauigkeiten beeinflusst ist, schneiden sich diese beiden Geraden in der Regel nicht in einem Punkt, sondern sind windschief. In diesem Fall wird der Mittelpunkt der Lotrechten auf die Geraden als Rekonstruktionspunkt verwendet. Die konkrete Berechnung kann dann z.B. nach [Gockel 06a] erfolgen:

Die beiden rekonstruierten Geraden

$$g : \mathbf{x} = \mathbf{a} + r\mathbf{u}$$

$$h : \mathbf{x} = \mathbf{b} + s\mathbf{v}$$

können umgeformt und gleichgesetzt werden zu:

$$\mathbf{a} + r\mathbf{u} = \mathbf{b} + s\mathbf{v}$$

$$\mathbf{a} - \mathbf{b} = s\mathbf{v} - r\mathbf{u}$$

Dies ergibt ein überbestimmtes Gleichungssystem, welches nach Auflösung mittels der Methode der kleinsten Quadrate nach Gauß, die Berechnung des gesuchten Punktes  $\mathbf{p}$  ermöglicht:

$$\mathbf{p} = \frac{\mathbf{a} + r\mathbf{u} + \mathbf{b} + s\mathbf{v}}{2} \quad (3.5)$$



## 3.2 Aktive Verfahren zur Abstandsmessung

Aktive Verfahren unterscheiden sich von den bereits vorgestellten passiven Verfahren dadurch, dass sie aktiv ein Signal in die Szene einbringen und dessen Reflektion messen. Hierbei können zwei grundlegende Verfahren unterschieden werden:

- Laufzeitverfahren, auch Time-of-flight (TOF) genannt
- Triangulationsverfahren, auch strukturiertes Licht genannt

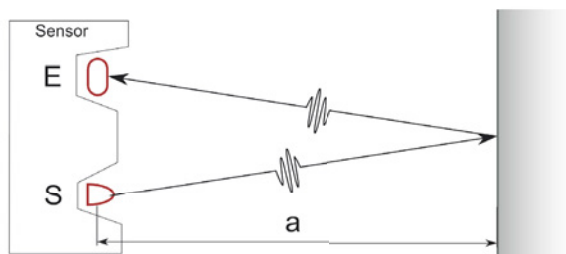
Da im weiteren Verlauf Sensoren eingesetzt wurden, welche mit strukturiertem Licht arbeiten, soll auf die Laufzeitverfahren hier nur kurz eingegangen werden.

### 3.2.1 Laufzeitverfahren

#### Time-of-Flight-Verfahren

Bei der Laufzeitmessung wird die Zeit zwischen dem Aussenden eines Signalimpulses und der Registrierung der Reflektion dieses Impulses gemessen. Aus der Signalgeschwindigkeit und dieser Laufzeit kann dann die Entfernung des Reflektionspunktes bestimmt werden. Abbildung 3.4 zeigt den schematischen Aufbau eines solchen Sensors. Der Abstand vom Sensor zum angemessenen Punkt lässt sich dann aus der Signalgeschwindigkeit  $c$  und der gemessenen Zeitdifferenz zwischen Aussenden und Empfang des reflektierten Signals  $\Delta t$  berechnen:

$$a = \frac{c \cdot \Delta t}{2} \quad (3.6)$$



**Abb. 3.4.** Prinzip der Abstandsmessung mittels Laufzeitbestimmung. Das Signal wird vom Sender S ausgesandt, an einem Hindernis reflektiert und vom Empfänger E aufgenommen.

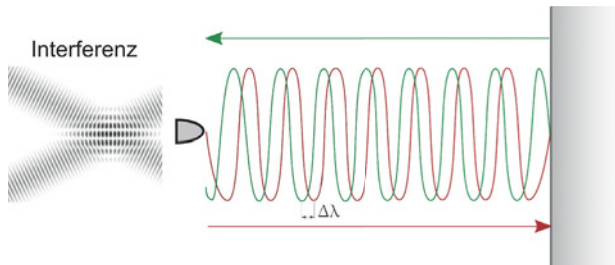
Beispielsensoren, die dieses Prinzip einsetzen, sind die Laserscanner der Firma Sick AG (z.B. LMS400, s. [SICK AG 11]) und die Swisstranger der Firma Mesa Imaging AG. Vorteile dieser Technik liegen in der relativ einfachen technischen Umsetzbarkeit und, je nach Auflösung der Zeitmessung, guter Genauigkeit. Nachteilig ist, dass sich laufzeitbasierte Sensoren nur schwierig flächig aufbauen lassen. Hierbei ist entweder mit einer geringen Auflösung zu rechnen (z.B. Swisstranger SR4000: 176x144 Pixel, s. [MESA Imaging AG 12]) oder mit einer entsprechend langen Scandauer durch Schwenken oder Rotieren des Sensorkopfes, was zu Problemen bei dynamischen Szenen führen kann. Typische laufzeitbasierte Sensormesssysteme erreichen Genauigkeiten im Millimeterbereich (Swisstranger SR4000: +/-10mm).

### **Interferometrische Verfahren**

Die Entfernungsmessung über die Phasenlage bedient sich des Phänomens, dass bei der Reflektion eines wellenförmigen Signals eine Phasenverschiebung im Vergleich zum Ursprungssignal auftritt. Diese kann z.B. durch Auswertung von Interferenzerscheinungen ermittelt werden. Die Vorteile des Phasenverschiebungsverfahrens liegen in einer einfachen technischen Umsetzung und hohen Genauigkeiten, da die Auflösung von der Wellenlänge des verwendeten Signals abhängt, bei sichtbarem Licht also im Nanometerbereich liegen kann. Da die Phasenverschiebung aber nur innerhalb einer Periode eindeutig ist muss für eine Messung von größeren Distanzen auch größerer technischer Aufwand betrieben werden. Zudem ist die maximal messbare Entfernung durch die Tatsache beschränkt, dass der Laser kontinuierlich strahlen muss und daher weniger leistungsstark ausgelegt ist (aus thermischen Gründen), als ein Puls laser wie er z.B. bei Laufzeitsensoren zum Einsatz kommt. Zudem kann mit einfacher Interferenzmessung lediglich eine Relativdistanz gemessen werden und kein absoluter Abstand.

#### **3.2.2 Triangulationsverfahren**

Neben den aktiven Verfahren zur Entfernungsmessung, denen eine Laufzeitmessung bzw. Phasenverschiebung zu Grunde liegt, gibt es noch Verfahren basierend auf Triangulation. Dazu wird ein mehr oder weniger komplexes Muster (Laserlinie bzw. Streifen- oder Rauschmuster) in die Szene projiziert und von einer Kamera beobachtet. Die Rekonstruktion der Tiefeninformation läuft nun ähnlich der passiven Triangulation ab, es



**Abb. 3.5.** Grundprinzip der Interferometrie. Das reflektierte Signal weist eine Phasenverschiebung auf, die durch Betrachtung der Interferenz gemessen werden kann. Zusammen mit der Wellenlänge  $\lambda$  lässt sich eine Relativdistanz bestimmen.

muss wieder das Korrespondenzproblem gelöst werden und die relativen Posen von Kamera und Musterprojektor müssen bekannt sein. Im Folgenden werden die beiden Verfahren vorgestellt, die im weiteren Verlauf der Arbeit zur Akquisition der 3D-Daten eingesetzt wurden. Eine detailliertere Übersicht über verschiedene Musterprojektionsverfahren findet sich in [Gockel 06a].

### Projektion von Linienmustern

Ein sehr einfach zu realisierendes Muster für die Distanzmessung mittels aktiver Triangulation ist eine einzelne Linie. Diese kann zum Beispiel durch einen Laser erzeugt werden, dessen Strahl über eine Linse aufgefächert und so zur Linie wird. Laserlicht bietet hohe Intensität und gute Fokussierbarkeit, was die Erkennung der Linie auf unterschiedlichen Oberflächen vereinfacht und gleichzeitig ein scharf abgegrenztes Muster erzeugt. Weiterhin ist es möglich mit Hilfe von Schablonen vor einem

Projektor Streifenmuster aus Schwarz-Weiß-Übergängen zu erzeugen, die ebenfalls als Linienprojektion verwendet werden können.

Um mit einem solchen Linienmuster nun Tiefeninformationen zu generieren, muss der Aufbau aus Linienprojektor und Kamera zunächst kalibriert werden. Der Linienprojektor erzeugt eine oder mehrere Ebenen, die den zu vermessenden Gegenstand schneiden. Die Lage dieser Ebenen relativ zur Kamera wird durch die Kalibrierung bestimmt. Diese erfolgt z.B. durch Aufnahme des projizierten Musters mit dem Bildaufnehmer und anschließender Korrespondenzfindung zum Ausgangsmuster (vgl. [Gockel 06a] und [Azad 03]).

Ist die Lage des Projektors relativ zum Bildaufnehmer bekannt, muss für einen Bildpunkt auf einer der projizierten Linien festgestellt werden zu welchem Teil des Musters diese gehört. Nur so kann die Lage der dazugehörigen Ebene ermittelt und dann mittels Triangulation der 3D-Punkt errechnet werden. Dazu stehen verschiedene Möglichkeiten zur Verfügung:

- Zeitlich codierte Verfahren
- Phasencodierte Verfahren
- Frequenzcodierte Verfahren
- Örtlich codierte Verfahren



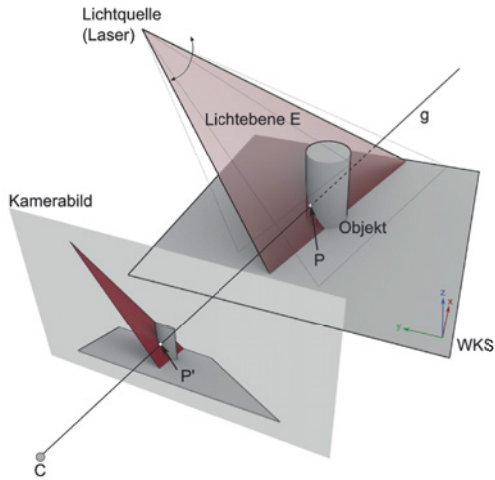
**Abb. 3.6.** Mögliche Abfolge eines zeitlich codierten Streifenmusters.

Zeitlich codierte Verfahren zeichnen sich dadurch aus, dass hier hintereinander verschiedene Streifenmuster projiziert werden, deren Abfolge bekannt ist und woraus sich dann für jeden Bildpunkt ergibt, an welcher Position im finalen Muster er sich befindet. Abbildung 3.6 zeigt eine einfache Codierung für solch ein Muster bei dem der Bildbereich jeweils wiederholt in Schwarz und Weiß aufgeteilt wird. Um gegenüber Erkennungsfehlern robuster zu sein kann dabei an Stelle des einfachen Binärcodes auch ein Graycode gewählt werden.

Bei phasencodierten Verfahren wird ein sinusförmiger Graustufenverlauf als Projektionsmuster verwendet. Dieses wird dann mit einer vorgegebenen Phasenverschiebung im Bereich von  $-\pi$  bis  $\pi$  ebenfalls mehrfach projiziert und die Zugehörigkeit eines Pixels lässt sich dann über die jeweilige Intensitätsabfolge rekonstruieren.

Will man lediglich mit einer einzigen Projektion auskommen (sog. One-Shot-Verfahren), so bietet sich zum Beispiel die Frequenzcodierung des Projektionsmusters an. Hierbei besteht das Muster aus Streifen unterschiedlicher Farben, deren Position im Muster bekannt ist. Über die von der Aufnahme gelieferte Farbe kann dann der passende Streifen gefunden werden. Probleme ergeben sich hier u.U. bei stark farblich kontrastierten Objekten bzw. bei der möglichst günstigen Anordnung und Auswahl der Farben für bestmögliche Unterscheidbarkeit.

Schließlich kann das Muster auch örtlich codiert werden. Dies kann derart realisiert werden, dass neben den eigentlichen Streifen z.B. farbige Kästchen im Muster vorhanden sind, deren Nachbarschaft möglichst eindeutig ist. Durch Betrachtung der Kästchen links und rechts eines Streifens kann so auf die Position des Streifens im Muster geschlossen werden. Auch hierbei ergeben sich allerdings Schwierigkeiten bei farbigen Objekten.



**Abb. 3.7.** Prinzip der 3D-Rekonstruktion mittels Laserlinienprojektion.

Da in dieser Arbeit ein Konica Minolta VI-900 Scanner eingesetzt wird, der auf Basis der Projektion einer Laserlinie arbeitet, wird im Folgenden nur auf dieses Verfahren näher eingegangen werden. Abbildung 3.7 zeigt den schematischen Aufbau eines 3D-Scanners auf Basis der Laserlinienprojektion. Er besteht aus einer Lichtquelle (hier ein aufgefächerter Laser) und einem Bildaufnehmer. Üblicherweise ist die Anordnung von Lichtquelle und Bildaufnehmer parallel, um die Abbildung anschaulicher zu gestalten wurde hier jedoch eine um  $90^\circ$  versetzte Anordnung gewählt. Der Laser spannt nun eine Ebene  $E$  auf, die auf das zu scannende Objekt trifft:

$$E : \mathbf{n}_0 \cdot \mathbf{x} - d = 0$$

Der Bildaufnehmer erfasst den Schnitt dieser Lichtebene mit dem Objekt. Mit Hilfe von Bildverarbeitungsmethoden wird nun die Position der Laserlinie auf dem Objekt im Bild ermittelt. Für einen Bildpunkt auf dieser Linie ( $P'$ ) kann nun mit Hilfe der Abbildungseigenschaften der Kamera die zugehörige Gerade gefunden werden:

$$g : \mathbf{x} = \mathbf{a} + r\mathbf{u}$$

Aus dem Schnitt dieser Geraden und der Lichtebene im Weltkoordinatensystem (WKS) kann der 3D-Punkt ( $P$ ) rekonstruiert werden:

$$\begin{aligned} \mathbf{n}_0(\mathbf{a} + r\mathbf{u}) - d &= 0 \\ \mathbf{a}\mathbf{n}_0 + r\mathbf{n}_0\mathbf{u} &= d \\ r &= \frac{d - \mathbf{a}\mathbf{n}_0}{\mathbf{n}_0\mathbf{u}} \\ \implies \mathbf{P} &= \mathbf{a} + \frac{d - \mathbf{a}\mathbf{n}_0}{\mathbf{n}_0\mathbf{u}} \cdot \mathbf{u} \end{aligned}$$

Auf diese Art lässt sich jedoch nur der von der Laserlinie getroffene Bereich rekonstruieren. Für eine vollständige Rekonstruktion kann beispielsweise die Laserebene geschwenkt werden oder das Objekt entsprechend bewegt werden. Bei der Bewegung des Objekts muss die Verschiebung ebenfalls bekannt sein und kann etwa durch einen Linearmotor realisiert werden. Die Verschiebung des Objekts bzw. die Lageänderung der Laserebene erfordern natürlich Zeit, was bedeutet, dass dieses Verfahren lediglich für statische Szenen gut geeignet ist.



### **Projektion von Rauschmustern**

Wie bereits im vorherigen Abschnitt angesprochen, kann neben der Liniensprojektion auch ein komplexeres Muster projiziert werden, um mit einer einzigen Aufnahme für das gesamte Kamerabild Tiefendaten zu rekonstruieren. Für die Szenenmodellierung, die in Kapitel 6 beschrieben wird, wird ein Kinect-Sensor von Microsoft verwendet, der auf diesem Prinzip beruht. Dessen Funktionsweise wird in diesem Abschnitt detailliert erläutert.

Der Microsoft Kinect Sensor enthält tatsächlich mehrere Sensoren und eine Schwenk-Neige-Einheit. Neben einem Mikrofon-Array, sind vor allem die beiden bildgebenden Sensoren (Farbe und Infrarot) sowie der laserbasierte Musterprojektor von Interesse. Der Messkopf bestehend aus den beiden Kamerasensoren und dem Projektor wurde von der Firma PrimeSense entwickelt und wird auch als eigenständiges Produkt (PS1080 SoC, [PrimeSense, Ltd. 12]) vertrieben. Die folgende Beschreibung der Funktionsweise basiert auf der zugehörigen Patentschrift [ZALEVSKY 07]. Das darin erwähnte Kalibrierverfahren für den Sensor ist ähnlich zur Patentschrift von Gockel und Azad [Gockel 06a].

Grundlage für die Akquisition der Tiefendaten ist die Projektion eines zufälligen Rauschmusters in die Szene. Dies kann beispielsweise durch einen Infrarotlaser mit vorgelagertem Diffusor oder durch eine mittels Holografie aufgenommene Reliefstruktur, die in den Laserstrahl eingebracht wird, realisiert werden. Das Rauschmuster entsteht dann durch Interferenzphänomene, wobei helle Punkte im Muster durch konstruktive Interferenz hervorgerufen werden. Diese Projektion wird nun durch einen geeigneten Sensor (z.B. CMOS-Chip) aufgenommen und somit in ein 2D-Bild

umgewandelt. Durch die Projektion auf die möglicherweise geschwungene oder schräge Objektoberfläche wird das Rauschmuster verformt und unterscheidet sich im aufgenommenen Bild vom ursprünglichen Muster. Der für die Rekonstruktion eingesetzte Matching-Algorithmus muss also mit einer variierenden Merkmalsgröße, zumindest in bestimmten Grenzen, umgehen können. Die Größe der einzelnen Merkmale im projizierten Rauschmuster hängt dabei von der Fokussierung und der Rayleighlänge<sup>1</sup> des eingesetzten Lasers ab. Die durchschnittliche Größe eines Merkmals im Kamerabild lässt sich berechnen zu:

$$\Delta x_{cam} = \frac{F}{\Phi_D} \cdot \lambda$$

wobei  $F$  die Brennweite des eingesetzten Objektivs,  $\lambda$  die Wellenlänge des Laserlichts und  $\Phi_D$  die Größe des Laserlichtpunkts auf dem Diffusor bezeichnet. Abbildung 3.8 zeigt das Infrarotrauschmuster wie es der Microsoft Kinect-Sensor projiziert. Um nun die 3D-Rekonstruktion durchzuführen wird ein Referenzbild des projizierten Rauschmusters vorgehalten. Bei einem Scan wird nun das durch ein Objekt veränderte Rauschmuster mit einer Kamera aufgenommen. Dieses Bild wird nun mit dem Referenzbild verglichen um entsprechende Korrespondenzen zu finden, z.B. durch eine ausschnittsweise Korrelationsberechnung. Durch ungünstige Reflektionseigenschaften eines Objektes können sekundäre Rauschmuster entstehen, die es zu minimieren gilt. Dies kann realisiert werden, indem die Größe der Linse der Kamera wesentlich größer gewählt wird als der Radius des Laserlichtpunkts auf dem Diffusor.

---

<sup>1</sup> Distanz entlang der optischen Achse eines Laserstrahls, aber der sich die Querschnittsfläche im Vergleich zum Fokuspunkt verdoppelt hat.



**Abb. 3.8.** Infrarot-Rauschmuster des Kinectsensors. Quelle: [Wikipedia, E. - User:Kolossos 12].

Der Korrelationsalgorithmus, der in der Patentschrift vorgeschlagen wird, beruht nun auf der Kontinuitätsannahme. Diese geht davon aus, dass Punkte, die in dem Bild eines Objektes nahe beieinander liegen sich in ihrem Tiefenwert nur minimal unterscheiden, also auf der selben Oberfläche liegen. Dies wird genutzt, in dem eine Prädiktion des Tiefenwerts für einen Nachbarpunkt ausgehend von einem bekannten Punkt durchgeführt werden kann. Ausgehend von den hellsten Punkten des Musters wird dann mit Hilfe eines Region-Growing-Verfahrens für möglichst alle Bildpunkte ein Tiefenwert bestimmt. Die eigentlich Tiefenberechnung erfolgt dann analog zu den bisher beschriebenen Verfahren.

### **3.3 Zusammenfassung**

In den vorangegangenen Abschnitten wurden die grundlegenden Prinzipien erläutert auf denen die im weiteren Verlauf der Arbeit verwendeten Sensoren aufbauen. Wesentlicher Bestandteil all dieser Sensoren sind Kameras, die aus der heutigen Servicerobotik nicht mehr weg zu denken sind. Wie bei jedem Sensor ist auch hier die Güte der Kalibrierung maßgeblich für die Qualität der erhaltenen Daten. Werden mehrere Kameras oder eine Kamera und eine Lichtquelle kombiniert so kann mit den vorgestellten Verfahren neben dem zweidimensionalen Intensitätsbild auch ein Tiefenbild rekonstruiert werden. Die einzelnen Techniken dazu unterscheiden sich in Genauigkeit, Schnelligkeit und Anwendbarkeit, basieren jedoch alle auf der Korrespondenzfindung und anschließender Triangulation.

## Konzept

Das folgende Kapitel gibt einen kurzen Überblick über das in dieser Arbeit vorgeschlagene und entwickelte Modellierkonzept und setzt die einzelnen Funktionskomponenten zueinander in Beziehung. Dies ermöglicht die Einordnung der Themen der folgenden Kapitel in die Lösung übergeordnete Fragestellung und damit auch der entwickelten Systemarchitektur.

### 4.1 Motivation

Serviceobotersysteme benötigen in hohem Maße Modellwissen zur Bewältigung der ihnen gestellten Aufgaben und zur Anpassung ihres Verhaltens an Veränderungen in ihrer Umgebung. Wie in Kapitel 2 angedeutet, ist die Erzeugung dieses Modellwissens jedoch nicht homogen und in der Regel nicht automatisiert. Dies führt zu sehr spezifischen Teilmodellen mit lokaler Sicht bzw. Gültigkeit. Diese Arbeit hat zum Ziel dieser Fragmentierung durch überwiegend lokalen Sichten und Repräsentationen zu

begegnen, indem ein Verfahren vorgeschlagen wird, welches die für Serviceroboter erforderlichen Modellierungsaufgaben weitgehend einheitlich und automatisiert durchführen kann.

## 4.2 Überblick

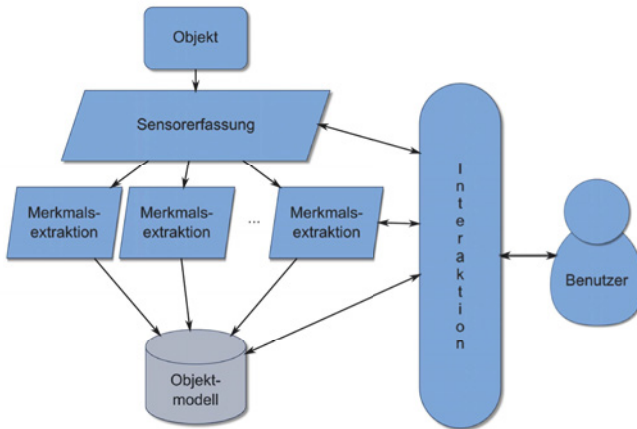
Zur Unterstützung der Autonomie von Servicerobotern fallen viele unterschiedliche Modellierungsaufgaben an, wie bspw. die kinematische Modellierung des Roboters selbst oder die Modellierung von Dialogen mit menschlichen Nutzern. Die Behandlung aller Modellierungsaufgaben in der Robotik übersteigt den Rahmen dieser Dissertation, weshalb der Fokus auf die „externen“ Modelle gelegt wird, also die Modellierung der Umwelt und insbesondere der dort zu berücksichtigenden Objekte und Gegenstände. Dieser Aufgabenbereich lässt sich dann weiter in die Bereiche *Objektmodellierung* und *Szenenmodellierung* unterteilen.

### 4.2.1 Objektmodelle

Zu den Aufgaben der Objektmodellierung gehört die Erzeugung von Daten und Repräsentationen fokussiert auf einzelne Objekte oder Gegenstände mit denen ein Robotersystem interagieren soll. Zur Lösung dieser Aufgabe sei folgende These aufgestellt:

**These 1** *Die im Bereich der Servicerobotik angewendeten Methoden und Algorithmen, die sich mit Fragen der Perzeption und Manipulation befassen, basieren in der Regel auf ähnlichen Objektmodellen. Diese Modelle können in einem separaten Schritt einheitlich und automatisiert gewonnen werden. Eine möglichst hohe Qualität der Modelle kann durch die Interaktion eines menschlichen Benutzers während des Modellierungsprozesses erzielt werden.*

Die These basiert auf den in Abschnitt 2.1.1 gewonnenen Einsichten und Schlüsse zum Stand der Forschung in den Bereichen Perzeption und Manipulation von Alltagsgegenständen. Die Modellierung eines Objektes er-



**Abb. 4.1.** Schematische Übersicht über das vorgeschlagene Konzept zur Objektmodellierung.

folgt so, dass zunächst eine sensorielle Erfassung des zu modellierenden Objektes stattfindet. Die sensorisch erfassten Objektdaten werden dann weiter verarbeitet und aus den daraus abgeleiteten Objektmerkmalen entsteht dann schrittweise das Objektmodell. Der gesamte Prozess wird durch die Interaktion eines menschlichen Benutzers unterstützt. Abbildung 4.1 zeigt eine schematische Übersicht über diesen Prozess. Ziel dieses Prozesses ist es, ein gegebenes Objekt derart zu digitalisieren, dass die entstehende Objektrepräsentation für möglichst viele Anwendungen nutzbar ist. Kapitel 5 beschreibt einen Vorschlag zur Implementierung dieses Modellierungsprozesses im Detail.

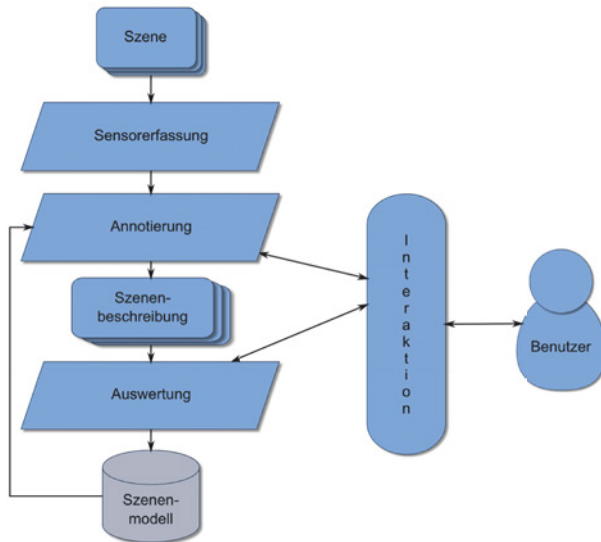
#### 4.2.2 Szenenmodelle

Ziel der Szenenmodellierung ist die Einordnung der einzelnen Objekte in einen kontextuellen Gesamtzusammenhang wie er beispielsweise in einer Alltagsumgebung vorhanden ist. Ausgangspunkt für die Modellierung von Alltagssituationen und -kontexten ist folgende These:

**These 2** *Gegenstände in einer Alltagsumgebung sind in der Regel nicht willkürlich im Raum verteilt, sondern die Anordnung resultiert aus funktionalen und kontextuellen Erfordernissen, welche darin kodiert sind. Die Beobachtung und Auswertung einer Vielzahl solcher Anordnungen ermöglicht die Erzeugung von Hintergrundwissen über diese Kodierung und deren Anwendung auf neu beobachtete Szenen.*



Diese These basiert auf den in Abschnitt 2.2 diskutierten Methoden zur Szenenklassifikation und dem Schluss, dass Kontextwissen die kognitiven Fähigkeiten eines Robotersystems deutlich verbessern kann. Die Model-



**Abb. 4.2.** Schematische Übersicht über das vorgeschlagene Konzept zur Szenenmodellierung.

lierung von Szenen dient also der Extraktion von Hintergrundwissen über räumliche Anordnungen von Gegenständen, die wiederum zu Aussagen über die Art der Gegenstände selbst zulässt. Der Modellierungsprozess gestaltet sich also - wie in Abb. 4.2 dargestellt - derart, dass eine große Zahl ähnlicher Alltagsszenen beobachtet und ausgewertet wird, was dann zu einem probabilistisch begründeten Szenenmodell führt. Auch wird durch

Interaktion das Objekt- und Szenenwissen des Menschen genutzt, um die Qualität der Modelle zu verbessern und die Abstraktion der Sensordaten auf Konzepte an die menschliche Wahrnehmung anzugleichen. Ziel dieses Prozesses ist es, zu einer gegebenen Szenenkatgorie, für die darin vorkommenden Objekte und deren räumliche Relationen ein probabilistisches Modell zu erzeugen. Ein solches durch vorangegangene Auswertungen gewonnenes Modell, kann dann in zukünftige Auswertungen zurückgeführt werden. Kapitel 6 beschreibt die Implementierung dieses Modellierungsprozesses im Detail.

### **4.3 Zusammenfassung**

Zur Objekt- und Szenenmodellierung wird ein vereinheitlichter Prozess zur Gewinnung von Objektmodellen im Kontext der Perzeption und Manipulation von Servicerobotern vorgeschlagen. Die Objekte werden durch ein probabilistisches Szenenmodell in Zusammenhang gebracht, wodurch die Objektmodelle selbst erweitert werden. Insgesamt kann damit eine umfassende Wissensbasis für Robotersysteme geschaffen werden, was sowohl deren Fähigkeiten erweitert, wie auch der Entwicklung von Algorithmen in diesem Umfeld zugute kommt.

## Sensorgestützte Modellierung von Einzelobjekten

### 5.1 Einleitung

In dem Hollywoodfilm „Indiana Jones und der letzte Kreuzzug“ geht es um die Suche nach dem heiligen Gral, dem Gefäß welches der Legende nach beim letzten Abendmahl die Runde durch die Jünger machte und von Jesus Christus selbst gereicht wurde. Nach langer und gefährvoller Suche erreichen der Held und sein Gegenspieler die letzte Kammer des Tempels in dem sich der Gral befinden soll. Doch entgegen dessen, was wohl die meisten erwarten würden, findet sich dort nicht der eine Becher aufgebahrt auf einer Empore, sondern eine Vielzahl unterschiedlichster Becher und Kelche, wovon jedoch nur einer der wahre Gral ist. Es gilt also zu wählen, doch welcher ist der Richtige? Keiner der beteiligten weiß wie der Gral aussieht, niemand hat eine Vorstellung davon. Der Filmbösewicht wählt schließlich einen mit Edelsteinen besetzten Goldpokal, weil er davon ausgeht, dass dieses kostbare Gefäß entsprechend aufwändig und wertvoll beschaffen ist. Seine Wahl ist jedoch falsch und der eigentliche

Gral, zumindest im Film, ist ein schlichter Becher aus wenig wertvollem Material.

Diese Filmszene enthält viel interessante Information über die Eigenschaften und Eigenarten von Modellen - nichts anderes ist schließlich die Vorstellung der Protagonisten vom heiligen Gral. Zum einen zeigt sich wie mächtig ein Modell sein kann, auch wenn es nur ungenau beschrieben ist, denn allen Beteiligten ist völlig klar, dass es sich um ein Gefäß handeln muss, also ein Objekt welches Flüssigkeit in sich aufnehmen kann und dessen Form und Funktion dadurch maßgeblich bestimmt ist. Keiner wäre auf die Idee gekommen den Tisch, auf dem sich alle Kelche befanden, in die Wahlmöglichkeit mit einzubeziehen. Es gibt also deutliche Überschneidungen zwischen den jeweiligen Vorstellungen, die sich auf ganz konkrete Merkmale des Objekts zurückführen lassen. Andererseits zeigt sich dass das Modell, zumindest auf Seiten des Antagonisten, nicht ausreichend formuliert war, denn er wählt den falschen Kelch und bezahlt dies mit seinem Leben. Der Held hingegen vervollständigt sein Modell mit weiterer Information („Der Kelch eines Zimmermanns.“) und wählt korrekt. Dies verdeutlicht, dass unzureichende Spezialisierung in bestimmten Fällen drastische Konsequenzen haben kann und damit also der Grad der Abstraktion des Modells je nach Anwendung richtig gewählt werden muss.

Bei Servicerobotern führen nicht nur die unzureichende Spezialisierung der benötigten Modelle zu Schwierigkeiten, es gibt noch weitere Aspekte zu beachten. Bei der Verwendung von Objektmodellen spielt die Repräsentation eine wichtige Rolle. Das Modell muss in einer Form vorliegen, die vom jeweiligen System verstanden und interpretiert werden kann und die ein der vorliegenden Aufgabe entsprechendes Abstraktionsniveau auf-

weist. So kann ein 3D-Modell für eine akustische Objekterkennung in der Regel wenig Information liefern oder die Beschreibung „blauer Becher, groß“ im Kontext einer Greifplanung kein effektiver Hinweis sein. Ein weiterer wichtiger Aspekt bei der Nutzung von Objektmodellen in der Robotik ist auch die Frage wie die Modelle generiert wurden. Denn ohne die Möglichkeit Objekte effizient zu modellieren, ist für die Umsetzung komplexer Aufgabenstellungen und hohe Autonomie nicht genügend Modellwissen vorhanden. Das folgende Kapitel widmet sich den Fragestellungen wie Objektmodelle für Anwendungen in der Servicerobotik aussehen können und vor allem wie solche Modelle effizient und präzise erzeugt werden können.

## 5.2 Problemstellung

Wie bereits angesprochen, sind Modelle von Objekten für die Servicerobotik besonders relevant. Bei der visuellen Objekterkennung und der Greifplanung setzen die meisten Algorithmen und Methoden ein Modell des jeweiligen Objekts als Eingabe voraus. Methoden, die ohne Vorwissen, also modellfrei arbeiten können, erfordern eine Explorationsstrategie oder sammeln Erfahrungswissen im Sinne generativer Prozesse. Derzeit verfügbare Ansätze können nur Objekte behandeln, die bereits a-priori bekannt und modelliert sind.

Obwohl Objektmodelle zentraler Bestandteil vieler Arbeiten sind, bleibt die Entwicklung von Methoden zur Erzeugung der Modelle weitgehend unbeachtet. Dies begründet sich vermutlich in praktischen Überlegungen: Für die konzeptuelle Entwicklung von Algorithmen zur Objekterkennung

oder Greifplanung sind relativ geringe Mengen an Modellen erforderlich, entscheidend ist hier vielmehr die Varianz der Modelle. Soll ein Robotersystem von seiner Umgebung auf Alltagstauglichkeit hin entwickelt werden, müssen Objekterkennungs- und Planungssysteme in der Lage sein eine Vielzahl von Objekten zu behandeln. Mit modellbasierten Ansätzen kommt man also nicht umhin eine umfangreiche Menge von Modellen zu generieren. Auch bei der Evaluierung der Skalierungseigenschaften von Algorithmen spielen große Datensätze eine wichtige Rolle.

Es besteht also ein hoher Bedarf an Methoden, entsprechende Datensätze effizient zu erzeugen. In dieser Arbeit werden entsprechende Methoden vorgeschlagen und entwickelt um einen hinreichenden Beitrag zur Unterstützung des Bedarfs an Modelldatensätzen für eine Reihe von Haushaltssituationen zu leisten.

### **5.3 Lösungsansatz**

Um die erzeugten Modelle und die dafür nötigen Methoden möglichst effizient an die Anforderungen in der Servicerobotik anzupassen, wurden Ansätze auf den Gebieten der Objekterkennung, der Greifplanung und der Objektmanipulation im Kontext der Servicerobotik untersucht und daraus die benötigten Informationen über die Objekte gewonnen. Es zeigt sich, dass die Gestalt eines Objekts, also seine dreidimensionale Repräsentation sowie verschiedenartige Ansichten für die meisten Algorithmen eine geeignete Dateneingabe darstellen.

Zur Erzeugung geeigneter Datensätze und Modell wird im Folgenden die Konstruktion einer spezialisierten Sensoranordnung vorgestellt, die die

Datenaufnahme unterstützen soll. Die dafür vorgesehenen Sensoren ermöglichen die Aufnahme von Objektansichten, 3D-Informationen und erlauben dabei einen hohen Grad an Automatisierung und Präzision.

Im Anschluss an die reine Datenaufnahme ist eine Nachbearbeitung der Rohdaten vorgesehen, um aus diesen geeignete Modelle zu erzeugen und hohe Genauigkeit zu gewährleisten. Dazu wird eine spezielle Software vorgeschlagen, um den manuellen Aufwand gering zu halten.

Nach Erzeugung der Modelldaten sollen diese einem breiten Anwenderkreis zur Verfügung gestellt werden. Dafür wird eine Verbreitungsplattform benötigt. Im vorliegenden Fall existieren davon zwei Ausprägungen, eine web-basierte Variante, die den kompletten Datensatz zur Verfügung stellt und auf die Entwicklung und Evaluierung von Algorithmen ausgerichtet ist und eine Datenbank, die die Objektmodelle direkt zur Laufzeit des Robotersystems zur Verfügung stellt und somit als eine Art Hintergrundwissen für das System fungiert.

In den folgenden Abschnitten werden die einzelnen Komponenten dieses Lösungsansatzes detailliert vorgestellt.

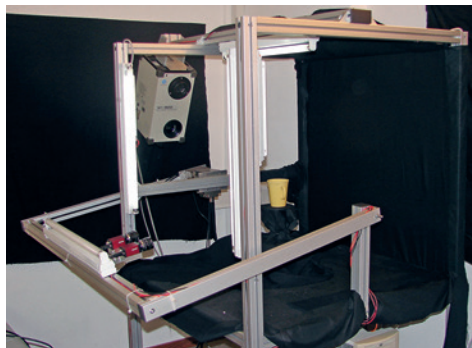
## 5.4 Sensoraufbau

Bei der Konzeption und Konstruktion des Sensoraufbaus, im Folgenden auch *Modellierungscenter* genannt, standen folgende Anforderungen und Kriterien im Vordergrund:

- Eine automatische 3D-Datenerfassung

- Die Erzeugung von Stereobildansichten aus unterschiedlichen Perspektiven, ähnlich einem Robotersystem
- Eine Regulierbare Beleuchtung
- Der Bezug zwischen Kameradaten und 3D-Daten
- Ein hoher Grad an Automatisierung des Aufnahmeprozesses

Zu Beginn dieser Arbeit existierte bereits ein erster Prototyp einer 3D-Sensorkonfiguration, die bereits die notwendige Sensorik und Aktorik enthielt, vgl. [Becher 08] und Abb. 5.1. Allerdings wies dieser Aufbau vor allem im Bereich der mechanischen Positionierung des Stereokamerasystems erhebliche Schwächen auf. Das Kamerasystem wurde an einem weit



**Abb. 5.1.** Erste Version des Sensoraufbaus zur Objektaufnahme. Hier wurde das Stereokamerasystem noch an einem Schwenkarm geführt.

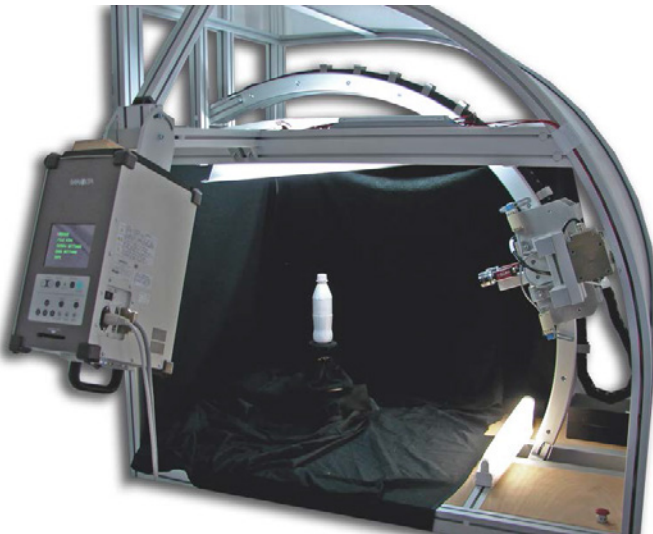
ausladenden Arm befestigt, der an zwei Gelenken rotatorisch aufgehängt war. Auf der einen Seite wurde der Arm dann von einem Motor bewegt um die Kameras in einem Kreisbogen um das Objekt zu führen. Durch



den einseitigen Antrieb und die ausladende Bauweise neigte dieser Aufbau zu Schwingungen und Torsionen, die eine exakte Positionierung der Kameras unmöglich machten.

Die Erfahrungen aus diesem ersten Prototypen beeinflussten die Konstruktion des im Rahmen dieser Arbeit entwickelten Aufbaus insofern, dass die Mechanik zur Positionierung der Kameras grundlegend verändert wurde.

#### 5.4.1 Mechanischer Aufbau



**Abb. 5.2.** Weiterentwickelte Version des Sensoraufbaus zur Objektaufnahme (Modellierungszentrum).

Das Sensoraufbau zur Digitalisierung beliebiger Objekte besteht aus einem 3D-Scanner, einem Rotationsteller, drei Beleuchtungseinheiten und einem Stereokamerasystem. Die spezifische Anordnung der sensorischen und mechanischen Elemente wurde im Rahmen der Arbeit entwickelt. Die Konstruktion und Fertigung des Gesamtaufbaus erfolgte in Zusammenarbeit mit der Firma Norcan<sup>1</sup>. Wie in Abbildung 5.2 zu sehen, ist das Stereokamerasystem des Modellierungscenters an einem Schlitten montiert, der entlang einer kreisförmig gebogenen Schiene verfahren werden kann. Das Zentrum dieses Kreises liegt dabei in der Mitte des Aufbaus, so dass die Kameras auf dem Schlitten ein im Zentrum positioniertes Objekt, sowohl waagrecht von der Seite als auch senkrecht von oben aufnehmen können. Der Abstand der Kameras zum Zentrum beträgt mit den eingesetzten Kameras  $\approx 62\text{ cm}$ .

Der Schlitten wird über einen Zahnriemen von einem auf dem Schlitten angebrachten Amtec Powercube PR 090 angetrieben. Das Antriebsverhältnis beträgt dabei  $\approx 1 : 28,83$ . Um also den Bereich von  $0^\circ$  (waagrecht) bis  $90^\circ$  (senkrecht) abzufahren, muss der Powercube eine Rotation von  $2595^\circ$  durchführen. Dieses hohe Übersetzungsverhältnis, zusammen mit der präzisen Positionsaufösung des PR 090 von 4 Winkelsekunden/Inkrement und 894 Inkrementen/Grad, erlaubt eine hochgenaue Positionierung der Kameras entlang der Schiene.

Im Zentrum des Modellierungscenters ist der Rotationsteller (Isel RF-1) angebracht, der es ermöglicht das darauf platzierte Objekt um die Längsachse zu rotieren, wobei auf diese Weise für die Positionierung der Kameras ein zweiter Freiheitsgrad erzeugt wird. Gleichzeitig wird da-

---

<sup>1</sup> <http://www.norcan.fr>

durch auch die automatische Aufnahme unterschiedlicher Objektansichten durch den 3D-Scanner ermöglicht.

Für den Rotationsteller wurden zwei verschiedene Aufsätze gefertigt (s. Abb. 5.3) um die Objekte im Zentrum positionieren zu können. Einer der beiden Aufsätze ist dabei in der Höhe verstellbar und ermöglicht so auch die Aufnahme von größeren Objekten, die mit dem starren Aufsatz z.B. aus dem Kamerabild herausragen würden.



**Abb. 5.3.** Aufsätze für den Rotationsteller. Links: feste Höhe, rechts: höhenverstellbar.

Um sowohl den Kameras, wie auch dem 3D-Sensor ein möglichst freies Blickfeld zu gewährleisten, ist der 3D-Sensor  $90^\circ$  zur Kreisbahn der Kameras versetzt angebracht und ebenfalls auf den Drehteller in der Mitte ausgerichtet. Der Sensor blickt leicht von schräg oben auf das Zentrum um für kleinere Objekte neben der Seitenansicht auch Teile der Aufsicht gleichzeitig erfassen zu können. Der Abstand des Sensors zum Zentrum beträgt  $\approx 100\text{ cm}$ .

Schließlich enthält das Modellierungscenter noch drei Lichtquellen, bestehend aus Lampen mit Leuchtstoffröhren. Die Lampen sind mit Diffusoren

ausgestattet und über elektronische Hochfrequenzsteuerungseinheiten in der Helligkeit regulierbar. Die Steuerungsgeräte sind über eine Schnittstellenkarte mit dem PC verbunden, was die automatische Helligkeitsregulierung per Software ermöglicht.

## 5.4.2 Sensorik

### 3D-Sensorsystem Konica Minolta VI-900



**Abb. 5.4.** 3D-Scanner Konica Minolta VI-900.

Zur Aufnahme der 3D-Daten wird ein Konica Minolta VI-900 eingesetzt. Der Sensor arbeitet mit einem Lichtschnittverfahren (s. Abschnitt 3.2.2)

und besteht dementsprechend aus einer Kamera und einem schwenkbaren Linienlaser. Der VI-900 kann einerseits offline betrieben werden, dazu kann der Scan über den eingebauten Bildschirm gesteuert werden und die Ergebnisse im internen Speicher oder auf einer Compact-Flash-Karte abgelegt werden. Weiterhin gibt es die Möglichkeit den Sensor ferngesteuert über die SCSI-Schnittstelle von einem Rechner aus zu betreiben. Dafür kann sowohl die vom Hersteller zur Verfügung gestellte Software verwendet werden, oder auf Basis des ebenfalls vom Hersteller erhältlichen SDK<sup>2</sup> eigene Applikationen erstellt werden. Letzteres wird in dieser Arbeit zur Ansteuerung des Scanners eingesetzt. Das Sensorsystem bietet eine maxi-

**Tabelle 5.1.** Spezifikationen des Konica Minolta VI-900 3D-Scanners

Spezifikationen gemäß Datenblatt

Measuring method	Triangulation light block method
AF	Image surface AF (contrast method), active AF
Image input range	0.6 to 2.5 m (with different lenses)
Measurement input range	0.6 to 1.2 m
Scan area	111 x 84 mm – 710 x 533 mm, max. 1300 x 1100 mm
Sample time	0.3 s to 2.5 s
Number of output pixels	640 x 480 (3D and color)
Geometrical resolution	x = 0.17 mm, y = 0.17 mm, z = 0.047 mm at 0.6 m

male Auflösung von 640x480 Bildpunkten, bei einer Tiefenauflösung von 0,047 mm. Der typische Objektstand beträgt 0,6 - 1,2 m. Die absolute

<sup>2</sup> Software Development Kit

Auflösung mit der Objekte aufgenommen werden können, hängt dabei von der eingesetzten Optik ab. Das System bietet drei verschiedene Objektive mit unterschiedlichen Brennweiten (Tele:  $f = 25\text{ mm}$ , Normal:  $f = 14\text{ mm}$ , Weitwinkel:  $f = 8\text{ mm}$ ), so dass bei Einsatz des Teleobjektivs auch kleine Gegenstände sehr dicht abgetastet werden können und der Einsatz des Weitwinkelobjektivs auch die vollständige Erfassung größerer Objekte in einem Abtastvorgang erlaubt. Genauere Spezifikationen des Systems finden sich in Tabelle 5.1.

Der VI-900 bietet zudem einige Automatismen, die die Erfassung der 3D-Daten vereinfachen. So ist beispielsweise ein aktiver wie ein passiver Autofokus verbaut, der das Objektiv automatisch auf die gemessene Objektdistanz einstellt. Weiterhin besteht die Möglichkeit, mit Hilfe eines mitgelieferten Kalibrierobjekts, die Rotationsachse eines eventuell vorhandenen Drehtellers zu ermitteln und so die Registrierung einzelner Scans zu vereinfachen. Eine genauere Erklärung dieses Mechanismus erfolgt in Abschnitt 5.5. Beim Zugriff auf die Sensordaten bietet das SDK verschiedene Möglichkeiten der Vorverarbeitung der Rohdaten, von der Ausgabe der reinen Tiefenwerte und des Kamerabildes bis zur Vorberechnung gefilterter, triangulierter Punktwolken. Nähere Ausführungen hierzu finden sich in Abschnitt 5.6.

### **Stereokamerasystem**

Für die Aufnahme der Objektansichten ist ein Stereokamerasystem, bestehend aus zwei AVT Marlin 145C2 Kameras mit IEEE1394-Schnittstelle, im Modellierungcenter verbaut. Die AVT Marlin 145C2 Kamera bietet eine Auflösung von  $1392 \times 1038$  Bildpunkten bei einer Frequenz von ma-



**Abb. 5.5.** Stereokamerasystem auf Schlitten im Modellierungcenter verbaut.

ximal 10 Bildern pro Sekunde. Weitere technische Details finden sich in Tabelle 5.2. Die Marlin-Kameras bieten eine Vielzahl von Einstellungs-

**Tabelle 5.2.** Daten der AVT Marlin 145C2 Kameras

Spezifikationen AVT Marlin 145C2 gemäß [Allied Vision 11]

Interface	IEEE1394a - 400MBit/s
Resolution	1392 x 1038 (YUV)
Sensor	SONY 1/2" progressive CCD
Framerate	up to 10 fps at full resolution

möglichkeiten; besonders interessant für die vorliegende Anwendung sind der manuelle Weißabgleich, der steuerbare Verschluss und die Verstärkung (gain). Die Ansteuerung der Kameras erfolgte über die IEEE1394-Schnittstelle mit Hilfe des offiziellen IEEE1394-Treibers der Firma AVT und der ebenfalls von AVT zur Verfügung gestellten Softwarebibliothek *FireGrab*.

Als Objektive für die Kameras wurden Modelle der Firmen Pentax und Schneider-Kreuznach verwendet. Details zu den Objektiven können Ta-

belle 5.3 entnommen werden. Abbildung 5.6 zeigt die beiden Objektivvarianten.

**Tabelle 5.3.** Spezifikationen der verwendeten Objektive

Spezifikationen gemäß Datenblatt

Modell	Pentax	Schneider	KMP-IR
		Cinegon	8/1,4 -
		M30,5	
Brennweite	8,5 mm	8 mm	
Öffnung	f1,5	f1,4	
Minimale Objekt- distanz	0,2 m	0,0 m	



**Abb. 5.6.** Im Stereokamerasystem verwendete Objektive (Schneider-Kreuznach links, Pentax rechts).

## 5.5 Kalibrierung

Mit Hilfe der verwendeten Sensorik können sowohl 3D- wie auch 2D-Daten erzeugt werden. Der mechanische Aufbau gibt zwar die grundlegende Konfiguration von Tiefen- und Bildsensorik vor, um die jeweiligen



Datensätze jedoch in eine räumliche Beziehung setzen zu können ist eine genauere Kalibrierung notwendig. Diese dient dazu, die Position des Kamerasystems innerhalb des durch den 3D-Sensor vorgegebenen Koordinatensystems, abhängig von der Stellung des Rotationstellers und des Schlittens auf der Kreisschiene, ermitteln zu können.

Die Kalibrierung muss folgende Gegebenheiten ermitteln:

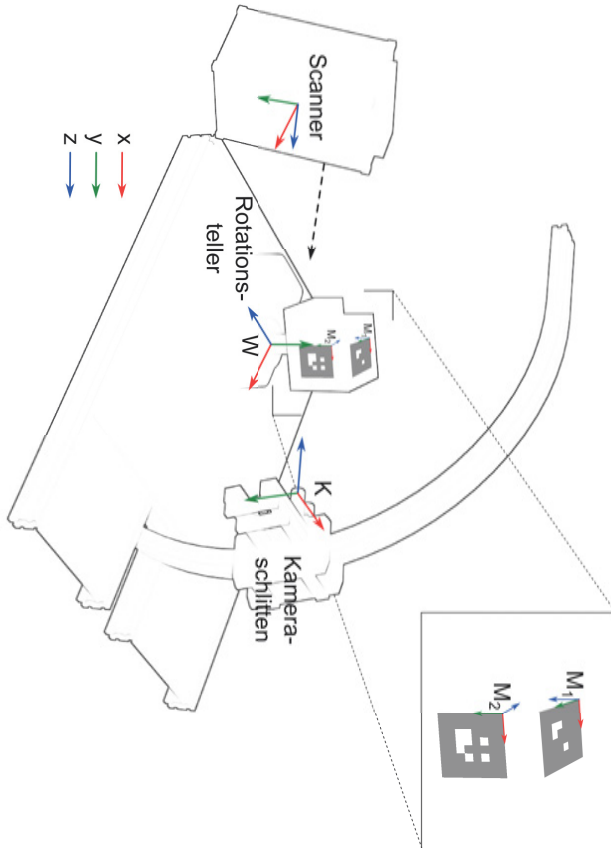
- Position und Orientierung der Drehachse des Rotationstellers in Scannerkoordinaten
- Position und Orientierung der Kameras zu gegebenen Stellungen des Kameraschlittens in Scannerkoordinaten

Die Kalibrierung verläuft insgesamt in mehreren Schritten:

1. Ermittlung der Drehteller-Rotationsachse
2. Ermittlung der Pose des Kalibrierobjekts in Weltkoordinaten
3. Ermittlung der Kameraposen relativ zum Kalibrierobjekt für vorgegebene Kamerastellungen
4. Optimierung der Kameraposen

### **Schritt 1: Ermittlung der Rotationsachse des Drehtellers**

Position und Orientierung der Rotationsachse des Drehtellers können mit Hilfe des Scanners direkt ermittelt werden. Dazu wird das in Abb. 5.8 dargestellte Kalibrierobjekt auf dem Rotationsteller positioniert und eingeschannt. Die beiden schräg zueinander stehenden Ebenen des Kalibrierobjekts schneiden sich in der Rotationsachse. Die Scannersoftware ermittelt



**Abb. 5.7.** Lage der Koordinatensysteme im Modellierungszentrum.  $W$  bezeichnet das Weltkoordinatensystem,  $K$  das Kamerakordinatensystem und  $M_{1,2}$  die Markerkoordinatensysteme.

diese Schnittgerade automatisch und legt das Scannerkoordinatensystem (im Folgenden das Weltkoordinatensystem) so, dass die y-Achse entlang der Rotationsachse verläuft. Die z-Achse zeigt dabei in Blickrichtung des Scanners, der Ursprung des Koordinatensystems liegt abhängig von der Position des Kalibrierobjektes.



**Abb. 5.8.** Kalibrierobjekt um die Rotationsachse des Rotationstellers zu bestimmen.

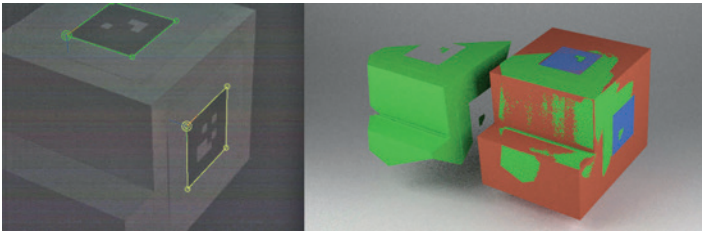
## **Schritt 2: Lokalisation des Kalibrierobjektes im Tiefenbild**

Für die Ermittlung der Kameraposen wurde ein spezielles Kalibrierobjekt erstellt, welches sowohl im Bild des Tiefensensors, wie auch in den Kamerabildern detektierbar ist. Das in Abb. 5.9 dargestellte, quaderförmige Kalibrierobjekt ist nicht rotationssymmetrisch, um die Detektion im Tiefenbild eindeutig durchführen zu können. Zusätzlich sind auf dem Kalibrierobjekt zwei spezielle Marker angebracht, die die Rekonstruktion der Kameraposen relativ zu diesen Markern über Bildverarbeitung ermöglichen. Das Kalibrierobjekt ist genau vermessen und die Positionen der



**Abb. 5.9.** Kalibrierobjekt zur Bestimmung der Kameraposen.

Marker auf dem Objekt sind ebenfalls bekannt. Somit lässt sich ein Referenzmodell erstellen, welches den Zusammenhang zwischen der Pose des Kalibrierobjekts in Weltkoordinaten mit den Positionen der Marker beschreibt. Um die Pose des Kalibrierobjektes im Weltkoordinatensystem



**Abb. 5.10.** Lokalisierung des Kalibrierobjekts im 3D-Scan. Rot: Referenzmodell des Kalibrierobjekts. Blau: AR-Marker im Referenzmodell. Grün: 3D-Scan des Kalibrierobjekts. Links im Bild die Ansicht des Scanners mit den nach der Registrierung gefundenen Markerpositionen. Das Referenzmodell wurde hier bereits mit dem Scan per ICP registriert (zweiter Scan in grün lediglich zu Visualisierungszwecken).

zu ermitteln wird das Objekt auf dem Rotationsteller positioniert und mit dem 3D-Sensor erfasst. Das so gewonnene Dreiecksnetz eines Teils des

Kalibrierobjekts wird nun zunächst von Rauschen und Ausreißern befreit. Anschließend wird das Referenzmodell, ebenfalls ein Dreiecksnetz, welches neben der Gestalt des Kalibrierobjekts auch die Posen der beiden Marker enthält, in die reale Aufnahme eingepasst. Dies geschieht mit Hilfe des Verfahrens *Iterative Closest Point (ICP)*. Ein beispielhaftes Ergebnis dieses Prozesses ist in Abb. 5.10 dargestellt. In der Verwendung dieses Verfahrens begründet sich auch die nicht rotationssymmetrische Form des Kalibrierobjekts. Dies verringert die Wahrscheinlichkeit, dass das iterative Registrierungsverfahren ein lokales Minimum als Lösung ausgibt. Um zu verifizieren, dass eine gute Lösung durch das ICP-Verfahren gefunden wurde, wird anschließend ein Gütemaß für die Registrierung berechnet. Dazu wird der kürzeste Abstand aller 3D-Punkte des Tiefenbildes zum registrierten Referenzobjekt berechnet und darüber die Quadratsumme gebildet. Für eine genaue Sensoraufnahme und eine möglichst gute Registrierung sollte dieser Wert minimal sein. Ist die berechnete Summe kleiner als ein empirisch bestimmter Schwellwert, wird die Lokalisierung des Kalibrierobjekts als erfolgreich angesehen. Die durch das ICP-Verfahren ermittelte Transformation des Referenzmodells kann nun auf die Posen der Marker angewendet werden, wodurch deren Posen im Weltkoordinatensystem bekannt sind.

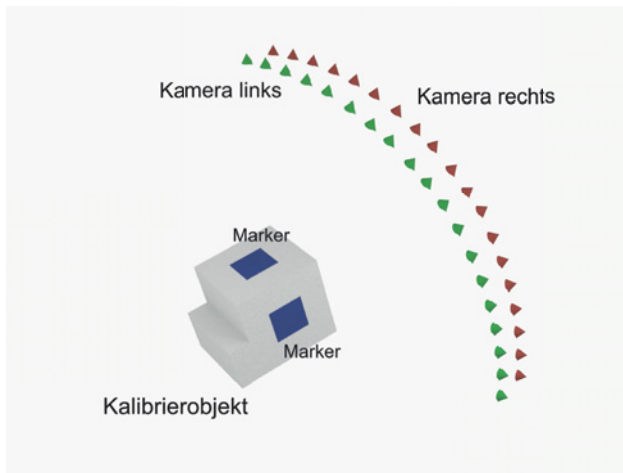
### **Schritt 3: Ermittlung der Kameraposen relativ zu den Markern**

Um die Pose der Kamera im Weltkoordinatensystem zu bestimmen, wird eine sog. Markererkennung verwendet. Dieses Verfahren ermöglicht die Bestimmung einer 6D-Pose bezüglich spezieller Muster, den sogenannten Markern. Auf dem in Abb. 5.11 dargestellten Kalibrierobjekt befinden



**Abb. 5.11.** Die auf dem Kalibrierobjekt angebrachten Marker. Links: ID=2, rechts: ID=10.

sich zwei dieser Marker. Die Markermuster sind binär kodierte Pixelmuster, die in einem Kamerabild eindeutig identifiziert werden können und deren Position und Orientierung rekonstruiert werden kann. Sind die Abmessungen des Markers bekannt, kann damit auch die Pose der Kamera relativ zum Marker bzw. umgekehrt berechnet werden. In der vorliegenden Arbeit wurde für diese Lokalisierung die Implementierung der Firma Keytech ([Azad 12]) verwendet, die diese freundlicherweise zu Testzwecken zur Verfügung gestellt hat. Diese Implementierung liefert neben der Markerpose auch ein Gütekriterium in Form des Rückprojektionsfehlers. Dieser wird berechnet in dem mit Hilfe der ermittelten Pose und den intrinsischen Parametern der Kamera, der entsprechende Marker in das zugrunde liegende Bild projiziert wird und die Abweichung der realen Markerpunkte zu den künstlich projizierten Markerpunkten ermittelt wird. Da im vorliegenden Aufbau beide Marker gleichzeitig im Bild sichtbar sein können, wird auf Basis dieses Gütekriteriums jeweils der günstigere Marker mit dem kleineren Rückprojektionsfehler für die Bestimmung der Kamerapose ausgewählt. Für die Initialisierung der Kalibrierung wird auf die-

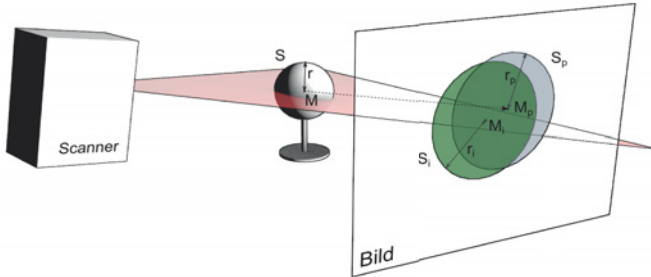


**Abb. 5.12.** Mit Hilfe der AR-Marker ermittelte Kameraposen (grün: linke Kamera, rot: rechte Kamera).

se Art nun für vorgegebene Stellungen der Kameras entlang der Schiene (z.B. alle  $5^\circ$ ) die Pose beider Kameras ermittelt. Zusammen mit den im vorherigen Schritt bestimmten Posen der Marker kann dann die jeweilige Kamerapose in Weltkoordinaten berechnet werden. Ein beispielhaftes Ergebnis dieses Schritts ist in Abb. 5.12 dargestellt. Diese Posen dienen dann als Eingabe für die im nächsten Schritt erfolgende Optimierung.

#### **Schritt 4: Optimierung der Kameraposen**

Um die in Schritt 3 ermittelten initialen Kameraposen zu optimieren wird eine Styroporkugel verwendet. Diese wird zunächst genau wie das Kalibrierobjekt mit dem Tiefensensor aufgezeichnet und dann ein Referenzmodell der Kugel in den Scan eingepasst, wieder mit Hilfe des ICP-



**Abb. 5.13.** Optimierung der extrinsischen Kamerakalibrierung mit Hilfe einer Kugel.

Verfahrens. Anschließend wird die Kugel mit den Kameras aus den selben Stellungen aufgenommen wie zuvor das Kalibrierobjekt. Für jede Stellung wird nun ein Fehlermaß berechnet. Dazu wird zunächst der Mittelpunkt und der Radius der Kugel im Kamerabild bestimmt. Dies geschieht durch eine Binarisierung mit einem empirisch ermittelten Schwellwert und eine anschließende Konturfindung. Nun wird mit Hilfe der initialen Kamerapose das Referenzmodell der Kugel in das Kamerabild projiziert, vgl. dazu Abb. 5.13. Für diese Projektion ist dann ebenfalls der Mittelpunkt und der Radius bekannt. Der Fehler für diese Projektion berechnet sich dann folgendermaßen:

$$E_{ges} = |M_i - M_p| + |r_i - r_p| \quad (5.1)$$



$M_i$  bezeichnet dabei den Mittelpunkt der Kugel im Kamerabild,  $M_p$  entsprechend den Mittelpunkt der projizierten Kugel im Bild.  $R_i$  bezeichnet den Radius der Kugel im Bild,  $r_p$  den Radius der projizierten Kugel. Alle Werte sind also in Pixeln gegeben. Der Gesamtfehler ergibt sich dann aus dem Abstand der beiden ermittelten Kugelmittelpunkte im Bild und dem Unterschied der Radien. Diese Fehlerfunktion ist nun die Grundlage für eine Optimierung basierend auf dem Rosenbrock-Verfahren [Rosenbrock 60], wobei jede Kamerapose einzeln für sich optimiert wird. Dabei wird in jedem Schritt eine neue Kamerapose berechnet, auf deren Basis dann eine neue Projektion der Kugel in das Bild durchgeführt wird. Mit dieser neuen Projektion kann dann erneut der Fehler berechnet werden. Das Verfahren terminiert wenn dieser Fehler minimal ist. Zu beachten ist hierbei, dass die möglichen Kameraposen stark eingeschränkt werden müssen um dem mechanischen Aufbau Rechnung zu tragen und nur kleine Abweichungen von der initial ermittelten Kamerapose zuzulassen.

## 5.6 Datenaufnahme

Der folgende Abschnitt beschreibt den vollständigen Prozess zur Digitalisierung eines Objektes. Dabei wird zunächst die dreidimensionale Gestalt des Objektes erfasst, anschließend aus verschiedenen Posen Aufnahmen mit dem Stereokamerasystem durchgeführt und diese anschließend zur Texturierung der 3D-Daten eingesetzt. Abschließend wird dem Objekt eine Klasse und ein Symbol zugewiesen und die Daten mit der Objektdatenbank synchronisiert. Eine Übersicht über diesen Prozess und die dabei entstehenden Daten findet sich in Abb. 5.14.

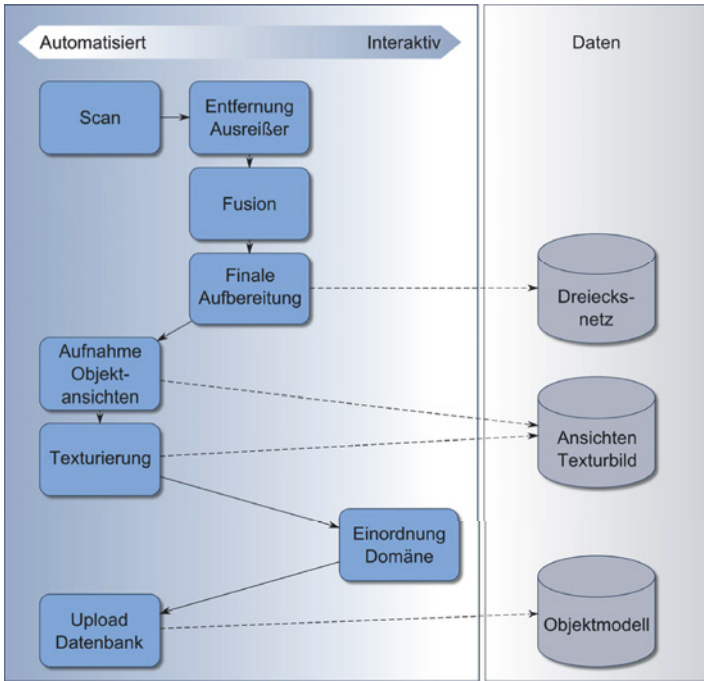


Abb. 5.14. Der Modellierungsprozess in der Übersicht mit Grad der Automatisierung und den resultierenden Daten.

### 5.6.1 3D Information

Die dreidimensionale Gestalt der Objekte, also eine Repräsentation ihrer räumlichen Ausdehnung, ist für die Anwendung in der Servicerobotik eine der wichtigsten Informationen. Sie kann genutzt werden um stabile Griffe für das Objekt zu berechnen, Objekterkennung und -lokalisierung im Zu-

sammenspiel mit einem 3D-Sensor durchzuführen oder eine realistische Darstellung der Objekte für eine Visualisierung zu erzielen.

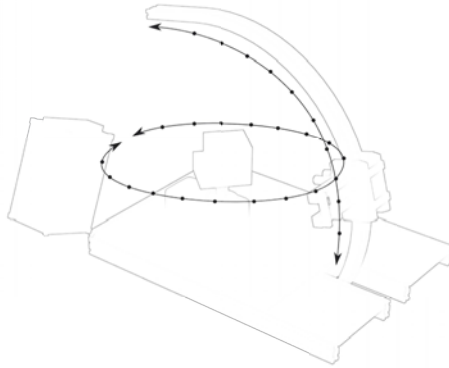
Die Erfassung der dreidimensionalen Gestalt eines Objektes erfolgt mit Hilfe des im Modellierungscenter verbauten Konica-Minolta Vi-900 Sensorsystems. Das genaue Vorgehen variiert dabei abhängig von der Form des zu erfassenden Objekts. Meist beginnt der Prozess jedoch mit der automatischen Erfassung mehrerer Teilscans mit Hilfe des Rotationstellers. Das Objekt wird darauf platziert und zu vorgegebenen Stellungen des Tellers je ein Scan durchgeführt (z.B. alle  $60^\circ$ ). Durch die Kalibrierung ist die Rotationsachse bekannt, was eine automatische Vorregistrierung der Einzelaufnahmen ermöglicht. Da i.d.R. jedoch, bedingt durch Verdeckungen, nicht alle Teile des Objekts durch diese Aufnahmen erfasst werden können, sind weitere Ansichten notwendig. Dazu wird das Objekt so vor dem Scanner positioniert, dass die vorher verdeckten Stellen vom Sensor erfasst werden können. Da bei diesen zusätzlichen Aufnahmen die Objektpose nicht automatisiert erfasst wird (wie bei der Rotation durch den Drehteller), müssen diese Aufnahmen manuell registriert werden. Dies kann durch Rotation und Translation oder durch die Angabe von drei Korrespondenzpunktpaaren geschehen. Ergebnis des Scanprozess an dieser Stelle sind mehrere zueinander grob registrierte Einzelaufnahmen (üblicherweise zwischen 8 und 12).

### **5.6.2 Objektansichten**

Neben der Aufnahme von 3D-Informationen sind auch Ansichten des Objektes aus unterschiedlichen Perspektiven von Interesse. Diese Daten können z.B. für das Training und die Evaluation einer auf Kamerabildern ba-

sierenden Objekterkennung und -lokalisierung verwendet werden. Daneben dienen diese Daten auch als Grundlage für die Texturierung der 3D-Datensätze.

Durch den gegebenen Aufbau des Modellierungscenars kann das Stereokamerasystem wie bereits beschrieben auf der Oberfläche einer gedachten Halbkugel oberhalb des Objektes positioniert werden. Die Schrittweite von einer Position zur nächsten ist dabei variabel, in der Praxis hat sich ein Wert von  $10^\circ$  sowohl für die Rotation des Drehtellers wie auch des Kameraschlittens entlang der Schiene als guter Kompromiss aus Anzahl an Kameraposen und Datenvolumen ergeben.



**Abb. 5.15.** Aufnahmeschema der Objektansichten. Das Objekt wird auf dem Rotationsteller gedreht, während die Kameras auf dem beweglichen Schlitten entlang verfahren werden.

Die Aufnahme der einzelnen Objektansichten erfolgt dann vollständig automatisiert. Da die Bewegung des Drehtellers schneller erfolgt als die Bewegung der Kameras entlang der Schiene, wird zunächst die Kamera ver-

---

**Algorithmus 1** Algorithmus für die Reihung der Kameraposen bei der automatisierten Aufnahme

---

*max<sub>a</sub>*: Maximalwinkel Kameraschlitten  
*step<sub>a</sub>*: Schrittweite Kameraschlitten  
*max<sub>i</sub>*: Maximalwinkel Drehteller  
*step<sub>i</sub>*: Schrittweite Drehteller

**Require:**  $max_a \leq 90^\circ$   
**Require:**  $max_i \leq 353^\circ$

**for**  $a = 0 \rightarrow max_a$  **do**  
  Kameras entlang Schiene zu Stellung  $a$  bewegen  
  **for**  $i = 0 \rightarrow max_i$  **do**  
    Drehteller in Stellung  $i$  rotieren  
    Kamerabilder aufnehmen  
     $i \leftarrow i + step_i$   
  **end for**  
   $a \leftarrow a + step_a$   
**end for**

---

fahren und dann in dieser Kamerastellung jede Drehtellerposition abgehandelt. Anschließend wird die Kamera in die nächste Position verfahren und dann wieder alle Drehtellerstellungen abgefahren. Dies geschieht bis alle gewünschten Ansichten aufgenommen wurden. Algorithmus 1 zeigt diesen Ablauf in formaler Darstellung, Abbildung 5.15 eine schematische Visualisierung.

Um die aufgenommenen Bilder weiter verwenden zu können ist eine intrinsische Kalibrierung der Kameras notwendig. Das Verfahren hierzu wird in Abschnitt 3.1.1 beschrieben. Mit Hilfe dieser Informationen können die Ergebnisbilder beispielsweise rektifiziert werden und eine 3D-Rekonstruktion durchgeführt werden. Um die Verwendung der Daten möglichst variabel zu gestalten und z.B. die Entwicklung oder Evaluierung

von Rektifizierungsalgorithmen zu ermöglichen werden die Bilddaten im Rohformat zusammen mit den Kalibrierungsdaten zur Verfügung gestellt. Es erfolgt lediglich eine Konvertierung in die verlustfreien Kompressionsformate TIF<sup>3</sup> und PNG<sup>4</sup>.

Weiterhin wird zu jedem aufgenommenen Objekt eine XML-Datei angelegt, in der zu jeder Objektansicht die aus der extrinsischen Kalibrierung abgeleitete Kamerapose und die Intensitäten der Beleuchtungseinheiten eingetragen sind. Listing 5.1 zeigt einen Ausschnitt aus solch einer Datei.

```
<?xml version="1.0" encoding="utf-8"?>
<Object Name="OrangeMarmelade" TranslationRotaryPlate="0/-38.66/0">
  <Images>
    <Image Camera="left" R1="0.050293" R2="0.007946" R3="-1.000280" R4=
      ="-0.013409" R5="-1.000220" R6="-0.005469" R7="-0.996782" R8=
      ="0.011310" R9="-0.050881" TransX="25.989300" TransY="
      25.753000" TransZ="613.002000" LightFront="70" LightBack="70"
      LightCenter="3">left\
      OrangeMarmelade_isel_0_amtec_0_lights_70_70_3_left.png|tif</
      Image>
    <Image Camera="right" R1="-0.076521" R2="0.019749" R3="-0.998534"
      R4="-0.008137" R5="-1.000110" R6="-0.016351" R7="-0.995169"
      R8="0.004110" R9="0.076428" TransX="10.272500" TransY="
      22.586300" TransZ="612.376000" LightFront="70" LightBack="70"
      LightCenter="3">right\
      OrangeMarmelade_isel_0_amtec_0_lights_70_70_3_right.png|tif</
      Image>
    ...
  </Images>
</Object>
```

**Listing 5.1.** Auszug aus einer XML-Datei generiert für das Objekt „OrangeMarmelade“

<sup>3</sup> Tagged Image Format

<sup>4</sup> Portable Network Graphics

## 5.7 Software und Nachbearbeitung

Nach der Aufnahme von Rohdaten erfolgt vor der weiteren Verwendung üblicherweise eine Nachbearbeitung, um die Qualität der Rohdaten zu verbessern oder diese eventuell in eine andere Form zu transformieren. Von den hier vorliegenden Daten betrifft dies lediglich die aufgenommenen 3D-Daten. Ergebnis des initialen Scan-Prozesses sind einzelne Dreiecksnetze, die unterschiedliche Bereiche des aufzunehmenden Objekts abdecken und rudimentär ausgerichtet sind. Diese gilt es in ein einzelnes Dreiecksnetz zu überführen und anschließend unter Verwendung der Objektansichten zu texturieren. Für die Nachbearbeitung der 3D-Daten wurde eine spezielle Anwendung auf Basis der Softwarebibliothek Rapidform DLL der Firma Inus Technology ([Inus Technology 10]) erstellt. Die Bibliothek stellt u.a. folgende Funktionalitäten zur Verfügung:

- Registrierung und Filterung von Punktwolken
- Entfernung von Ausreißern und Spikes (spitze Dreiecke)
- Triangulation von Punktwolken
- Fusion von Teilnetzen
- Reduktion der Netzkomplexität

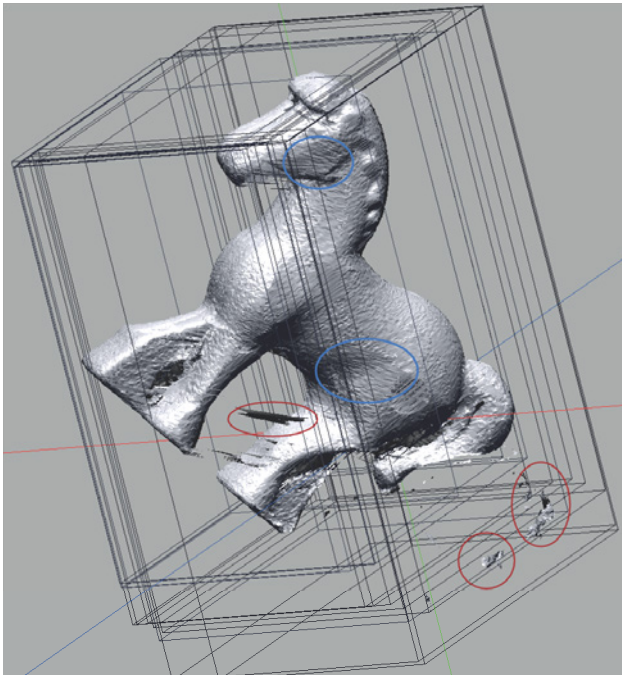
Leider ist zu diesen Funktionalitäten nicht definiert welche spezifischen Verfahren im einzelnen implementiert wurden, weshalb in den nachfolgenden Abschnitten der Vollständigkeit halber exemplarisch mögliche Verfahren angegeben werden, die jedoch nicht zwingend in der Arbeit verwendet werden.

### 5.7.1 Aufbereitung Dreiecksnetze

Nach der Aufnahme verschiedener Teilsfans durch das Sensorsystem werden die so gewonnenen 3D-Daten für die finale Fusion nachbearbeitet. Dies beinhaltet das Entfernen von Ausreißern und das manuelle Abschneiden von nicht zum Objekt gehörenden Punkten (z.B. der Sockel unter dem Objekt). Ein automatisches Verfahren zur Rauschreduzierung ist z.B. der räumliche Tiefpassfilter, wie er in [Pauly 01] beschrieben wird. Die Entfernung von Ausreißern kann ebenfalls über eine Abschätzung geschehen, die eine Likelihood für jeden Punkt berechnet, dass dieser auf der abgetasteten Oberfläche liegt, abhängig vom umgebenden Punkthaufen. [Schall 05] zeigt, dass auf solch gefilterten Punktwolken eine Oberflächenrekonstruktion besser durchführbar ist als auf verrauschten Daten. Abbildung 5.16 zeigt das Ergebnis der initialen Teilsfans. Diese sind lediglich durch die Stellung des Rotationstellers registriert und weisen noch erhebliche Mengen an Ausreißern auf, verursacht durch Sensorrauschen und Materialabhängigkeiten des Scanverfahrens. Neben den Ausreißern lassen sich auch Teilstücke beobachten, die aus sehr langgezogenen, spitzwinkligen Dreiecken bestehen. Verursacht werden diese wenn die Blickrichtung des Sensors quasi tangential über mehrere Kanten des Objekts verläuft, was dazu führt, dass Punkte auf der Oberfläche des Objekts, die sehr unterschiedliche Distanzen zum Sensor aufweisen, im Kamerabild des Sensors direkt nebeneinander liegen. Dies resultiert dann in einer falschen Triangulierung dieser Punkte.

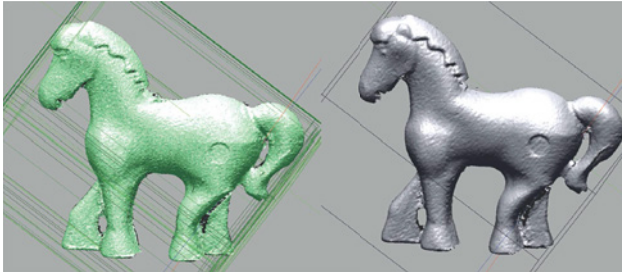
In einem nächsten Schritt werden die Einzelaufnahmen mit Hilfe des ICP-Verfahrens registriert. Dieser Schritt ist notwendig, da die Vorregistrierung durch die Stellung des Rotationstellers bzw. die manuelle Registrierung nicht genau genug sind, um eine qualitativ hochwertige Fusion





**Abb. 5.16.** Typische Fehler bei der 3D-Digitalisierung. Rot markiert sind Ausreißer durch Sensorrauschen, blau markiert sind spitzwinklige Dreiecke die durch Abrisse an Kanten entstehen können. Quelle: [Kasper 12a]

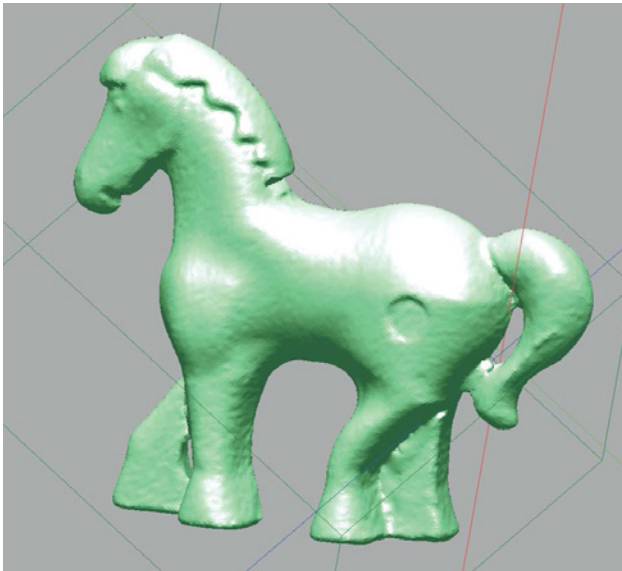
zu ermöglichen. Einen Überblick über häufig verwendete Varianten des ICP-Algorithmus für die Registrierung von Punktwolken findet sich in [Rusinkiewicz 01]. Nach der Feinregistrierung erfolgt die Fusion der Einzelaufnahmen zu einem einzigen Dreiecksnetz. Dabei werden sich überlagernde Dreiecke bzw. Punkte fusioniert und teilweise auch vorhandene Löcher bereits geschlossen (s. Abb. 5.17). Durch diesen Fusionsprozess entstehen weitere Artefakte, die nun ebenfalls bereinigt werden.



**Abb. 5.17.** Links: Ergebnis der Feinregistrierung mittels ICP, rechts: Ergebnis der Fusion. Quelle: [Kasper 12a]

Je nach Ausgangsdaten können im fusionierten Dreiecksnetz neben Löchern in der Oberfläche auch mannigfaltige Kanten, das heißt Kanten an die mehr als zwei Dreiecke angrenzen, vorhanden sein. Um automatisiert Löcher in der Objektoberfläche zu schließen, kann z.B. das Verfahren [Davis 02] verwendet werden. Dabei wird eine Distanzfunktion in der Umgebung der Löcher definiert, die den Wert 0 auf der Oberfläche annimmt. Diese wird dann durch einen Diffusionsprozess im Raum erweitert, bis dessen Nullmenge die vorhandenen Löcher überbrückt. Beim Füllen der Oberflächenlöcher sowie beim Entfernen der mannigfaltigen Kanten stoßen automatische Verfahren allerdings oft an Grenzen und es bedarf hier manueller Interaktion. Das Ergebnis der 3D-Datenaufnahme ist beispielhaft in Abbildung 5.18 dargestellt. Dieser Teil der Objektaufnahme ist deshalb auch der arbeits- und zeitintensivste Schritt.

Nach der Bereinigung und Reparatur des Dreiecksnetzes ist dieses 2D-mannigfaltig, nach Möglichkeit geschlossen und besteht durchschnittlich aus 150.000 Dreiecken. Im letzten Schritt dieser Phase wird dieses Netz nun noch auf je 25.000, 5.000 und 800 Dreiecke reduziert um die Komple-



**Abb. 5.18.** Ergebnis der 3D-Datenaufnahme und -nachbearbeitung. Quelle: [Kasper 12a]

xität für verschiedene Anwendungen wie z.B. eine Greifplanung zu verringern. Die Arbeiten [Luebke 01] und [Vogt 00] geben einen Überblick über verschiedene polygonale und punkt-basierte Reduktionsverfahren. Die vier Versionen des Dreiecksnetzes sind damit bereit zur Texturierung.

### 5.7.2 Texturierung

Ziel des Texturierungsschrittes ist es, aus einer Untermenge der Objektansichten, dem generierten Dreiecksnetz des Objekts und den Kameraposen ein einzelnes Texturbild zu erzeugen, welches mit der entsprechenden Ab-

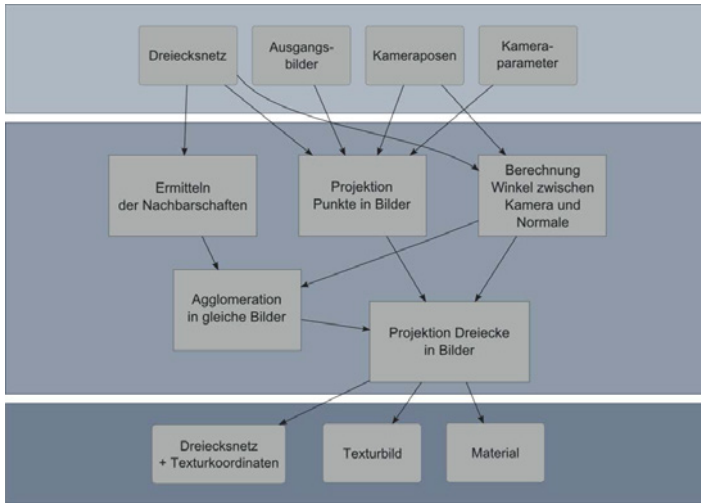


Abb. 5.19. Übersicht über das Texturierungsverfahren.

bildung die Oberfläche des Dreiecksnetzes mit Farbinformation vervollständigt. Das bedeutet, dass auf jedes Dreieck des Netzes ein bestimmter Ausschnitt aus einer der Objektansichten projiziert wird und somit die Farbe der Objektoberfläche innerhalb dieses Dreiecks durch die Farbinformation in diesem Bildausschnitt bestimmt ist. Abbildung 5.19 zeigt eine Übersicht über die Eingabedaten, die wichtigsten Verarbeitungsschritte und die Ausgabedaten, Abb. 5.20 das gewünschte Ergebnis.

In einem ersten Schritt werden alle 3D-Punkte des Objektes in alle Ausgangsbilder projiziert. Dazu werden die zu den jeweiligen Ausgangsbildern gehörenden Kameraposen (vorhanden in der XML-Datei) und die Kameraparameter verwendet. Die Projektionsergebnisse werden nun darauf geprüft, ob sie innerhalb des Bildes liegen. Falls ja wird das Bild



**Abb. 5.20.** Ziel des Texturierungsprozesses: Dreiecksnetz mit Farbinformation aus Objektansichten.

in die Liste der validen Bilder für diesen Punkt aufgenommen. Dieser Schritt dient dazu eventuell vorhandene Ausreißerpunkte, die in keinem Bild sichtbar sind, zu finden, damit diese später nicht mehr berücksichtigt werden müssen. Für die Berechnung in diesem Schritt sei  $P$  die Menge der Punkte und  $B$  die Menge der Objektansichten. Es wird also nun jeder Punkt  $\mathbf{p}_i \in P$  in jedes Bild  $b_j \in B$  mit Hilfe der zu  $b_j$  gehörenden Abbildungsmatrix  $A_j$  projiziert.  $A_j$  ergibt sich aus der Kamerapose  $T_j$  und der jeweiligen Projektionsmatrix  $C_l$  bzw.  $C_r$  der linken oder rechten Kamera des Stereosystems zu:

$$A_j = C_{lr} \cdot T_j$$

Die Projektion eines Punktes  $\mathbf{p}_i$  in ein Bild  $b_j$  auf  $\bar{\mathbf{p}}_i$ , ist dann:

$$\bar{\mathbf{p}}_i = A_j \cdot \mathbf{p}_i$$

Nun muss nur geprüft werden ob der projizierte Punkt innerhalb des Bildes liegt. Es muss also gelten:

$$0 \leq \bar{p}_{ix} \leq b_{ix} \wedge 0 \leq \bar{p}_{iy} \leq b_{iy}$$

Dabei ist  $b_{ix}$  die Breite und  $b_{iy}$  die Höhe des Bildes. Algorithmus 2 zeigt den Ablauf für diese Untersuchung.

---

**Algorithmus 2** Validitätsprüfung für die 3D-Punkte des Objekts und dessen Ansichten.

---

*P*: Menge der 3D-Punkte des Objektes

*B*: Menge der Objektansichten mit zugehörigen Kameraposen

**for**  $b_i \in B$  **do**

**for**  $p_j \in P$  **do**

    Projiziere  $p_j$  in  $b_i$  mit der Abbildungsmatrix  $C_i$

**if**  $p_j \in b_i$  **then**

$B_i$  gültig für  $p_j$

**end if**

**end for**

**end for**

---

Als weitere Vorbereitung auf die spätere Abbildung der Dreiecke in das Texturbild werden die Nachbarschaftsbeziehungen zwischen den Dreiecken des Netzes ermittelt. Dies dient später dazu, möglichst viele benachbarte Dreiecke in das selbe Ausgangsbild abzubilden. Dazu werden alle Kanten jedes Dreiecks betrachtet und überprüft, ob und zu welchem anderen Dreieck diese Kante ebenfalls gehört. Für jedes Dreieck wird so eine Liste von Nachbardreiecken angelegt.

In der Regel kann ein beliebiges Dreieck in mehr als ein einzelnes Bild projiziert werden, d.h. dieser Teil des Objektes ist aus mehreren Kamera-

perspektiven sichtbar (insbesondere da hier Verdeckungen außer Acht gelassen werden). Es gilt also, für die Bestimmung der Texturierung für jedes Dreieck ein optimales Ausgangsbild zu finden. Als Optimalitätskriterium für die Abbildung wird der Winkel zwischen der Normalen des Dreiecks und der Blickrichtung der Kamera des jeweiligen Bildes gewählt. Eine möglichst senkrechte Projektion, also ein möglichst kleiner Winkel, reduziert die Verzerrungen und erzeugt dadurch die beste Abbildung. Es wird also für jedes Dreieck des Netzes zu jedem gültigen Bild (d.h. dass dieses Bild für alle drei Punkte des Dreiecks gültig ist) dieser Winkel bestimmt. Dies ergibt für jedes Dreieck eine sortierte Liste von Bildern, beginnend mit dem Bild, dem der kleinste Winkel zugeordnet wurde. Die in Abb. 5.21 dargestellte Berechnung des Winkels geschieht durch folgende Beziehungen: Sei das Dreieck  $t_i \in M$  gegeben, wobei  $M$  die Menge aller Dreiecke des Netzes bezeichnet. Jedes Dreieck  $t_i$  sei gegeben durch drei Punkte  $p_{i1}$ ,  $p_{i2}$  und  $p_{i3}$ . Die normale  $\mathbf{n}_i$  des Dreiecks ergibt sich dann zu:

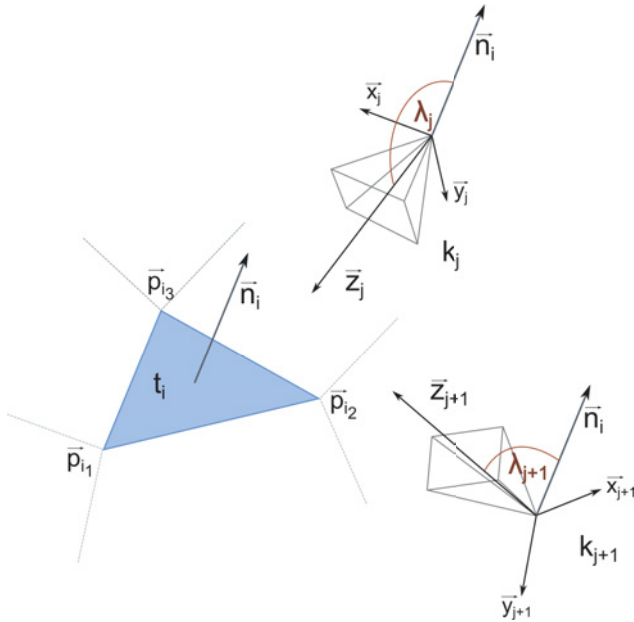
$$\mathbf{n}_i = (\mathbf{p}_{i2} - \mathbf{p}_{i1}) \times (\mathbf{p}_{i3} - \mathbf{p}_{i1}) \quad (5.2)$$

Als Blickrichtung für die Kamera  $k_j$ , die zu Bild  $b_j$  gehört, wird die positive z-Achse des Kamerakoordinatensystems,  $\mathbf{k}_j$  gewählt. Der Winkel zwischen der Normalen und der Blickrichtung ergibt sich dann aus

$$\cos(\lambda_j) = \frac{\mathbf{n}_i \cdot \mathbf{k}_j}{\|\mathbf{n}_i\| \cdot \|\mathbf{k}_j\|} \quad (5.3)$$

Der Ablauf dieser Berechnung ist in Algorithmus 3 zusammengefasst.

Würde nun einfach jedes Dreieck in das Bild mit dem so berechneten kleinsten Winkel abgebildet, kann es bei welligen Oberflächen oder Objekten mit vielen Vertiefungen vorkommen, dass benachbarte Dreiecke in



**Abb. 5.21.** Berechnung der optimalen Objektansicht für ein Dreieck. Es wird der Winkel zwischen der Kamerapose der Objektansicht und der Normalen des Dreiecks ermittelt.

---

**Algorithmus 3** Berechnung der Winkel zwischen Kamera und Dreiecken.

---

$M$ : Menge der Dreiecke des Netzes

$K$ : Menge der Kameraposen zu den Objektansichten

**for**  $k_i \in K$  **do**

**for**  $t_j \in M$  **do**

    Berechne Dreiecksnormale für  $t_j$

    Berechne Winkel zwischen Normale und Kamera

    Sortiere  $k_i$  in die Liste der Bilder für  $t_j$  abhängig von  $\lambda_j$

**end for**

**end for**

---



verschiedene Bilder abgebildet werden, die eigentlich zur gleichen Teiloberfläche (wie etwa einer Seite eines Würfels) gehören. An diesen Kanten bilden sich, bedingt durch unterschiedliche Helligkeiten zwischen den Bildern, stark kontrastierte Grenzen, die die Qualität des Ergebnis negativ beeinflussen. Zudem ist ein solches Abbildungsergebnis im Nachhinein für den Menschen nur schwer verständlich (z.B. bei der Bearbeitung in einem Modellierungsprogramm). Es ist also wünschenswert, dass möglichst alle Dreiecke einer Teiloberfläche in das selbe Ausgangsbild abgebildet werden. Um dies zu erreichen werden die Nachbarschaftsbeziehungen der Dreiecke ausgenutzt. Dazu wird die Liste der möglichen Bilder (sortiert nach Winkel) jedes Dreiecks mit den Listen der möglichen Bilder seiner Nachbardreiecke verglichen. Nun wird der Winkel jedes Bildes, welches sich auch in der Liste eines Nachbarn befindet, um einen bestimmten, einstellbaren Betrag verringert. Dies sorgt dafür, dass sich die Optimalität eines Bildes für ein Dreieck mit jedem Nachbarn erhöht, der ebenfalls in dieses Bild abgebildet werden kann. Die Agglomeration der Dreiecke ist

---

**Algorithmus 4** Agglomeration der Dreiecke in möglichst gleiche Ansichten.

---

```

M: Menge der Dreiecke des Netzes
Bi: Menge der validen Bilder für Dreieck ti
for t1i ∈ M do
  for t2j ∈ M/t1i do
    if t2j ist Nachbar von t1i then
      for b ∈ Bi ∩ Bj do
        Verringere λ von t1i um ε
      end for
    end if
  end for
end for

```

---

in Algorithmus 4 veranschaulicht.

Eine weitere mögliche Fehlabbildung ergibt sich bei der Betrachtung von konvexen Objekten. Hier kann es vorkommen, dass die Normale eines Dreiecks in Richtung der Kamera zeigt, das Dreieck selbst jedoch nicht von der Kamera aus gesehen werden kann, da es von anderen Flächen des Objekts verdeckt wird. Um diesen Fall auszuschließen, wird für jedes Dreieck mit Hilfe eines Strahltests von der Kamera zum Dreiecksmittelpunkt die Sichtbarkeit des Dreiecks geprüft.

Nach der Betrachtung der Nachbarschaften und der damit einhergehenden Agglomeration benachbarter Dreiecke in die gleichen Ausgangsbilder erfolgt die eigentliche Abbildung. Für jedes Dreieck werden nun die Texturkoordinaten im bestimmten optimalen Bild gespeichert, die Berechnung der Projektionen wurde bereits im ersten Schritt vorgenommen.

Nun kann aus den einzelnen Ausgangsbildern das resultierende Texturbild zusammengesetzt werden. Zunächst wird für jedes Ausgangsbild der minimale Ausschnitt mit Hilfe der minimalen und maximalen Texturkoordinaten bestimmt. Anschließend werden diese Ausschnitte in ein neues Bild eingefügt, welches aus zwei Reihen und entsprechend vielen Spalten besteht (s. Abb. 5.22). Abhängig von der Ausschnittwahl und der Position des Ausschnitts im Ergebnisbild müssen nun noch die Texturkoordinaten angepasst werden. Im letzten Schritt erfolgt die Ausgabe des Ergebnisses, also die Vervollständigung des Dreiecksnetzes mit Texturkoordinaten und die Speicherung des fusionierten Texturbilds.

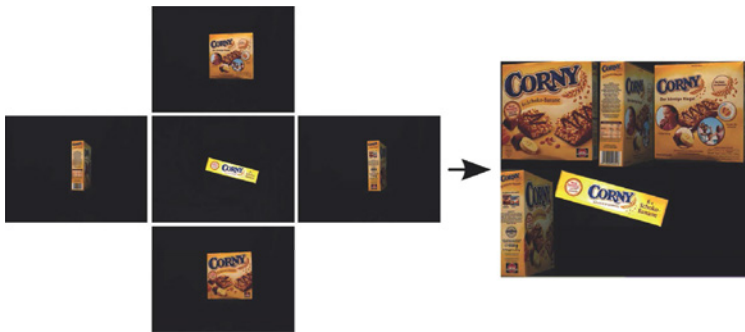


Abb. 5.22. Fusion der einzelnen Objektansichten zur Objekttextur.

## 5.8 Objektdatenbank

Die gewonnenen Objektmodelle sollen sowohl direkt in einem Robotersystem Verwendung finden, wie auch als Trainings-, Test- und Evaluationsdatensätze für die Entwicklung von Algorithmen und Methoden im Umfeld der Servicerobotik dienen. Um dies zu ermöglichen müssen verschiedene Kanäle geschaffen werden über die die Objektdaten von Dritten abgerufen werden können. Für den Zugriff zur Laufzeit des Roboters werden reduzierte Datensätze über eine dedizierte Schnittstelle zur Verfügung gestellt. Die Verteilung der Objektdaten an andere Forschungseinrichtungen und weitere Interessierte geschieht über ein Webinterface.

### 5.8.1 Webdatenbank

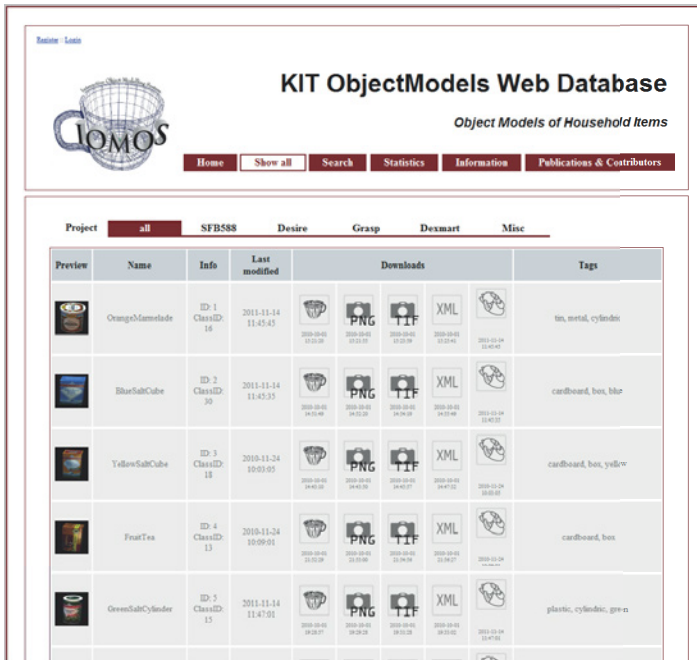
Der Einsatz der Objektdatensätze sollte sich von Anfang an nicht nur auf die Verwendung zur Laufzeit in einem Roboter beschränken, sondern auch

als Grundlage für die Entwicklung von Methoden und Algorithmen im Bereich der Objekterkennung und -lokalisierung, der Greifplanung und ähnlicher Gebiete dienen. Um die Daten einem möglichst breiten Publikum zugänglich zu machen wurde ein Webinterface entwickelt, welches den Zugriff auf die Datensätze ermöglicht.

Die Schnittstelle erlaubt über einen Webbrowser folgende Informationen einzusehen und Funktionalitäten zu nutzen:

- Darstellung aller vorhandenen Datensätze in einer Übersicht
- Einschränkung der Datensätze auf bestimmte Projekte
- Suche innerhalb der Datensätze nach Begriffen in Objektnamen, Beschreibungen und Tags
- Vorschau in der Übersicht mit Datum der letzten Aktualisierung
- Erweiterte Vorschau eines einzelnen Datensatzes und Quellennachweis
- Herunterladen von
  - 3D-Daten
  - Objektansichten in PNG- und TIF-Format
  - XML-Datei mit Kameraposen
  - Vorausberechneten Griffen für diverse robotische Greifsysteme

Das Webinterface erlaubt damit einen komfortablen Zugriff auf alle durch das System erzeugten Daten. Abb. 5.23 zeigt die Übersicht aller Objekte im Interface, Abb. 5.24 die Detailansicht für ein einzelnes Objekt.



The screenshot shows the KIT ObjectModels Web Database interface. At the top, there is a logo for IOMOS (Object Models of Household Items) and the title "KIT ObjectModels Web Database". Below the title, there are navigation links: Home, Show all, Search, Statistics, Information, and Publications & Contributors. The main content area displays a table of object models, with a "Project" dropdown menu set to "all". The table has columns for Preview, Name, Info, Last modified, Downloads, and Tags. The data rows are as follows:


























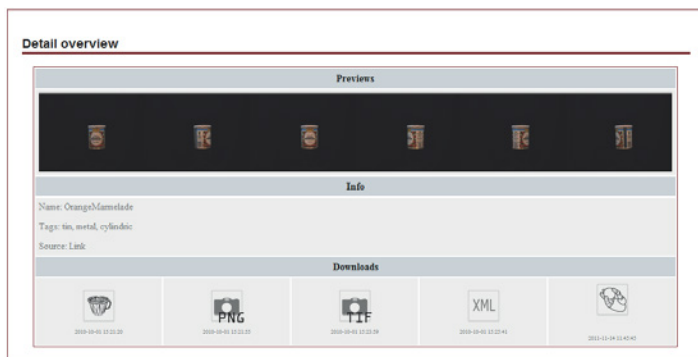
Project	all	SFB588	Desire	Grasp	Dexmart	Misc	
Preview	Name	Info	Last modified	Downloads			Tags
	OrangeCannetade	ID: 1 ClassID: 16	2011-11-14 11:45:45	  		tin, metal, cylinder	
	BlueSubCube	ID: 2 ClassID: 30	2011-11-14 11:45:35	  		cardboard, box, blue	
	YellowSubCube	ID: 3 ClassID: 18	2010-11-24 10:03:05	  		cardboard, box, yellow	
	FruitTea	ID: 4 ClassID: 13	2010-11-24 10:09:01	  		cardboard, box	
	GreenSubCylinder	ID: 5 ClassID: 15	2011-11-14 11:47:01	  		plastic, cylindrical, green	

Abb. 5.23. Weboberfläche der Objektdatenbank (Gesamtübersicht).

## 5.9 Zusammenfassung

Die vorgestellte Objektmodellierung basiert auf einem speziellen Sensoraufbau, der einen 3D-Sensor sowie ein Stereokamera paar beinhaltet. Durch die eingebaute Aktorik können sowohl das Objekt, wie auch die Kameras maschinell bewegt werden, was die Automatisierung des Aufnahmeprozesses ermöglicht. Es werden 3D-Daten in Form von Dreiecksnetzen und 2D-Bilddaten generiert und zu einem texturierten Gesamtmodell weiterverarbeitet. Der entwickelte Algorithmus zur Projektion der Ka-



**Abb. 5.24.** Weboberfläche der Objektdatenbank (Detailansicht).

merabilder auf das Dreiecksnetz ermöglicht es, zusammen mit der exakten Kalibrierung des Sensoraufbaus, diesen Prozess ebenfalls automatisiert ablaufen zu lassen. Die resultierenden Datensätze stehen Roboterentwicklern und -anwendern durch die entwickelte Webdatenbank in einem standardisierten Format zur Verfügung.

## **Modellierung von Szenen auf Basis von räumlichen Relationen**

### **6.1 Einleitung**

Nachdem in Kapitel 5 die Modellierung einzelner Objekte ausführlich betrachtet wurde, soll nun die Perspektive auf mehrere Objekte erweitert werden. Dies ist erforderlich, da sich die Interaktion mit der Umwelt eines Serviceroboters nicht auf vereinzelte Objekte beschränkt, sondern innerhalb einer komplexen Szene mit vielen verschiedenen Objekten stattfindet. Besitzt das System Hintergrundwissen über die Verbindungen und Zusammenhänge der verschiedenen Objekte innerhalb eines bestimmten Typs von Umgebung, erweitern sich die Interaktionsmöglichkeiten enorm. Veranschaulicht werden soll dies an folgender möglichen Anweisung, die einem Serviceroboter in einem Haushalt gegeben werden könnte: „Decke den Tisch für das Abendessen.“

Für eine menschliche Haushaltshilfe wäre das Erfüllen dieser Aufgabe kein Problem; für ein derzeitiges Robotersystem jedoch mit erheblichen

Schwierigkeiten verbunden, da dieses viele der zur Erledigung benötigten Informationen schlicht nicht besitzt. Welche Informationen sind dies?

Die Aufgabe soll in kleinere Teilprobleme zerlegt werden um zu illustrieren, welche Arten von Informationen über Objekte, eingebettet in eine Umgebung, hilfreich sein können. Eine erste Analyse ergibt, dass ein nicht näher spezifizierter Tisch gedeckt werden soll unter der Einschränkung, dass vermutlich an diesem Tisch dann von einer nicht angegebenen Anzahl Personen das Abendessen eingenommen wird. Es muss also ermittelt werden um welchen Tisch es sich höchst wahrscheinlich handelt. Nimmt man als Einsatzumgebung des Roboters eine durchschnittliche Wohnung an, finden sich hier mehrere mögliche Kandidaten (Küchentisch, Esstisch, Wohnzimmer Tisch, Wickeltisch, etc.). Hier sollte nun eine Verteilung vorhanden sein, welcher Tisch am wahrscheinlichsten in der gegebenen Situation zu verwenden ist. Nun muss bekannt sein was unter „den Tisch decken“ zu verstehen ist, aus Objektsicht also welche Objekte an dieser Aktion beteiligt sind. Der Zusatz „für das Abendessen“ schränkt diese Auswahl zusätzlich ein, bzw. fordert möglicherweise das Vorhandensein ganz bestimmter Objekte. Nachdem die benötigten Objekte identifiziert wurden, gilt es diese im Haushalt zu finden. Wo befinden sich Teller, Gläser und Besteck? Auch diese Frage ist durch spezielles Hintergrundwissen zu beantworten. Schließlich sollen die benötigten Utensilien auf dem Tisch platziert werden. Auch hier gilt es spezifische Anordnungen zu befolgen (Besteck neben dem Teller, Glas rechts oder links oberhalb des Tellers, etc.), die wiederum von der konkreten Aufgabenstellung abhängen. Für ein Abendessen sieht die Zusammensetzung des Besteckes u.U. schon je nach Art des Essens sehr verschieden aus. Ein Fünf-Gänge-Menü wird sicher mehr Besteck und Teller benötigen als ein Snack mit Sandwichs.



## 6.2 Problemstellung

Dem Beispiel aus der Einleitung folgend, können für die Szenenmodellierung folgende Fragestellungen ermittelt werden:

- Welche Objekte sind in einer bestimmten Umgebung zu erwarten?
- In welchen Anordnungen sind Objekte in bestimmten Umgebungen und Situationen anzutreffen?

Die zweite Fragestellung kann weiter spezialisiert werden:

- Welchen Einfluss hat das Vorhandensein eines bestimmten Objektes auf die in seiner Umgebung befindlichen Objekte?
- Sind bestimmte Typen von Objekten häufig an ähnlichen Orten zu finden?
- Befinden sich bestimmte Objekte häufig in einer bestimmten Anordnung zueinander?
- Kann von einem Objekt auf die Objekte in seiner Umgebung geschlossen werden?

Die Problemstellung ergibt sich also aus der Beobachtung alltäglicher Szenen. Es soll herausgefunden werden wie kontextuelles Wissen über Objekte gewonnen und angewendet werden kann. Der Mensch ist offensichtlich in der Lage bestimmte Objekte in Zusammenhang zu bringen und auch aus dem Vorhandensein und der Kombination von Objekten auf die aktuelle Lage zu schließen. Es gilt also, ähnlich wie bei der Objektmodellierung, zum einen herauszufinden wie der Mensch dieses Wissen akquiriert und wie dementsprechend ein Robotersystem an ähnliche Informationen

gelangen kann. Gleichzeitig gilt es Wege zu finden wie das Szenenwissen eines Menschen an das Maschinensystem weitergegeben werden kann.

### 6.3 Lösungsansatz

Da die Begriffe *Objektklasse*, *Objekt*, *Szenenklasse* und *Szene* im Kontext der Servicerobotik unterschiedlich interpretiert und verwendet werden, sollen kurze Definitionen an dieser Stelle Klarheit darüber schaffen, was in der vorliegenden Arbeit unter diesen Begriffen verstanden wird (s.a. [Kasper 11]):

**Definition 1 (Objektklasse und Objekt)** *Eine Objektklasse ist das allgemeine Konzept, welches Alltagsobjekte beschreibt, die gemeinsame Eigenschaften und Funktionalitäten haben, in ihrer räumlichen Ausdehnung beschränkt und beweglich sind (ihre Position im Raum kann sich im Laufe der Zeit verändern). Beispiele für Objektklassen sind Tasse, Stuhl, etc. Ein Objekt ist dann eine konkrete Instanz einer Objektklasse.*

**Definition 2 (Szenenklasse und Szene)** *Eine Szenenklasse ist das allgemeine Konzept einer begrenzten Region im Raum, die spezifische Eigenschaften hat und/oder einem bestimmten Zweck dient. Eine Szene ist dann eine konkrete Instanz einer Szenenklasse und besteht aus einer unbestimmten Anzahl an mehr oder weniger beweglichen Ob-*

*jekten und wird z.B. durch Wände oder Decken/Böden begrenzt, entspricht also einem Raum in einem Gebäude oder einem Ausschnitt davon. Beispiele für Szenenklassen sind: Badezimmer, Küche oder Schreibtisch.*

Um einem Robotersystem Hintergrundwissen über Szenen mit Objekten zu vermitteln, sollen reale Szenen beobachtet und ausgewertet werden. Dies bildet einerseits die wiederholte, alltägliche Beobachtung ähnlicher Szenen durch den Menschen nach, zum anderen unterstützt es die Anforderung, dass bereits das Hintergrundwissen, mit für Serviceroboter typischer Hardware, erworben werden soll, um die spätere Anwendung möglichst einfach und die zu erwartenden Ergebnisse gut vorhersagbar zu machen.

Um das bereits vorhandene Wissen eines Menschen über die Szenen zu nutzen, sollen die sensorisch erfassten Trainingsdaten von einem Benutzer annotiert und anschließend „maschinengerecht“ aufbereitet werden. Dies bedeutet, dass ein möglichst großer Satz an Trainingsdaten in Verteilungen umgewandelt wird, die später als Basis für eine maschinelles Lernverfahren dienen. Dies ermöglicht es dem Robotersystem selbständig Hypothesen zu bilden, die möglichst sichere Antworten auf die in der Problemstellung angesprochenen Fragen geben können. Grundsätzlich können verschiedenste Aspekte der annotierten Daten ausgewertet und als Hintergrundwissen eingesetzt werden. In der vorliegenden Arbeit liegt der Fokus auf den räumlichen Beziehungen zwischen den einzelnen Objekten und dabei speziell auf den Relationen „ist auf“ und „ist neben“.

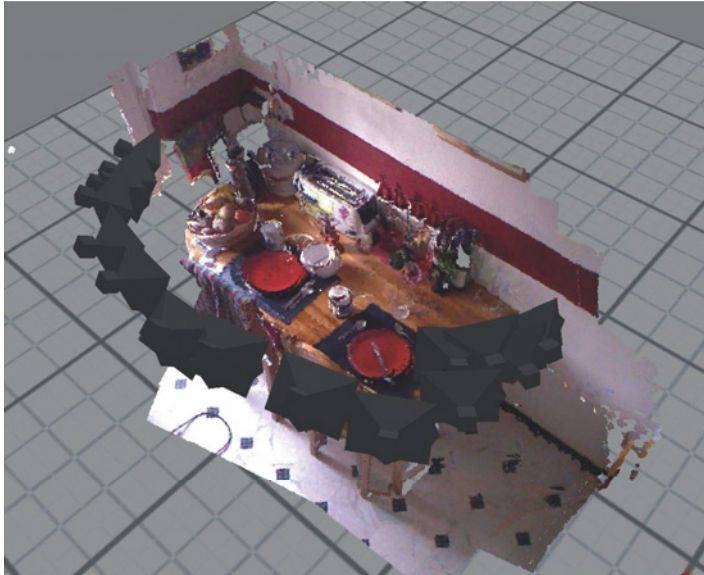
Schließlich kann diese Art Hintergrundwissen jederzeit durch weitere Beobachtungen, die u.U. nicht mehr von einem menschlichen Benutzer annotiert sein müssen, erweitert und verbessert werden.

## 6.4 Datenakquise

Wie bereits eingangs erwähnt, soll die Datengrundlage auf der das Hintergrundwissen aufgebaut wird, bereits mit Sensorik aufgezeichnet werden, die einem heutigen Serviceroboter zur Verfügung steht. Dies schränkt die möglichen Modi ein auf visuelle, akustische oder taktile Informationen. Für die Untersuchung von räumlichen Relationen, also hauptsächlich relativer Objektposen, eignen sich visuelle Informationen am besten, da diese eine großflächige und schnelle Erfassung der gesamten Szene ermöglichen, wie auch Objekte erfassen, die u.U. keine Geräusche erzeugen und eine hohe Genauigkeit aufweisen.

Betrachtet man die gängigen visuellen Sensoren von Servicerobotern, so finden sich verschiedenste 2D- und 3D-Sensoren, beispielsweise Kameras und Laserscanner. Für die digitale Reproduktion der realen Szene hinsichtlich der räumlichen Auswertung bietet sich die Verwendung von 3D-Sensorik an, da hierdurch direkt auch die benötigte Tiefeninformation zur Verfügung steht. Einen Überblick über mögliche Techniken mit Hilfe solcher Sensorik eine Szene zu rekonstruieren gibt [Liu 11]. In der Praxis hat sich für diese Aufgabe vor allem der auf Musterprojektion (vgl. 3.2.2) basierende Sensor der Firma PrimeSense (bekannt durch die Verwendung in Kinect<sup>TM</sup>) bewährt. Der Sensor erzeugt kolorierte Tiefendaten in einer Auflösung von 640x480 Pixeln, bei einer maximalen Aufnahme Frequenz

von 30Hz in einem Bereich von 1,2-3,5 m und Sichtfeld von  $57^\circ$  horizontal bzw.  $43^\circ$  vertikal ([Wikipedia, E. 12]). Diese Eigenschaften sind nahezu ideal für die Erfassung typischer Haushaltsszenen.



**Abb. 6.1.** Ergebnis der Szenendigitalisierung mittels Kinect-Sensor und SLAM-Verfahren mit visualisierten Kameraposen der Einzelaufnahmen.

Aufgrund der Sichtfeldeinschränkung des Sensors und der aus dem Messverfahren resultierenden Verdeckungen, reicht eine einzige Aufnahme nicht aus um die Szene vollständig zu erfassen. Die Aufnahme mehrerer Perspektiven und deren Fusion zu einer Gesamtaufnahme sind die typischen Problemstellungen im Bereich SLAM (Simultaneous Localisation And Mapping - Simultane (Selbst)Lokalisierung und Kartierung). Für

die Szenendigitalisierung mit Hilfe des Kinect-Sensors wurde das SLAM-Verfahren von [Engelhard 11] verwendet und erweitert. Das Verfahren nutzt dabei geschickt aus, dass der Sensor sowohl Tiefen- als auch Farbdaten liefert. Zunächst wird auf den 2D-Kamerabildern eine Merkmalsextraktion durchgeführt, die zu einer Reihe von sog. SURF<sup>1</sup>-Merkmalen führt. Diese werden dann mit den im vorherigen Schritt extrahierten Merkmalen verglichen und entsprechende Korrespondenzen bestimmt. Im Bereich der so gefundenen Korrespondenzpunkte wird die Tiefeninformation ausgewertet, was zu einer Reihe von 3D-Punkt-Paaren führt, die wiederum die Schätzung einer Transformation zwischen den beiden Aufnahmen erlaubt. Zur Schätzung dieser Transformation wird das RANSAC<sup>2</sup>-Verfahren eingesetzt. Durch die so gewonnene Transformation können sukzessive Aufnahmen in das Koordinatensystem der ersten Aufnahme transformiert werden. Um die initiale Schätzung der Transformation zu verbessern, wird in einem weiteren Verarbeitungsschritt die neu aufgenommene Punktwolke mittels ICP<sup>3</sup> an die letzte Punktwolke angepasst und die Transformation so verbessert. Über den gesamten Verlauf der Aufnahme werden die einzelnen Aufnahmen in einen Graphen eingetragen, der schließlich eine letzte Optimierung ermöglicht. Die Ausgabe dieses Verfahrens ist letztendlich eine farbige Punktwolke, die alle Einzelaufnahmen umfasst, wobei diese in ein gemeinsames Koordinatensystem transformiert wurden.

Um als Ausgangspunkt für die Annotierung zu dienen, wurde dieses Verfahren angepasst und teilweise erweitert. Zunächst wird die Ergebnispunktwolke mit einem gitterbasierten Filter von Ausreißern befreit und

---

<sup>1</sup> Speed Up Robust Features

<sup>2</sup> Random Sample Consensus

<sup>3</sup> Iterative Closest Point, [Zhang 92]

eine homogene Punktdichte erzeugt. Die Gitterlänge beträgt dabei 1 cm, was einen empirisch ermittelten Kompromiss zwischen der Sensorauflösung und dem Detailgrad der Punktwolke darstellt. Weiterhin werden die zu den einzelnen Aufnahmen gehörenden 2D-Farbbilder mit der jeweiligen Kamerapose abgespeichert um später bei der Annotierung als zusätzliche Informationsquelle genutzt werden zu können.

## 6.5 Annotierung

Nachdem im ersten Schritt die reale Szene dreidimensional digital rekonstruiert wurde, sollen nun die darin befindlichen Objekte gefunden und identifiziert werden. Um den Erfahrungsschatz des Menschen nutzen zu können, gleichzeitig eine gemeinsame Basis zwischen Mensch und Maschine zu schaffen und zusätzlich eine möglichst hohe Qualität der Daten zu erreichen, wird für die Objektidentifikation auf die Annotierung durch menschliche Benutzer zurückgegriffen. Dieses Verfahren ist im Bereich des maschinellen Lernens, besonders im Rahmen von Objekterkennung und -lokalisierung, weit verbreitet ([Russell 08], [Yao 09], [Deng 09]).

Für die Annotierung wurde eine spezielle Software entwickelt, die es ermöglicht die aufgenommene Punktwolke zu importieren und zu visualisieren. Die importierte Punktwolke muss nun zunächst so orientiert werden, dass entweder die Boden- oder eine vorhandene Tischfläche parallel zur  $x$ - $y$ -Ebene liegt (dies ist relevant für die spätere Auswertung der Relationen). Der Benutzer markiert anschließend alle für die Szene relevanten Objekte. Falls vorhanden, kann dafür ein vorab erzeugtes 3D-Modell verwendet werden, andernfalls eine orientierte Boundingbox. Diese werden so in der

3D-Szene positioniert, dass sie möglichst deckungsgleich mit dem betreffenden Ausschnitt aus der Punktwolke sind. Um bei der Identifikation und der Positionierung zu helfen, wird das aktuelle Annotierungselement mit Hilfe der gespeicherten Kameratransformationen gleichzeitig in die 2D-Farbbilder projiziert. Aufgrund der begrenzten räumlichen Auflösung des 3D-Sensors erleichtert dies dem menschlichen Auge in vielen Fällen die korrekte Erkennung des betreffenden Szenenobjekts. Neben der Positionierung in der Szene und der Auswahl eines geeigneten Referenzmodells bzw. der Dimensionierung der Boundingbox ordnet der Benutzer das Objekt einem Konzept in WordNet [Stark 98] zu. Ein Annotierungselement besteht also aus:

- Position und Orientierung relativ zur Szene
- 3D-Referenzmodell oder Boundingbox mit angepasster Größe
- Verweis auf Konzept in WordNet

Die Gesamtheit der Annotierungselemente macht schließlich die Annotierung aus. Alle Annotierungen werden für die spätere Auswertung in einer Datenbank gespeichert.

Da eine vollständig manuelle Markierung aller Objekte in einer komplexen Szene sehr zeitaufwändig ist, wird der Benutzer auf verschiedene Arten vom System unterstützt. Für die Ausrichtung der Szene steht ein automatisches Verfahren, basierend auf dem RANSAC<sup>4</sup>-Algorithmus, zur Verfügung. Hierbei wird versucht eine Ebene in die gegebene Punktwolke einzupassen. Ist eine entsprechend große planare Fläche (wie Boden oder Tisch) in der Punktwolke vorhanden, so kann deren Orientierung bestimmt werden und damit die gesamte Punktwolke entsprechend neu ausgerichtet

---

<sup>4</sup> [Fischler 81]



werden. Weiterhin kann eine automatische Vorauswahl an möglichen Objektkandidaten durchgeführt werden, für die dann bereits in Größe, Position und Orientierung angepasste Boundingboxen in der Szene platziert werden. Die Identifizierung möglicher Objektkandidaten geschieht mit Hilfe des Verfahrens der euklidischen Ballung, wie sie in [Rusu 09] beschrieben ist. Zunächst muss für die gesamte Punktwolke eine k-d-Baum-Datenstruktur erstellt werden, um das Finden nächster Nachbarn möglichst effizient und einfach zu gestalten. Ein k-d-Baum [Bentley 75] ist eine effiziente Datenstruktur zur Unterteilung von k-dimensionalen Räumen. Jeder Knoten des Baums besteht aus einem k-dimensionalen Punkt, wobei jeder Knoten, der kein Blatt des Baums ist, als Hyperebene verstanden wird, die den Raum in zwei Teile unterteilt. Die Punkte im Unterbaum dieses Knotens liegen dann alle entweder links oder rechts von dieser Hyperebene. Diese Eigenschaft ermöglicht eine effiziente Suche nach nächsten Nachbarn eines gegebenen Punktes, da hierfür lediglich die entsprechenden Unterbäume untersucht werden müssen. Für die euklidische Ballung werden nun alle Punkte der Punktwolke untersucht. Zu jedem Punkt wird die Untermenge aller Punkte gesucht, die sich innerhalb einer Kugel mit einem gegebenen Radius um den Punkt befinden. Punkte aus dieser Untermenge, die noch zu keiner anderen Ballung gehören, werden in die aktuelle Ballung aufgenommen. Algorithmus 5 zeigt den Pseudocode für dieses Vorgehen.

Nachdem mögliche Objektkandidaten mit diesem Verfahren gefunden wurden, muss der Benutzer lediglich falsch erkannte Objekte löschen, Größe und Position anpassen und ein entsprechendes Konzept zuweisen. Abbildung 6.2 zeigt den Weg von der importierten Punktwolke hin zur fertig annotierten Szene. In der Praxis zeigt sich, dass die vorgestellten

---

**Algorithmus 5** Algorithmus für die euklidische Ballung (nach [Rusu 09])

---

Erzeuge k-d-Baum für Punktwolke  $P$   
**Require:** Leere Liste von Ballungen  $C$   
**Require:** Liste von Punkten, die abgearbeitet werden müssen  $Q$   
**for**  $p_i \in P$  **do**  
    Füge  $p_i$  zu  $Q$  hinzu  
    **for**  $p_i \in Q$  **do**  
        Finde Untermenge  $P_i^k$  von Nachbarpunkten in Kugel um  $p_i$  mit  
        Radius  $r < d_{th}$   
        **for**  $p_i^k \in P_i^k$  **do**  
            Falls  $p_i^k$  noch nicht bearbeitet, füge zu  $Q$  hinzu  
        **end for**  
    **end for**  
    Füge  $Q$  zu  $C$  hinzu  
**end for**

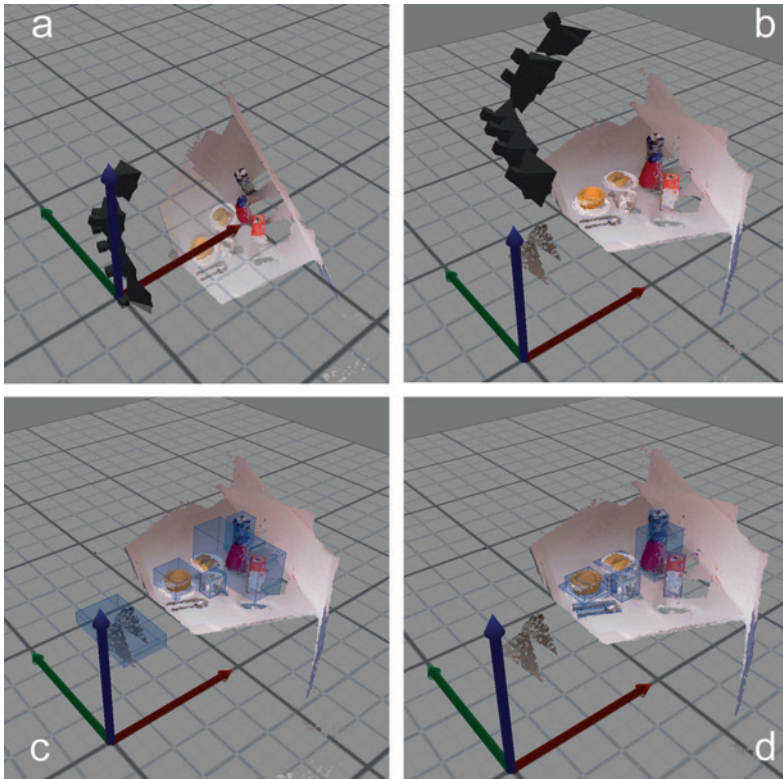
---

Hilfsfunktionen den Zeitaufwand für eine Annotierung zum Teil erheblich verkürzen können.

## 6.6 Relationen

Die Motivation für die Szenenmodellierung ist die Beobachtung, dass in einer natürlichen Alltagsumgebung die dort zu findenden Objekte in der Regel nicht zufällig verteilt, sondern vielmehr entsprechend spezifischer Zusammenhänge zwischen den Objekten, angeordnet sind. Besonders relevante Relationen sind dabei:

- ein Objekt befindet sich *auf* einem anderen
- ein Objekt befindet sich *neben* einem anderen



**Abb. 6.2.** Ablauf der Szenenannotierung. a: Importierte Punktwolke; b: automatisch ausgerichtete Punktwolke; c: Ergebnis der euklidischen Ballung; d: fertig annotierte Szene.

Die Implikationen, die sich aus der Beobachtung dieser räumlichen Relationen ergeben, sowie eine Möglichkeit diese mathematisch zu modellieren und anhand von vorhandenen Daten zu berechnen, werden in den folgenden Abschnitten beschrieben.

### 6.6.1 „Ist Auf“-Relation

Befindet sich ein Objekt A auf einem Objekt B, so dient B als unterstützendes Objekt für A. Diese Verbindung zweier Objekte lässt sich im Alltag häufig beobachten; so befinden sich Tassen und Teller häufig auf einem Tisch, Bücher sind auf dem Bücherregal gestapelt usw. Die Ist-Auf-Relation teilt die verschiedenen Objektklassen so u.a. in Objekte, die auf andere gelegt werden können, und in Stützobjekte. Dies kann bei der Suche nach einem Objekt von Nutzen sein, in dem der mögliche Suchraum z.B. auf die Oberfläche eines Tisches reduziert wird, oder bei der Planung von Handlungsabläufen, indem klar ist, dass sich auf dem zu bewegendem Objekt noch ein weiteres befindet und diese nicht rigide miteinander verbunden sind.

Es gibt viele verschiedenen Möglichkeiten eine solche räumliche Relation zu definieren und im Anwendungsfall zu berechnen; eine Übersicht hierzu findet sich in [Cohn 01]. Die in dieser Arbeit verwendete Methode wurde von Sjöö et al. entwickelt und wurde ausgewählt, da ihre spezielle Formulierung Ungenauigkeiten, die durch die Datenaufnahme mit Sensoren entstehen, berücksichtigt. Sie stellt sich wie folgt dar (vgl. [Sjöö 10]).

Sjöö et al. modellieren eine räumliche Relation als Funktion, abhängig von den involvierten Objekten (A und B), die aus dem Raum aller möglicher Objektposen in das Intervall  $[0, 1]$  abbildet:

$$\mathbf{R}_{A,B} : \{\pi_A, \pi_B\} \rightarrow [0, 1] \quad (6.1)$$

Dabei bedeutet ein Funktionswert von 1, dass die Relation vollständig erfüllt ist, wohingegen ein Wert von 0 so zu interpretieren ist, dass die Rela-

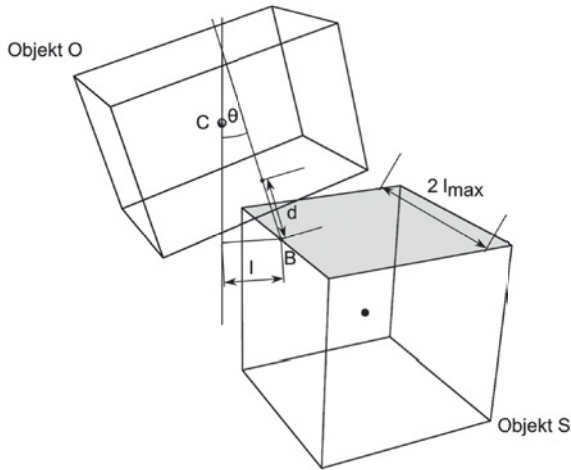
tion nicht erfüllt ist. Es handelt sich hierbei also um eine Art Fuzzyifizierung der Relation, die den Vorteil bietet, dass Ungenauigkeiten beim sensorischen Erfassen der Objekte berücksichtigt werden. Im Gegenzug macht dies jedoch allgemeine Aussagen über Reflexivität, Symmetrie und Transitivität solcherlei definierter Relationen unmöglich. Sjöö et al. bauen in [Sjöö 11] zwar ein axiomatisches System auf den Relationen  $On$ ,  $On_t$  und  $In$  auf, fundieren dieses jedoch nicht in den mathematischen Termen zur Berechnung, sondern leiten dieses lediglich aus der Anschauung ab.

Die spezielle „Ist-Auf“-Relation zwischen einem Stützobjekt  $S$  und dem Trajektorobjekt  $O$ , wird mit  $On(O, S)$  bezeichnet und ihr Wert hängt von drei Kriterien ab:

- Abstand zwischen den Objekten
- Horizontaler Abstand zwischen dem Massenschwerpunkt und dem Kontaktpunkt
- Angriffswinkel der Schwerkraft

Der Abstand zwischen den Objekten meint dabei den kürzesten Abstand zwischen den Objektflächen. Damit ein Objekt von einem anderen gestützt werden kann, müssen sich diese berühren. Aufgrund von Ungenauigkeiten in der Sensorerfassung kann es allerdings auch zu einem negativen Abstand kommen, der eine scheinbare Durchdringung der beiden Gegenstände ausdrückt. Um diesen Ungenauigkeiten zu begegnen wirkt sich der Abstand negativ auf den Relationswert aus, je größer also der Abstand, desto geringer der Relationswert.

Der horizontale Abstand zwischen dem Schwerpunkt und dem Kontaktpunkt beschreibt die Tatsache, dass ein Gegenstand nur dann auf einem



**Abb. 6.3.** Berechnung der Ist-Auf-Relation. Hierbei bezeichnet  $d$  den minimalen Objektabstand,  $\theta$  den Winkel zur Gravitationskraft,  $l$  den horizontalen Abstand des Schwerpunktes ( $C$ ) zum Berührungspunkt ( $B$ ) und  $l_{max}$  den maximal möglichen Abstand zum Berührungspunkt innerhalb der Kontaktfläche.

andere ruhen kann, wenn sich sein Schwerpunkt horizontal innerhalb der Kontaktfläche mit dem stützenden Gegenstand befindet. Auch dieser Abstand wirkt sich negativ auf den Relationswert aus.

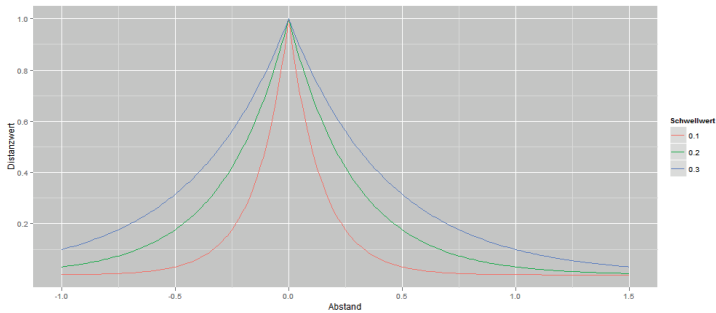
Der Angriffswinkel der Schwerkraft, also der Winkel zwischen der Normalen des Kontaktpunktes und der Richtung der Schwerkraft, fließt in die Berechnung des Relationswertes ein, um das mögliche Abrutschen eines Objektes von seinem Stützobjekt zu berücksichtigen. Aus der Physik ist bekannt, dass die Kraft entlang der Kontaktnormalen mit dem Kosinus des Winkels zur Gewichtskrafttrichtung abnimmt.

Sjöö et al. entwickeln aus diesen drei Kriterien zwei Faktoren, aus denen sich der endgültige Relationswert ergibt. Der Distanzfaktor ergibt sich zu:

$$On_{distance}(O, S) \triangleq \exp\left(-\frac{d}{d_0(d)} \ln(2)\right) \quad (6.2)$$

Abbildung 6.4 zeigt den Verlauf dieser Funktion für den Abstandsbereich im Intervall  $(-1, 1, 5)$ . In Gleichung 6.2 bezeichnet  $d$  dabei den minimalen Abstand zwischen den Objekten  $O$  und  $S$  und  $d_0$  die Distanz bei der der Wert der Relation sich auf die Hälfte verringert, wobei  $d_0^+$  und  $d_0^-$  (unterschiedliche) positive Werte (ohne und mit Durchdringung) sind:

$$d_0 = \begin{cases} -d_0^-, & d < 0 \\ d_0^+, & d \geq 0 \end{cases}$$

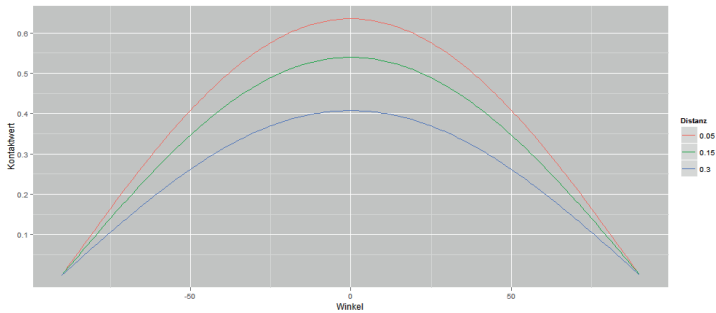


**Abb. 6.4.** Beispielhafter Verlauf der Funktion für den Distanzanteil der Ist-Auf-Relation, hier für drei verschiedene Schwellenwerte  $d_0$ .

Der Kontaktfaktor ergibt sich aus dem horizontalen Abstand des Schwerpunktes zum Kontaktpunkt und dem Angriffswinkel der Schwerkraft:

$$On_{contact}(O, S) \triangleq \cos(\theta) \cdot \frac{1 + \exp(-(1-b))}{1 + \exp\left(-\left(\frac{-l}{l_{max}} - b\right)\right)} \quad (6.3)$$

Dabei bezeichnet  $l$  den horizontalen Abstand des Schwerpunktes zum Kontaktpunkt,  $l_{max}$  den maximal möglichen Abstand innerhalb der Kontaktfläche und  $b$  einen möglichen Offset, welcher üblicherweise zu 0 gewählt wird. Abbildung 6.3 zeigt die wichtigen Größen und deren Zusammenhang dieser Berechnung. Abbildung 6.5 zeigt den Verlauf der Kontaktfunktion für den Bereich von  $-90^\circ$  bis  $90^\circ$  für den Winkel  $\theta$ .



**Abb. 6.5.** Beispielhafter Verlauf der Funktion für den Kontaktanteil der Ist-Auf-Relation, hier für drei verschiedene Distanzwerte  $d$ .

Für den endgültigen Relationswert wird nun das Minimum der beiden Faktoren gebildet:

$$On(O, S) \triangleq \min(On_{distance}, On_{contact}) \quad (6.4)$$

Dies berücksichtigt, dass die am wenigsten erfüllte Bedingung die finale Aussage macht, wie sehr sich Objekt  $O$  auf Objekt  $S$  befindet.



### 6.6.2 „Ist Neben“-Relation

Im Gegensatz zur „Ist-Auf“-Relation, bei der sich exakt formulieren lässt unter welchen Umständen sie gilt, lässt sich die Frage, wann sich ein Objekt  $A$  *neben* einem Objekt  $B$  befindet, weniger eindeutig beantworten. Trotzdem wird diese räumliche Anordnung vor allem im täglichen Sprachgebrauch häufig verwendet, z.B. bei der Beschreibung wo ein Objekt zu finden sei, spielt dessen relative Position zu anderen Objekten eine wichtige Rolle. Anzumerken ist dabei, dass oftmals spezialisierte Versionen der „Ist-Neben“-Relation, wie „links von“ oder „vor“ bzw. „hinter“ benutzt werden. Diese sind aber immer nur in einem speziellen Kontext, hauptsächlich von einer spezifischen Perspektive aus sinnvoll und wahr, was die mathematische Formulierung zusätzlich erschwert bzw. einschränkt. Deshalb wird an dieser Stelle das allgemeinere „Ist neben“ zugrunde gelegt.

Analog zur „Ist-Auf“-Relation wird die „Ist-Neben“-Relation aus mehreren Termen gebildet, die sich aus den Posen und den Ausdehnungen der beiden beteiligten Objekte berechnen lassen. Auch hier soll die Relation als Funktion abhängig von den Objektposen betrachtet werden, die auf einen Wert zwischen 0 (Relation ist nicht erfüllt) und 1 (Relation ist vollständig erfüllt) abbildet, s. Gleichung 6.1.

Die Relation, ob sich ein betrachtetes Objekt  $O$  neben einem Referenzobjekt  $R$  befindet, wird mit  $Next(O, R)$  bezeichnet. Für die Berechnung des Relationswertes sind zwei Faktoren entscheidend:

- Abstand zwischen den Objekten
- Relative Höhe der Objektzentren

Der Abstand zwischen den Objekten meint hier wieder den minimalen Abstand zwischen den Objektoberflächen, wobei ein negativer Abstand eine Durchdringung signalisiert. Damit sich ein Objekt neben einem anderen befinden kann, sollten sich diese nicht durchdringen. Allerdings werden zwei Objekte, die sich zu weit voneinander entfernt befinden, allgemein nicht als nebeneinander liegend betrachtet. So würde beispielsweise das Glas auf dem Küchentisch als neben der Spülmaschine auf der anderen Seite des Raumes liegend gelten. Um zu entscheiden, ob zwei Objekte nebeneinander sind, muss der Objektabstand in Relation zu den Objektgrößen betrachtet werden. Im Idealfall berühren sich zwei nebeneinander befindliche Objekte, der Abstand beträgt also 0. Für die Abnahme des Relationswertes wird ein spezieller Normierungsfaktor verwendet ab dem dieser Term kleiner als 0,5 ist. Der Normierungsfaktor bestimmt sich aus den Ausdehnungen der beteiligten Objekte in die drei Raumrichtungen. Hierfür ergibt sich also folgender Term:

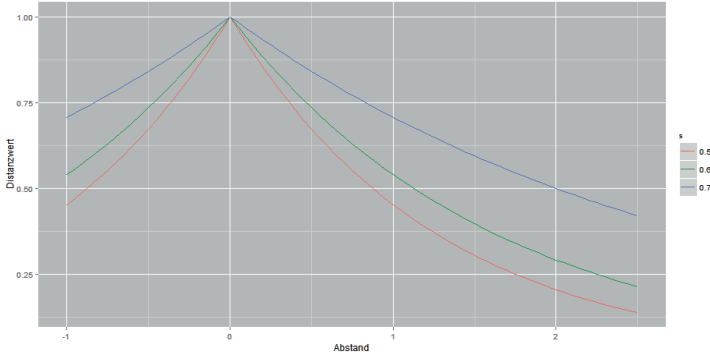
$$Next_{distance}(O,R) \triangleq \exp\left(-\frac{d}{s(d)} \ln(2)\right) \quad (6.5)$$

Hierbei bezeichnet  $d$  den minimalen Abstand zwischen den Objekten (wie in Gl. 6.2) und  $s$  die größte Ausdehnung des kleineren Objekts ( $O$  bzw.  $R$ ) in der  $x$ - $y$ -Ebene, wobei gilt:

$$s_{max}(d) = \begin{cases} -\frac{1}{4} \cdot s, & d < 0 \\ s, & d \geq 0 \end{cases}$$

Für eine negative Distanz wird dabei der Schwellwert auf  $1/4$  reduziert, um eine Durchdringung der beiden Objekte stärker zu bestrafen und den Relationswert entsprechend stärker zu reduzieren. Abbildung 6.6 zeigt den

Verlauf dieser Funktion für drei beispielhafte Werte von  $s$ . Der Verlauf ist analog zu 6.4.

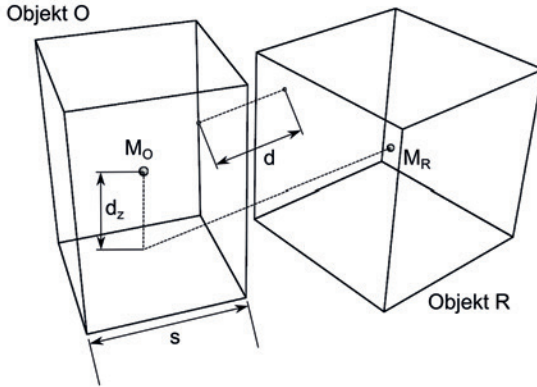


**Abb. 6.6.** Beispielhafter Verlauf der Funktion für den Distanzanteil der Ist-Neben-Relation, hier für drei verschiedene Schwellwerte  $s$ .

Die relative Höhe der Objektzentren ist der Anschauung nachempfunden, dass die Relation „neben“ nur auf Objekte angewendet wird, die sich in etwa auf gleicher Höhe befinden. Der Mittelpunkt von  $O$  sollte sich hierfür vertikal zwischen dem oberen und unteren Ende des Referenzobjekts  $R$  befinden. Analog zu Gleichung 6.5 ergibt sich:

$$Next_{vertical}(O, R) \triangleq \exp\left(-\frac{d_z}{s(d_z)} \ln(2)\right) \quad (6.6)$$

Dabei bezeichnet  $d_z$  den vertikalen Abstand der Objektzentren, also  $O_z - R_z$ . Für  $s$  gilt wieder die gleiche Fallunterscheidung wie oben, es wird hier jedoch lediglich die Ausdehnung in  $z$ -Richtung betrachtet. Abbildung 6.7 zeigt diese Größen und deren Zusammenhang.



**Abb. 6.7.** Berechnung der Ist-Neben-Relation. Hierbei bezeichnet  $d$  den minimalen Objektabstand und  $d_z$  den vertikalen Abstand der beiden Objektzentren ( $M_O$  bzw.  $M_R$ ), während  $s$  die Ausdehnung von  $O$  bzw.  $R$  ist (je nachdem welches Objekt kleiner ist).

Diese beiden Terme werden schließlich zum Gesamtwert über eine gewichtete Summe zusammengesetzt:

$$Next(O, R) \triangleq w \cdot Next_{distance} + (1 - w) \cdot Next_{vertical} \quad (6.7)$$

Die gewichtete Summierung basiert auf der Annahme, dass sowohl die vertikale Verschiebung, wie auch die horizontale Distanz zum Relativwert beitragen, jedoch unterschiedlich stark. Die Bestimmung des Gewichtungsfaktors  $w$  erfolgt mit Hilfe einer Benutzerstudie in Abschnitt 8.3.1.

## 6.7 Statistische Daten

Neben der Berechnung der Relationen „Ist auf“ und „Ist neben“, können während der Szenenannotierung gleichzeitig noch weitere Daten erhoben werden. Wichtig für die spätere Auswertung der Annotationsdaten ist vor allem das Auftreten der jeweiligen Objektklassen in der Szene. Zum einen die Anzahl an Instanzen pro Szene:

$$S_C = |\{o \in O | class(o) = C\}| \quad (6.8)$$

Hierbei bezeichnet  $C$  die aktuell betrachtete Objektklasse und  $o$  ein Objekt aus der Menge der Objekte  $O$  in der Szene. Aus diesen Zählungen kann dann eine Verteilung gebildet werden, die Auskunft gibt wie viele Instanzen einer bestimmten Klasse in einer Szene erwartet werden können.

Weiterhin sollen der Höhenversatz, also die Distanz in z-Richtung, sowie der Abstand parallel zum Boden, also in der x-y-Ebene, zwischen zwei Objekten betrachtet werden. Diese Betrachtungen basieren auf [Fisher 10], in deren Arbeit gezeigt wird, dass diese beiden Werte durchaus charakteristisch sind für die räumliche Anordnung von Objekten in Szenen. Die Werte werden, jeweils ausgehend von den Zentren der betrachteten Objekte A und B, berechnet:

$$d_v = |A_z - B_z| \quad (6.9)$$

$$d_h = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2} \quad (6.10)$$

Das Ergebnis dieser Berechnungen sind statistische Verteilungen über die Abstände zwischen verschiedenen Objektklassen.

Schließlich können für alle Objekte die jeweiligen Ausdehnungen in x-, y- und z-Richtung bestimmt werden und daraus ebenfalls Verteilungen für die Größen, speziell des Volumens der Objektklassen gebildet werden. Diese Angaben sind z.B. bei der Erkennung von Objekten interessant um Plausibilitätsprüfungen durchzuführen oder den Suchraum entsprechend einzugrenzen.

## 6.8 Auswertung und mögliche Anwendung

Um das Hintergrundwissen, welches in den fertigen Annotierungen und den daraus resultierenden Relationen enthalten ist, in anderen Kontexten anwenden zu können, muss es in einer entsprechenden Form repräsentiert werden. Wird eine möglichst große Anzahl an Szenen annotiert können aus den Werten für die jeweiligen Relationen Verteilungen gebildet werden, die schließlich Grundlage für verschiedene Methoden, wie etwa einem Bayes'schen Schätzer, sein können.

Ausgangspunkt der Annotierungsauswertung sind die folgenden Mengen:

- Die Menge  $O_C$  der Objektklassen
- Die Menge  $O_S$  der Objekte in Szene S
- Die Menge  $S$  der Szenen

Der Benutzer führt dabei während der Annotierung die Zuordnung der Objekte zu den Klassen durch:

$$z : O_S \mapsto O_C \tag{6.11}$$

Die Berechnung der Relationen erfolgt nun für jede Szene separat, wobei innerhalb der Szene die jeweiligen Relationen für alle möglichen Objektpaare berechnet werden, wie in Algorithmus 6 beschrieben. Aus der

---

**Algorithmus 6** Algorithmus für die Berechnung der Objektrelationen

---

```

for  $S_i \in S$  do
  for  $O \in O_{S_i}$  do
    Inkrementiere Anzahl von  $z(O)$  für Szene  $S_i$ 
  end for
  for  $(O_j, O_k) \in O_{S_i}$  do
    if  $O_j \neq O_k$  then
      Berechne Relation  $R(O_j, O_k)$ 
      Speichere Tupel  $(O_j, O_k, R)$ 
    end if
  end for
end for

```

---

Menge an Tupeln kann dann durch Diskretisierung der Relationswerte die Verbundwahrscheinlichkeit für das Auftreten eines bestimmten Relationswertes für zwei Objektklassen ermittelt werden. Aus dieser lässt sich die a-posteriori Wahrscheinlichkeit für die Zugehörigkeit eines an der Relation beteiligten Objekts, abhängig vom Relationswert und der Objektklassenzugehörigkeit des ersten Objekts, nach dem Satz von Bayes ableiten:

$$P(R, O_1, O_2) = P(R|O_1, O_2) \cdot P(O_1, O_2) \quad (6.12)$$

$$P(O_2|R, O_1) \cdot P(R, O_1) = P(R|O_1, O_2) \cdot P(O_1, O_2) \quad (6.13)$$

$$\Rightarrow P(O_2|R, O_1) = \frac{P(R|O_1, O_2) \cdot P(O_1, O_2)}{P(R, O_1)} \quad (6.14)$$

Hierbei ist  $P(O_1, O_2)$  die Wahrscheinlichkeit, dass die Objekte  $O_1$  und  $O_2$  gemeinsam in einer Szene auftreten.  $P(R|O_1, O_2)$  beschreibt die Wahrscheinlichkeit, dass die Relation zwischen diesen Objekten den Wert  $R$  annimmt. Das Produkt dieser beiden Wahrscheinlichkeiten wird schließlich mit Hilfe der Wahrscheinlichkeit für den Relationswert  $R$  von Objekt  $O_1$  zu irgendeinem Objekt normiert.

Analog zu Gleichung 6.14 kann die a-posteriori Wahrscheinlichkeit für die Zugehörigkeit eines Objektes  $O$  anhand des festgestellten Volumens  $V$  ermittelt werden:

$$P(V, O) = P(V|O) \cdot P(O) \quad (6.15)$$

$$P(O|V) \cdot P(V, O) = P(V|O) \cdot P(O) \quad (6.16)$$

$$\Rightarrow P(O|V) = \frac{P(V|O) \cdot P(O)}{P(V)} \quad (6.17)$$

Eine mögliche Anwendung ergibt sich etwa für den Fall, dass zwei Objekte beobachtet werden, jedoch nur für eines der beiden die Klassenzugehörigkeit bestimmt oder angenommen werden kann. Über die Auswertung der Relationswerte für diese Konfiguration kann nun eine Verteilung über die Klassenzugehörigkeit des zweiten Objektes ermittelt werden. Dies kann z.B. genutzt werden um eine Objekterkennung zu unterstützen.

In der vorliegenden Arbeit werden die Wahrscheinlichkeitsverteilungen für die Relationen „ist auf“ und „ist neben“, sowie für das Objektvolumen verwendet um die Annotierung neuer Szenen zu unterstützen. Sobald mindestens ein Objekt der Szene annotiert wurde, können für alle weiteren Objekte, ausgehend vom Hintergrundwissen, Hypothesen aufgestellt und



dem Benutzer als Entscheidungshilfe angezeigt werden. In dieser Situation sei also Folgendes gegeben:

- ein Objekt  $O_1$  mit unbekannter Objektklasse  $U$
- ein Objekt  $O_2$  mit bekannter Objektklasse  $A$
- die Menge der bekannten Objektklassen  $C$

Es werden zunächst folgende Werte berechnet:

$$R_1 = On(O_1, O_2)$$

$$\bar{R}_1 = On(O_2, O_1)$$

$$R_2 = Next(O_1, O_2)$$

$$V = Volume(O_1)$$

Es wird davon ausgegangen, dass  $O_1$  zu einer bereits bekannten Objektklasse gehört. Mit Hilfe der oben ermittelten Werte können nun, auf Basis der vorangegangenen Annotierungen, folgende Wahrscheinlichkeiten für alle  $B \in C$  ermittelt werden:

$$P_1(B) = P(B|A, R_1)$$

$$\bar{P}_1(B) = P(B|A, \bar{R}_1)$$

$$P_2(B) = P(B|A, R_2)$$

$$P_V(B) = P(B|V)$$

Man erhält also für jede bekannte Objektklasse und jeden Relations- bzw. Eigenschaftswert eine Wahrscheinlichkeit. Um eine Gesamtwahrscheinlichkeit für jede mögliche Objektklasse zu erhalten, werden die jeweiligen Werte für eine bestimmte Objektklasse  $B$  nun wie folgt kombiniert:

$$P(B) = \max(P_1(B), \bar{P}_1(B)) \cdot P_2(B) \cdot P_V(B) \quad (6.18)$$

Dabei wird zwischen  $P_1(B)$  und  $\bar{P}_1(B)$  das Maximum gewählt, da die „ist auf“-Relation nicht kommutativ ist und daher davon ausgegangen werden muss, dass einer der beiden Werte in den meisten Fällen 0 sein wird. Die in Abschnitt 6.7 ebenfalls vorgestellten Abstandsrelationen werden hier nicht berücksichtigt, da diese bereits in den komplexeren Relationen „ist auf“ und „ist neben“ enthalten sind. Abgesehen von dieser Einschränkung wird hier somit ein naiver Bayes-Klassifikator eingesetzt, in dem nun diejenige Klasse  $U$  für Objekt  $O_1$  gewählt wird, für die gilt:

$$P(U) = \max(\{P(B)|B \in C\})$$

Befinden sich weitere Objekte mit bekannter Klasse ( $O_i$ ) in der Szene, wird diese einfache Berechnung erweitert. Die oben aufgeführten Berechnungen werden nun für alle bereits klassifizierten Objekte und dem gesuchten unbekanntem Objekt durchgeführt. Man erhält nun für alle  $O_i$  eine Menge  $P_i = \{P_i(B)|B \in C\}$ . Es wird angenommen, dass die berechneten Wahrscheinlichkeiten für jedes bekannte Objekt unabhängig sind. Daher können die zu gleichen potentiellen Klassen gehörenden Wahrscheinlichkeiten  $P_i(B)$  für alle  $i$  miteinander verknüpft werden. Dies ergibt schließlich für jede mögliche Ergebnisklasse eine Gesamtwahrscheinlichkeit:

$$P_{ges}(B) = \prod_i P_i(B)$$

Es wird wiederum die Klasse mit der größten Gesamtwahrscheinlichkeit

als beste Schätzung angenommen:

$$P_{ges}(U) = \max(\{P_{ges}(B) | B \in C\})$$

## 6.9 Zusammenfassung

Die in diesem Kapitel vorgestellte Szenenmodellierung bedient sich räumlicher Relationen zwischen Gegenständen in Alltagsumgebungen. Konkret werden die Relationen „ist auf“ und „ist neben“ als Funktionen modelliert, die abhängig von den relativen Objektposen Werte zwischen 0 und 1 annehmen. Durch die Digitalisierung von realen Alltagsszenen und deren Annotierung durch einen Benutzer wird eine Menge von Verteilungen über das Auftreten und die Relationswerte zwischen den beobachteten Objekten erzeugt. Zusätzlich zu den räumlichen Relationen wird das Objektvolumen als weitere statistische Größe eingeführt. Die gewonnenen Verteilungen der Wahrscheinlichkeiten von Relationen, Volumina und Auftreten bilden schließlich das probabilistische Szenenmodell. Diese kann nun in verschiedenen Anwendungen, bspw. zur Perzeption in einem Robotersystem, weiterverwendet werden. Die Beispielanwendung zeigt, wie mit Hilfe des Szenenmodells und einem oder mehrerer bekannter Objekte in einer Szene, mögliche Identifikationen für ein unbekanntes Objekt ermittelt werden können.



## Implementierte Softwarewerkzeuge

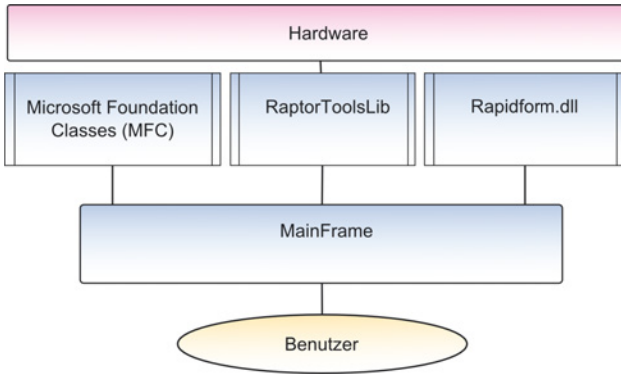
Im folgenden Kapitel werden die verschiedenen Applikationen und Bibliotheken, die für die Evaluierung und Realisierung der beschriebenen Methoden und Techniken implementiert wurden, beschrieben. Auf eine genaue Auflistung der einzelnen Klassen und Strukturen wird zugunsten der Übersichtlichkeit verzichtet. Alle Applikationen und Bibliotheken wurden in C++<sup>1</sup> implementiert.

### **7.1 Raptor (Rapid Textured Object Generator) - Applikation zur Aufnahme der 3D-Daten**

Die Raptor-Applikation bildet zusammen mit der im nächsten Abschnitt vorgestellten RaptorTools-Applikation das Herzstück der Objektmodellierung. Die Applikation stellt eine grafische Benutzeroberfläche zur Verfügung, die die Steuerung der Hardware im Modellierungscenter ermöglicht.

---

<sup>1</sup> <http://www.isocpp.org>



**Abb. 7.1.** Struktur der Raptor-Applikation

Gleichzeitig ermöglicht das Programm die Aufnahme der 3D-Daten und deren Nachbearbeitung mit Hilfe der kommerziellen Softwarebibliothek Rapidform.dll (vgl. B.2). Eine interaktive 3D-Visualisierung auf Basis von OpenGL<sup>2</sup> dient der schnellen Rückmeldung für den Benutzer. Eine Auswahl der mit dem Programm möglichen Bearbeitungsschritte wurde bereits in Abschnitt 5.7.1 vorgestellt.

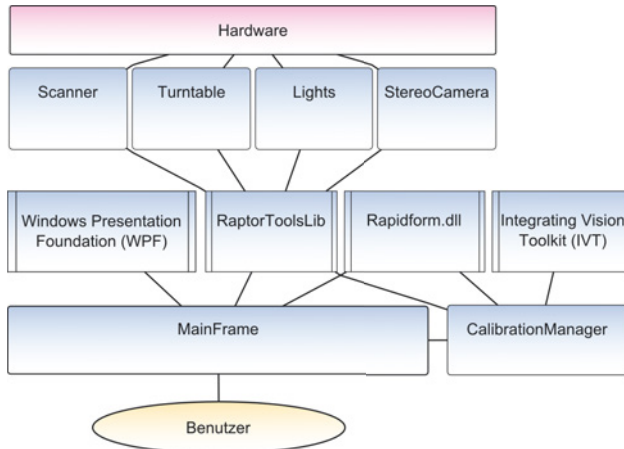
Für die Erzeugung der grafischen Benutzeroberfläche wird das Softwareframework der Microsoft Foundation Classes (MFC) verwendet. Dieses stellt Implementierungen für typische Oberflächenelemente wie Knöpfe oder Dialoge zur Verfügung. Die für die Abstraktion der Hardware entwickelte Bibliothek RaptorToolsLib wird in Abschnitt 7.2 vorgestellt.

Abbildung 7.1 zeigt den strukturellen Aufbau der Applikation mit den internen Verbindungen der verschiedenen Bibliotheken. MainFrame bezeichnet hierbei das Hauptfenster der grafischen Benutzeroberfläche. Di-

<sup>2</sup> Open Graphics Library - <http://www.opengl.org>

verse Dialoge zum Im- und Export von Daten oder Einstellungen wurden der Übersichtlichkeit halber nicht eingezeichnet.

## 7.2 RaptorTools - Applikation zur Aufnahme der Bilddaten



**Abb. 7.2.** Struktur der RaptorTools-Applikation

Um die Applikation zur Aufnahme der 3D-Daten nicht mit Funktionalität zu überladen und gleichzeitig eine modularisierte Entwicklung zu ermöglichen, wurden die Funktionen zur Aufnahme der Bilddaten mit Hilfe des Stereokamerasystems in eine eigene Applikation ausgelagert. Diese diente zusätzlich auch zur Entwicklung und dem Test der für die Ansteuerung und Inbetriebnahme des Hardwareaufbaus notwendigen Bibliotheken.

Wichtigster Bestandteil der sog. RaptorTools-Software ist die Bibliothek zur Abstraktion der verwendeten Hardware: RaptorToolsLib. Diese enthält Klassen für die einzelnen Hardwarekomponenten: Scanner, Rotationssteller, Beleuchtung und Stereokamerasystem. Diese Klassen vereinfachen den Zugriff auf die unterliegenden Low-Level-APIs, in dem spezielle Methoden zur Verfügung gestellt werden, die auf die spezifischen Bedürfnisse beim Betrieb des Modellierungscenars zugeschnitten sind. Unter anderem wird hierbei die SCSCI-Schnittstelle zum Konica-Minolta Vi-900, die serielle Kommunikation mit dem Rotationssteller und der Beleuchtung sowie die Verwendung des IEEE1394-Treibers der Kameras gekapselt.

Für die Interaktion mit dem Benutzer wurde wiederum eine grafische Benutzeroberfläche entworfen und implementiert, basierend auf den Windows Presentation Foundation (WPF) Klassen von Microsoft. Dieses moderne Softwareframework stellt ausgereifte Komponenten zur Visualisierung von Bildern, Bearbeitung von XML<sup>3</sup>-Dateien und weitere benötigte Funktionalitäten zur Verfügung.

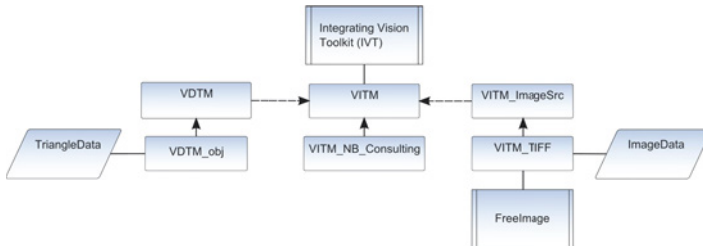
Eine besondere Komponente der Applikation stellt der CalibrationManager dar. Diese Klasse bündelt alle Funktionalitäten um den gesamten Hardwareaufbau zu kalibrieren. Dabei wird neben dem Zugriff auf die Hardware, die 3D-Softwarebibliothek Rapidform.dll (s. Abschnitt 7.1 und B.2) zur Lokalisierung des Kalibrierobjekts, sowie die Bildbearbeitungsbibliothek IVT (Integratign Vision Toolkit, vgl. Abschnitt B.6) zur Positionsbestimmung und Kalibrierung der Kameras verwendet. Abbildung 7.2 stellt die Zusammenhänge grafisch dar.

---

<sup>3</sup> Extensible Markup Language



## 7.3 TextureMapping - Applikation zur automatischen Texturerzeugung



**Abb. 7.3.** Struktur der TextureMapping-Applikation

Um die automatisierte Erzeugung der Texturen für die generierten Objekte modularisiert und getrennt entwickeln zu können, wurde für diese Aufgabe eine eigene Applikation entwickelt. Abbildung 7.3 zeigt die wichtigsten Klassen und Programmteile. Die Implementierung wurde in zwei Hauptteile aufgespalten: zum einen ein visualisierungsabhängiger Teil (VDTM), welcher das konkrete Format, in dem die 3D-Daten vorliegen abstrahiert; zum anderen ein visualisierungsunabhängiger Teil (VITM), der die eigentliche Texturzuordnung und -generierung durchführt. Hierdurch können verschiedene Formate leicht integriert werden. Im vorliegenden Fall wurde eine entsprechende Implementierung für das Wavefront OBJ-Format<sup>4</sup> erstellt (VDTM\_obj). Diese ermöglicht das Laden und Speichern der Objektdaten in diesem weit verbreiteten Format.

Das Laden, Verändern und Abspeichern der Bilddaten wird ebenfalls abstrahiert (VITM\_ImageSrc) und muss für spezifische Bildformate imple-

<sup>4</sup> [http://en.wikipedia.org/wiki/Wavefront\\_.obj\\_file](http://en.wikipedia.org/wiki/Wavefront_.obj_file)

mentiert werden (hier geschehen für das Tagged Image File Format, TIFF (VITM\_TIFF)). Für diese Implementierung wurde die Bibliothek FreeImage (s. Abschnitt B.8) verwendet.

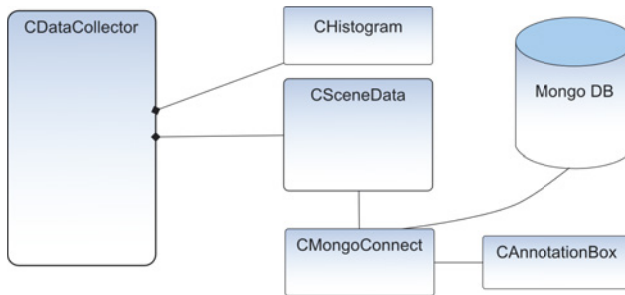
Die VITM-Klasse führt nun 3D-Daten und Bilddaten in einer verallgemeinerten Form zusammen. Der eigentliche Mapping-Algorithmus ist wiederum in eine spezifische Implementierung ausgelagert um die Möglichkeit zu bieten, verschiedene Algorithmen zu testen. In der vorliegenden Arbeit wurde ein Algorithmus implementiert, der das Mapping basierend auf einer Untersuchung der Dreiecksnachbarschaften durchführt, wie bereits in Abschnitt 5.7.2 beschrieben. Diese verwendet, neben den aus der XML-Datei des im Modellierungszentrums erzeugten Objektdatensatzes, die Kalibrierungsdaten des Stereokamerasystems zur Projektion. Hierfür wird wiederum die Bildbearbeitungsbibliothek Integrating Vision Toolkit (IVT) verwendet.

## **7.4 AnnotationLib - Bibliothek für die Relationsberechnung**

Im Bereich der Szenenmodellierung wurde die Funktionalität zur Berechnung der Interobjektrelationen und deren probabilistische Modellierung in eine eigene Bibliothek ausgelagert. Abbildung 7.4 zeigt den Aufbau der Bibliothek mit den wichtigsten Klassen und Modulen. Da die Rohdaten der Annotierung in einer Mongo<sup>5</sup>-Datenbank abgelegt werden, benötigt die Bibliothek zum Abruf der Daten eine Schnittstelle zu dieser Datenbank, welche die Klasse CMongoConnect zur Verfügung stellt. Diese

---

<sup>5</sup> <http://www.mongodb.org>

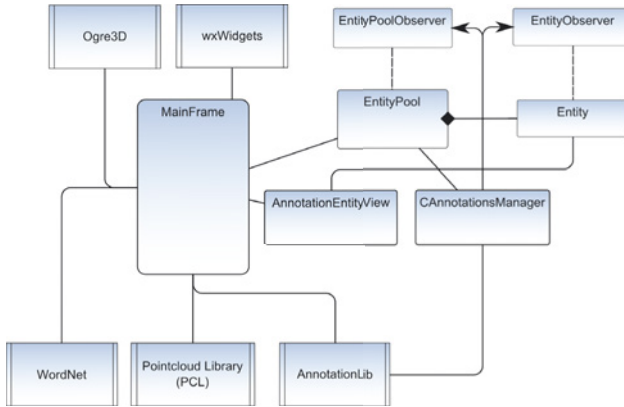


**Abb. 7.4.** Struktur der AnnotationLib-Bibliothek

wandelt die Datenbankeinträge in spezielle Container vom Typ CAnnotationBox um, die anschließend für die Berechnungen verwendet werden können. Die Klasse CSceneData kapselt alle Annotierungen und die daraus resultierenden Daten einer einzelnen Szene. Die übergeordnete Klasse CDataCollector sammelt nun alle Szenendaten und verarbeitet diese mit Hilfe von Histogrammen (CHistogram) zum endgültigen probabilistischen Modell. Gleichzeitig stellt die Klasse ein Interface zum Zugriff auf das Modell dar.

## 7.5 OViSEAnnotation - Applikation zur Annotierung von 3D-Szenen

Die zentrale Applikation im Rahmen der praktischen Umsetzung der Szenemodellierung stellt das Programm OViSEAnnotation dar. Diese basiert auf dem Framework „Ogre Virtual Scene Environment (OViSE)“ (vgl. Abschnitt B.5) und erweitert dieses um die speziell für die Annotierung von 3D-Punktwolken benötigten Funktionalitäten. Das Programm stellt eine



**Abb. 7.5.** Struktur der OViSEAnnotation-Applikation

grafische Benutzeroberfläche (erzeugt mit der Bibliothek wxWidgets, s. Abschnitt B.4) mit integrierter 3D-Visualisierung (basierend auf der Bibliothek Ogre3D, vgl. Abschnitt B.3) zur Verfügung. OViSE repräsentiert Objekte intern als sog. Entitäten, die von einem Entitäten-Pool verwaltet und mit Hilfe frei implementierbarer Plugins spezifisch anpassbar visualisiert werden. Diese Implementierung stellt in der Annotierungsapplikation die Klasse AnnotationEntityView zur Verfügung.

Da die Annotierungsobjekte spezielle Eigenschaften haben, die über die internen Entitäten hinausgehen, wurde ein spezieller AnnotationsManager implementiert, der die eingegebenen Annotierungsobjekte verwaltet. OViSE bietet die Möglichkeit mit Hilfe eines Observer-Patterns sowohl auf Änderungen einzelner Entitäten, wie auch des Entitäten-Pools zu reagieren. Für die Speicherung und das Laden von annotierten Szenen bedient sich der AnnotationsManager der bereits vorgestellten AnnotationLib, die den Zugriff auf die Mongo-Datenbank bereitstellt. Gleichzeitig geschieht

hierdurch die Rückführung des bereits gewonnenen Hintergrundwissens in den Annotierungsprozess.

Weitere Abhängigkeiten ergeben sich, wie in Abbildung 7.5 dargestellt, durch die automatisierte Verarbeitung der geladenen Punktwolken mit Hilfe der Pointcloud Bibliothek (PCL, s. Abschnitt B.1) und dem Zugriff auf die WordNet-Datenbank.

## **7.6 Zusammenfassung**

Für die jeweils umgesetzten Methoden zur Objekt- und Szenenmodellierung wurden verschiedene Anwendungen und Bibliotheken entworfen und implementiert. Um den Anforderungen an Performanz gerecht zu werden und die Ansteuerung der verwendeten Hardware zu ermöglichen wurde C++ als Programmiersprache verwendet. Die Aufteilung der Software in einzelne Module ermöglicht die Wiederverwendung zentraler Komponenten an verschiedenen Stellen und reduziert den Implementierungsaufwand. Der Einsatz sowohl kommerzieller wie auch von Open-Source-Bibliotheken von Drittanbietern vergrößert den Funktionsumfang der Anwendungen.



## **Experimente, Ergebnisse und Bewertung**

Das folgende Kapitel beschreibt die Experimente und Ergebnisse, die die vorangegangenen Überlegungen in die Praxis überführen. Zunächst sollen die Resultate der Objektmodellierung vorgestellt werden, anschließend erfolgt die Auswertung der durch die Szenenmodellierung gewonnenen Daten. In diesem Zusammenhang werden die jeweiligen Ansätze und Methoden hinsichtlich Genauigkeit sowie technischem wie zeitlichem Aufwand untersucht.

### **8.1 Evaluierung der Kalibrierung des Objektmodellierungscenars**

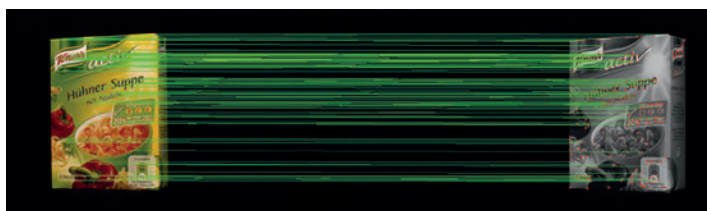
#### **8.1.1 Methodik**

Um die Güte der Kalibrierung von Stereokamerasystem zu 3D-Scanner zu ermitteln, werden die Bilddaten mit den 3D-Daten des Scanners in Be-

ziehung gesetzt. Dafür wird aus den Bildern eine Punktwolke rekonstruiert, die dann mit Hilfe der in der Kalibrierung ermittelten Kameraposen in das Weltkoordinatensystem transformiert wird. Schließlich kann diese Rekonstruktion mit den Scandaten verglichen werden und eventuelle Abweichungen bestimmt werden.

Für die Rekonstruktion der Bilddaten wird das Integrating Vision Toolkit (IVT, s. B.6) verwendet. Wie bereits in Abschnitt 3.1.2 erklärt, basiert diese Rekonstruktion auf der Triangulierung von Bildpunktpaaren in den jeweiligen Bildern. Dazu müssen entsprechende Korrespondenzen im linken und rechten Bild gefunden werden, die dann mit Hilfe der Kamerakalibrierung in einen Tiefenpunkt umgewandelt werden können. Zur Korrespondenzfindung stehen mehrere Möglichkeiten zur Verfügung. Für die Evaluierung werden an dieser Stelle die schon mehrfach angesprochenen SIFT-Merkmale herangezogen. Diese sind einerseits eindeutig, d.h. korrespondierende Merkmale im linken und rechten Bild können also einfach zugeordnet werden. Andererseits ermöglicht eine entsprechende Einstellung der Parameter, dass für die vorliegenden Bilder quasi garantiert werden kann, dass Merkmalspunkte lediglich auf der Objektoberfläche gefunden werden. Abbildung 8.1 zeigt für ein Beispielobjekt und ein Bildpaar wie diese Punkte und deren paarweise Zuordnung aussehen. Die mit Hilfe der Triangulation gewonnene Punktwolke ist zunächst in Kamerakoordinaten gegeben (s. Abb. 5.7) und muss nun in das Weltkoordinatensystem transformiert werden, um den Vergleich mit dem 3D-Scan zu ermöglichen. Die dafür benötigte Transformation ( $T$ ) ist durch die Gesamtkalibrierung gegeben und kann aus der XML-Datei, die jedem Datensatz beiliegt entnommen werden. Da es sich dabei um die Transformation von Welt- zu Kamerakoordinaten handelt, muss diese invertiert werden und kann an-





**Abb. 8.1.** Bestimmung der Bildpunktkorrespondenzen für die 3D-Rekonstruktion auf Basis von SIFT-Merkmalen.

schließlich auf die Punkte der Rekonstruktion ( $P$ ) angewendet werden:

$$\forall p_c \in P : p_w = T^{-1} \cdot p_c \quad (8.1)$$

Für den Vergleich der rekonstruierten Punktwolke mit den Scandaten wird für jeden Punkt der minimale Abstand zur Oberfläche des Scans berechnet. Für konvexe Objekte kann dies z.B. mit Hilfe des Gilbert-Johnson-Keerthi-Verfahrens [Gilbert 88] geschehen. Kann Konvexität nicht vorausgesetzt werden, müssen die einzelnen Teilflächen untersucht werden. Algorithmus 7 skizziert ein mögliches Vorgehen.

### 8.1.2 Ergebnisse

Zur Evaluierung der Punktabweichung wurden die oben beschriebenen Rekonstruktionsschritte und Abstandsberechnungen für zwei Datensätze durchgeführt. Die gewonnenen Abstände werden in einem Histogramm gesammelt, s. Abbildungen 8.2 und 8.3.

Neben dieser quantitativen Analyse kann mit Hilfe der Texturierung des 3D-Modells, basierend auf den Kameraaufnahmen, zusätzlich eine qua-

---

**Algorithmus 7** Algorithmus zur Berechnung der kürzesten Abstände einer Punktmenge zu einer Polygonfläche aus Dreiecken.

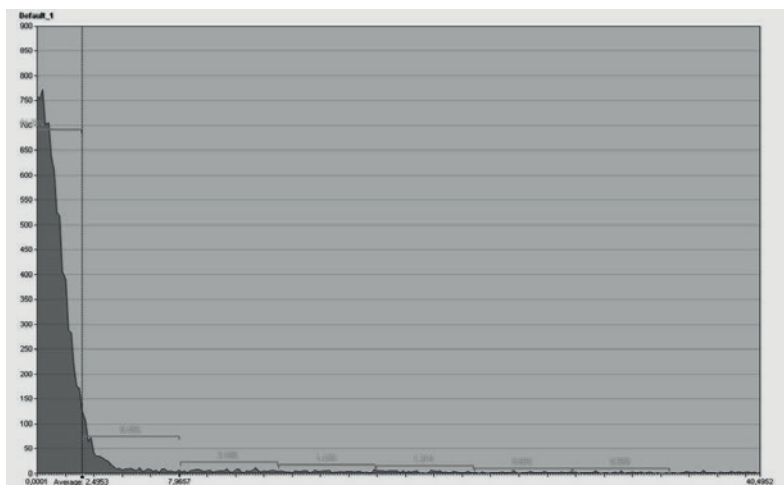
---

```

P: Punktmenge
M: Dreiecksmenge
for  $p \in P$  do
  for  $t \in M$  do
    if  $\mathbf{n}_t$  zeigt in Richtung  $p$  then
      Berechne Abstand  $d$  von  $p$  zu  $t$ 
    end if
    if  $d < d_{min}$  then
       $d_{min} = d$ 
    end if
  end for
end for

```

---



**Abb. 8.2.** Histogramm der Punktabstände der 3D-Rekonstruktion aus Stereobildern und gescanntem Objekt „danish ham“. Horizontale Achse: Abstand in mm, vertikale Achse: Häufigkeit.



daten vorgestellt und analysiert werden. Zunächst werden einige Beispielobjekte vorgestellt, die die Stärken und Schwächen des gewählten Ansatzes illustrieren. Anschließend wird anhand von wissenschaftlichen Arbeiten, die unter Verwendung der Daten aus dem Modellierungcenter entstanden sind, gezeigt, dass diese den gestellten Anforderungen genügen. Schließlich wird die Verbreitung der Objektmodelle in der Servicerobotik-Gemeinschaft mit Hilfe der Zugriffsstatistiken auf die Webdatenbank vorgestellt.

### 8.2.1 Beispielobjekte

Im folgenden Abschnitt soll, anhand von einigen ausgewählten Beispielobjekten, die erzielbaren Ergebnisse mit dem vorgestellten Modellierungsprozess diskutiert werden. Abbildung 8.4 zeigt den Vergleich von Kamerabilddern des Objekts *OrangeMarmelade* mit, auf Basis des erzeugten Objektmodells, computergenerierten Visualisierungen. Gezeigt werden sechs verschiedene Blickwinkel des Objekts, die jeweils linke Spalte („O“) beinhaltet die Kamerabildder, wie sie im Modellierungcenter aufgenommen wurden. Die jeweils rechte Spalte („R“) zeigt die aus nahezu identischer Pose künstlich generierte Visualisierung. Die Rekonstruktionen wurden mit dem Softwarepaket Blender<sup>1</sup> erzeugt. Dafür wurde die Lichtkonfiguration des Modellierungcenters exakt simuliert, wie auch die Parameter der verwendeten Kamera (Brennweite, Sensorgröße) entsprechend eingestellt. Als Modell wurde das Dreiecksnetz mit einer Auflösung von ca. 5.000 Dreiecken verwendet. Es wird deutlich, dass die Rekonstruktion nahezu fotorealistisch ist, an einigen wenigen Stellen können kleine Fehler in der Texturierung erkannt werden.

<sup>1</sup> <http://www.blender.org>

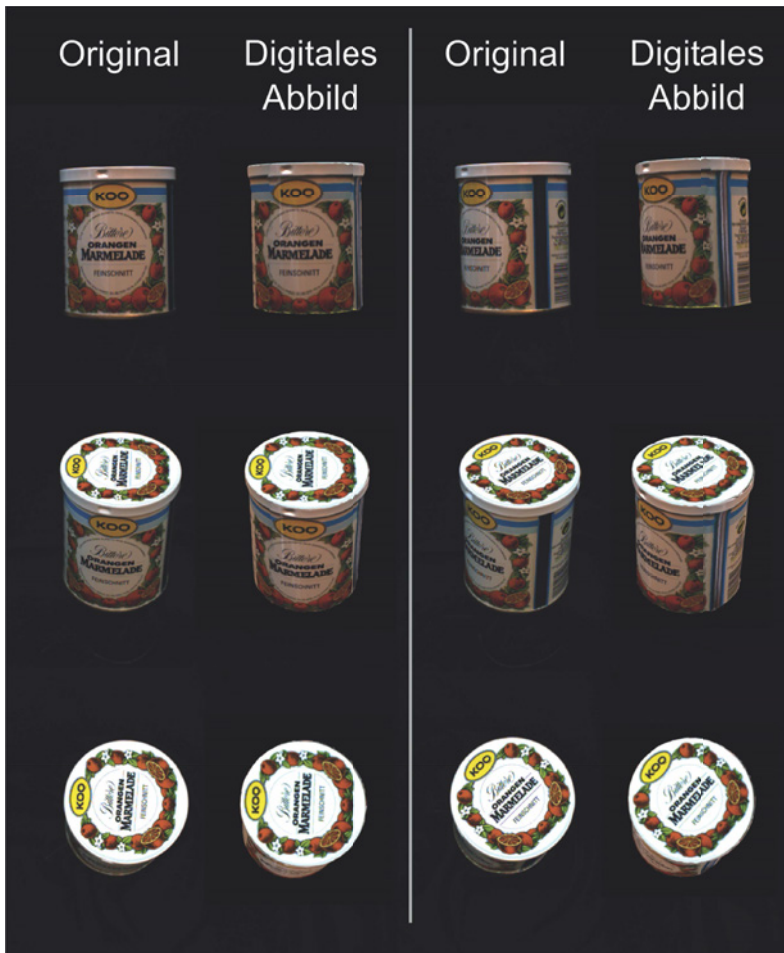
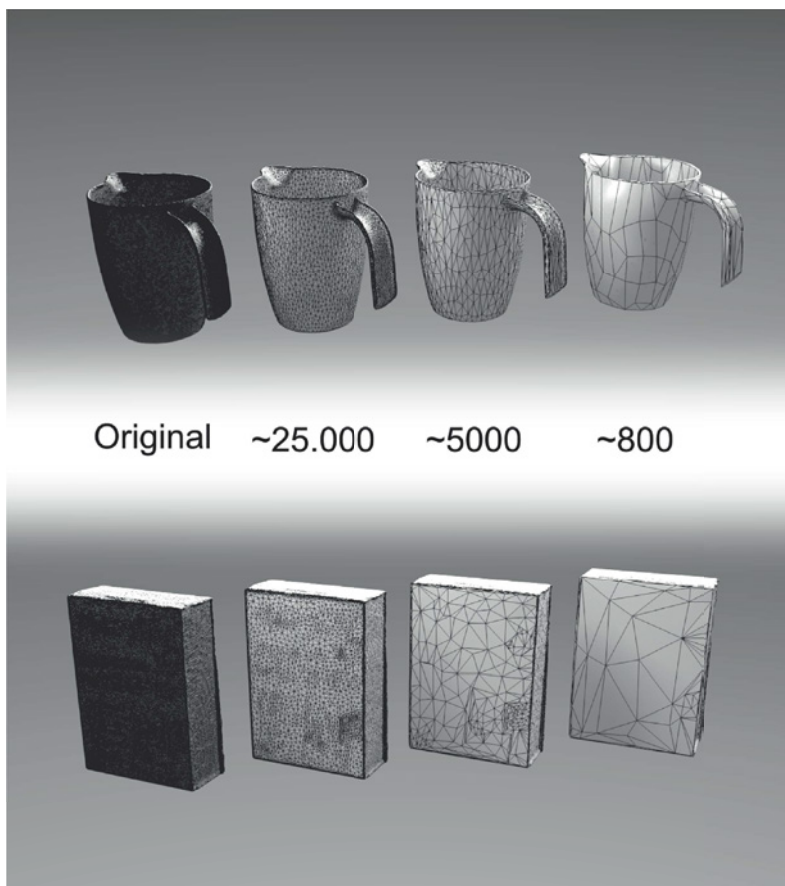


Abb. 8.4. Vergleich zwischen Kameraaufnahme und digitaler Rekonstruktion für sechs verschiedene Blickwinkel.



**Abb. 8.5.** Illustration der verschiedenen Auflösungen für die generierten Dreiecksnetze. Von links nach rechts: original Scanauflösung, erste Reduktion auf ca. 25.000 Dreiecke, zweite Reduktion auf ca. 5.000 Dreiecke und finale Reduktion auf ca. 800 Dreiecke.

Neben der visuellen Güte der Reproduktion, ist die Qualität der Dreiecksnetze für viele Anwendungen ein entscheidender Faktor. Abbildung 8.5 zeigt für zwei Beispielobjekte die vier erzeugten unterschiedlichen Auflösungen. Ganz links im Bild ist die originale Auflösung zu sehen, wie sie nach Fusion und Nachbearbeitung der einzelnen Originalscans vorhanden ist. Die Anzahl an Dreiecken variiert hier je nach Objekt, beträgt in der Regel jedoch zwischen 150.000 und 300.000 Dreiecken. Dieses Ausgangsnetz wird dann in drei Schritten automatisiert reduziert, es entstehen so Netze mit einer Auflösung von ca. 25.000 Dreiecken, ca. 5.000 Dreiecken und ca. 800 Dreiecken. Die für die Reduktion verwendete Software erhält dabei die wichtigen Formmerkmale, wie am Henkel des Messbechers (8.5 oben) gut zu sehen ist.



**Abb. 8.6.** Veranschaulichung der Texturierung aus verschiedenen Ansichten. Farblich codiert sind die einzelnen Ausgangsbilder, die zur Gesamttextur fusioniert wurden, mit ihrer Position auf dem Dreiecksnetz.

Schließlich soll die Qualität der erzeugten Texturen betrachtet werden. Abbildung 8.6 zeigt für das Objekt *OrangeMarmelade*, wie sich die Textur aus den einzelnen Ansichten zusammensetzt und wo diese auf dem Dreiecksnetz zu finden sind. Die vier Seitenansichten sind gelb, rot, blau und grün eingefärbt, die Ansicht von oben ist violett getönt. Die Illustration zeigt deutlich, dass bei der Zuordnung der Ansichten zu Dreiecken eine Fragmentierung fast vollständig verhindert werden konnte, wodurch wenige große und zusammenhängende farbige Flächen entstanden sind. Am oberen Ende des Übergangs von blau zu grün, kann jedoch auch beobachtet werden, dass unter bestimmten Umständen kleinere Fragmentierungen entstehen. Insgesamt ermöglicht der verwendete Ansatz zur Erzeugung der Texturierung jedoch auch eine weitere manuelle Nachbearbeitung der Modelle.

### **8.2.2 Nutzung der generierten Modelle**

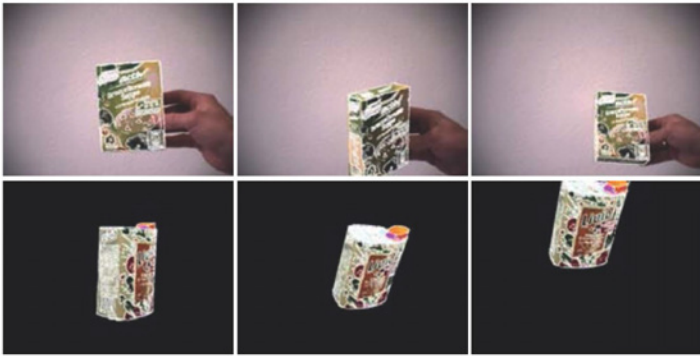
Bereits in Abschnitt 2.1 wurde auf die speziellen Anforderungen an Objektmodelle in der Servicerobotik eingegangen. Der entwickelte Prozess zur Objektmodellierung basiert auf diesen Erkenntnissen und wurde speziell auf die Verwendung der entstehenden Daten für Anwendungen der Servicerobotik, wie Objekterkennung und -lokalisierung oder Greifplanung hin optimiert. Nun gilt es zu validieren, ob die erzeugten Daten tatsächlich den Anforderungen der Arbeiten auf diesem Gebiet genügen. Dazu sollen einige ausgewählte Publikationen vorgestellt werden, die die Datensätze aus dem Modellierungcenter verwenden.



### *Objekterkennung und -lokalisierung*

Im Bereich der Objekterkennung und -lokalisierung haben unterschiedliche Anwender die Daten aus dem Modellierungscenter verwendet. In den Arbeiten [Kuehnle 09], [Grundmann 08], [Grundmann 10b] und [Grundmann 10a] werden jeweils die Stereobilddaten in Verbindung mit den 3D-Daten verwendet um auf Basis von SIFT-Merkmalen eine Objekterkennung und -lokalisierung durchzuführen. Mit Hilfe der kalibrierten Objektansichten berechnen die Autoren eine Punktwolke bestehend aus SIFT-Merkmalen, die auf der Oberfläche des jeweiligen Objektes liegen. Diese Berechnungen können für jedes zu erkennende Objekt offline und a-priori ausgeführt werden. Während der Erkennung wird dann im aktuell vorliegenden Kamerabildpaar eine SIFT-Merkmalsextraktion durchgeführt und die erhaltenen Korrespondenzpunkte werden trianguliert. Nun kann diese SIFT-Merkmalpunktwolke mit den a-priori berechneten Merkmalspunktwolken verglichen werden. Dies ermöglicht gleichzeitig die Klassifizierung wie auch die Bestimmung der Objektpose.

Eine Objekterkennung und -lokalisierung in Kamerabildern kann auch, wie in den Arbeiten von [Wächter 11] und [Azad 11] vorgestellt, alleine auf Basis der im Modellierungscenter generierten 3D-Daten erfolgreich durchgeführt werden. Azad et al. verwenden die 3D-Objektmodelle hierbei als Grundlage für die Erzeugung künstlicher Objektansichten. Das Modell wird mit Hilfe von modernen Computergrafikverfahren abgebildet. Dieses künstlich erzeugte Bild wird dann mit dem aktuellen, realen Kamerabild verglichen um so zu einer Schätzung für die gegenwärtige Objektpose zu gelangen.



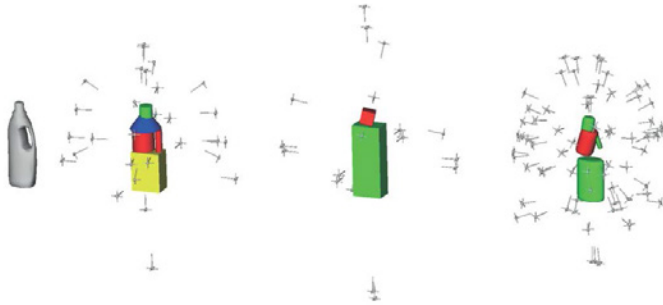
**Abb. 8.7.** Beispiel zur Objekterkennung mit Hilfe der generierten Objektmodelle. Quelle: [Azad 11] ©2011 IEEE.

### *Greifplanung*

Im Bereich der Greifplanung für Serviceroboter verwenden alle Arbeiten, die Daten aus dem Modellierungszentrum benutzen, ausschließlich die generierten 3D-Daten. Diese sind z.B. Grundlage für eine kraftbasierte Simulation, mit deren Hilfe verschiedene Griffe durchgeführt und auf Kraftschluß und Stabilität hin untersucht werden können ([Xue 09a], [Xue 09b]).

Die Arbeit von [Huebner 09] verwendet die 3D-Daten zur Approximation der Objektform durch orientierte, minimale Boundingboxen. Auf Basis dieser Boxen wird dann ein möglichst optimaler Griff, inklusive Anrücktrajektorie ausgewählt bzw. geplant.

[Popovic 11] verwenden ein Bildverarbeitungssystem zur Extraktion spezieller Merkmale (Kanten und Oberflächen), die dann Grundlage für die Greifplanung eines Robotersystems sind. Die 3D-Modelle des Modellierungszentrums wurden in dieser Arbeit für die Validierung innerhalb einer



**Abb. 8.8.** Beispiel zur Greifplanung mit Hilfe der generierten Objektmodelle. Aus dem generierten Modell werden mögliche Richtung zum Anrücken des Greifers erzeugt. Quelle: [Xue 09a] ©2009 IEEE.

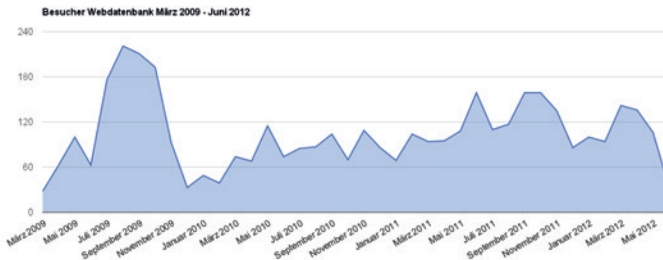
Simulationsumgebung verwendet. Wird der Griff in der Simulation als erfolgreich und damit stabil bewertet, kann er anschließend auf dem Robotersystem durchgeführt werden.

[Ulbrich 11] stellen einen Benchmark für Greifplanungsalgorithmen auf Basis der Objektmodelle vor. Dafür wurde eine nahtlose Verarbeitungskette, ausgehend von der Modellierung der Roboterhand, über die Planung der Griffe, bis hin zur simulierten Ausführung des geplanten Griffs entwickelt. Die 3D-Modelle können innerhalb dieser Verarbeitungskette vollkommen automatisch aus der Datenbank ausgewählt und integriert werden.

### 8.2.3 Verbreitung durch Webdatenbank

Im vorherigen Abschnitt ist bereits angedeutet, dass die generierten Objektmodelle von einer breit gestreuten Nutzergruppe zu unterschiedlichen

Zwecken verwendet werden. An dieser Stelle sollen einige Zahlen weiteren Aufschluss über die Verbreitung der Daten mit Hilfe der Webdatenbank geben. Die Daten für diese Analyse wurden mit Hilfe der Software *Google Analytics*<sup>2</sup> erstellt.

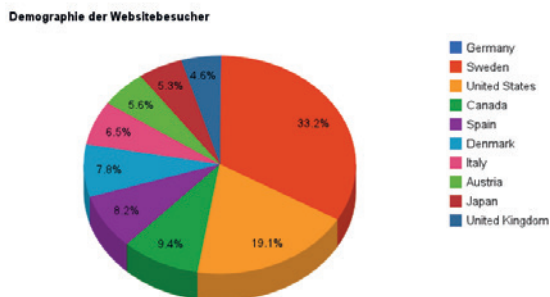


**Abb. 8.9.** Zeitlicher Verlauf der eindeutigen Besucherzahlen auf der Webdatenbank im Zeitraum 11.03.2009 - 10.06.2012.

In Abbildung 8.9 ist die Anzahl eindeutiger Besucher der Website über den Zeitraum vom 11. März 2009 (Tag der Inbetriebnahme) bis zum 10.06.2012. Insgesamt haben in diesem Zeitraum 4147 Personen die Website besucht. Von dieser Zahl gelangten 23,80% über eine Suchmaschine, 19,32% über eine verweisende Website und 56,88% auf direktem Weg zur Webdatenbank.

Abbildung 8.10 zeigt die örtliche Verteilung der Besucher der Webdatenbank. Die meisten Besucher finden sich damit innerhalb von Deutschland, jedoch auch Besucher aus Schweden und den USA sind häufig vertreten.

<sup>2</sup> <http://www.google.com/intl/de/analytics/>



**Abb. 8.10.** Standort der Besucher der Webdatenbank im Zeitraum 11.03.2009 - 10.06.2012.

Im gleichen Zeitraum wurden 3D- und Bilddatensätze insgesamt 5580 Mal heruntergeladen.

## 8.3 Ergebnisse Szenenmodellierung auf Basis von räumlichen Relationen

### 8.3.1 Evaluierung der Relation „Ist neben“ mittels Benutzerstudie

Im Gegensatz zur Relation „Ist auf“, die sich auf Grund physikalischer Gegebenheiten wie dem Massenschwerpunkt, Reibungskräften und relativen Positionen mathematisch gut nachbilden lässt, ist die Relation „Ist neben“ wesentlich subjektiver und kontextabhängiger. Die in Abschnitt 6.6 vorgeschlagene Modellierung basiert dabei auf den relativen Größen der beteiligten Objekte und deren relativer Posen. Da die Modellierung dieser Relation auch zur Kommunikation mit potentiellen menschlichen Benutzern

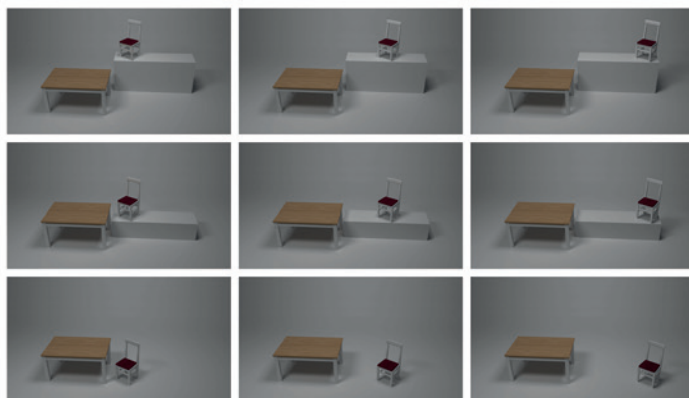
verwendet werden soll, gilt es zu evaluieren, wie sich die Ergebnisse des Modells mit der menschlichen Einschätzung decken. Zu diesem Zweck wurden zwei verschiedene Beispielszenarien künstlich modelliert. Szene A beinhaltet zwei ähnlich große Objekte, eine Mikrowelle und ein Radio, während Szene B zwei verschieden große Objekte, einen Tisch und einen Stuhl, enthält. Die absolute Größe der Objekte in Szene A und B unterscheidet sich dabei ebenfalls erheblich, um eine mögliche Korrelation der Relation mit diesem Faktor aufzuzeigen. Die beiden Objekte in Szene A und B wurden jeweils in neun unterschiedlichen relativen Posen angeordnet, vgl. Abb. 8.11 und Abb. 8.12.



**Abb. 8.11.** Anordnung der Testobjekte in Szene A.

Die Anordnung der Objekte in beiden Szenen ist ähnlich, wobei die 9 gewählten Posen repräsentativ für eine Mehrheit an möglichen Objektkonstellationen gewählt wurden. Die Objekte befinden sich jeweils in drei unterschiedlichen horizontalen Distanzen zueinander und dabei jeweils auf

drei verschiedenen Höhen. Diese Anordnung resultiert aus der Modellierung der Relation über die beiden Distanzterme in horizontaler und vertikaler Richtung.



**Abb. 8.12.** Anordnung der Testobjekte in Szene B.

Diese 18 Beispielszenen wurden dann mit Hilfe einer webbasierten Umfrage von Testpersonen daraufhin bewertet, wie sehr ihrer Ansicht nach die Relation „ist neben“ für die jeweils gezeigten Objekte erfüllt ist. Abbildung 8.13 zeigt ein Bildschirmfoto der Umfragewebsite. Die Teilnehmer wählten dabei für die Relation Werte zwischen 0 und 1 in Inkrementen von 0,1.

An der Befragung nahmen insgesamt 33 Personen teil. In Abbildung 8.14 sind die Ergebnisse dieser Befragung im Vergleich zu den für die Test-szenarien mit Hilfe des Modells unter Verwendung der Standardparameter

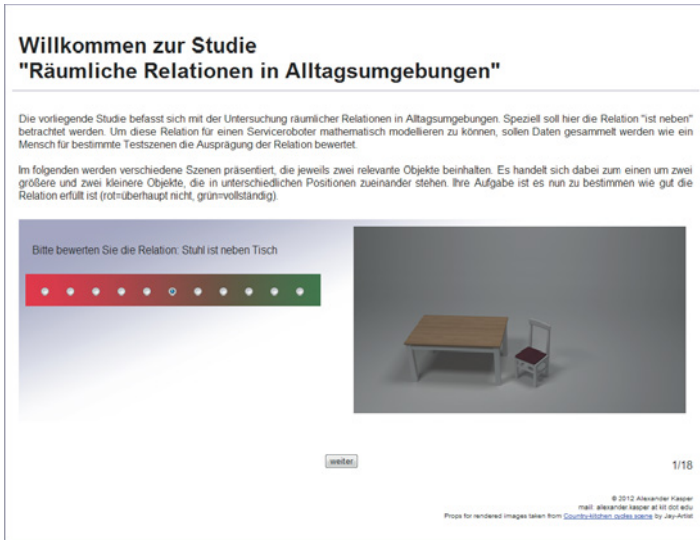


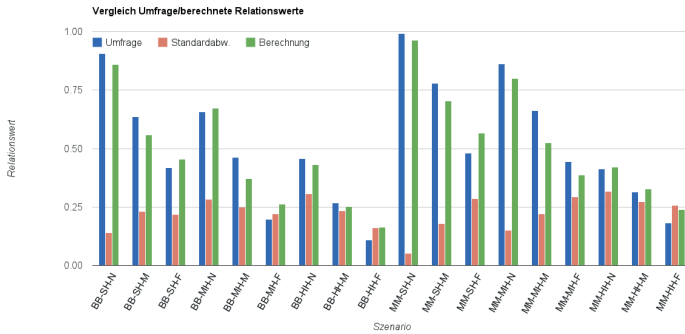
Abb. 8.13. Bildschirmfoto der webbasierten Umfrage.

berechneten Relationswerte dargestellt. Die im Diagramm verwendeten Kodierungen für die Szenarien schlüsseln sich wie folgt auf:

- Größe der Objekte: BB = big/big (Stuhl und Tisch), MM = medium/-medium (Mikrowelle und Radio)
- Vertikaler Abstand: SH = same height (gleiche Höhe), MH = medium height (mittlere Verschiebung), HH = huge height (große Verschiebung)
- Horizontaler Abstand: N = near (nah), M = medium (mittel), F = far (weit)



Die jeweiligen Bewertungen für die einzelnen Szenarien wurden gemittelt und mit den Modellwerten verglichen, wie in Abb. 8.14 zu sehen. Der Gewichtungsparemeter  $w$  für Gleichung 6.7 wurde an dieser Stelle empirisch ermittelt um die Umfrageergebnisse möglichst gut nachzuempfinden und ergibt sich schließlich zu 0,6. Es zeigt sich, dass sich das Modell in diesem Fall sehr gut mit den Einschätzungen der Umfrageteilnehmer deckt. Allerdings muss beachtet werden, dass die Umfrageergebnisse teilweise streuen, der Durchschnittswert in diesen Fällen lediglich als Kompromiss betrachtet werden kann. Gerade aber die Abhängigkeit der Relationsberechnung von den jeweiligen Objektgrößen erweist sich hier aber als günstiger Ansatz.



**Abb. 8.14.** Vergleich der Relationswerte zwischen Umfrage und nach Modell berechnet. Zusätzlich angegeben ist die Standardabweichung der Umfrageergebnisse.

### 8.3.2 Datensatz

Im folgenden Abschnitt soll der für die Evaluierung der Szenenmodellierung verwendete Datensatz kurz vorgestellt werden. Da die vorliegende Arbeit im Rahmen des Sonderforschungsbereichs 588 „Lernende und kooperierende multimodale Roboter“<sup>3</sup> durchgeführt wurde und hier die Küche als Hauptszenario festgelegt wurde, gliedern sich die hier erzeugten Datensätze in dieses Umfeld ein. Konkret wird als Szenenklasse der „Frühstückstisch“ gewählt. Die Beschränkung auf Objekte, die sich auf einem Tisch befinden, erleichtert die sensorielle Erfassung und die Annotierung, wobei trotzdem eine ausreichende Varianz der vorkommenden Objekte und deren Positionen gewährleistet ist. Für den Trainingsdatensatz wurden 23 verschiedene Szenen aufgebaut, mit Hilfe des Kinect-Sensors erfasst und anschließend in der speziell entwickelten Software annotiert. Insgesamt enthalten die 23 Szenen 23 verschiedene Objektklassen. Die Abbildungen 8.15 und 8.16 zeigen Ansichten aus zwei der 33 Szenen.

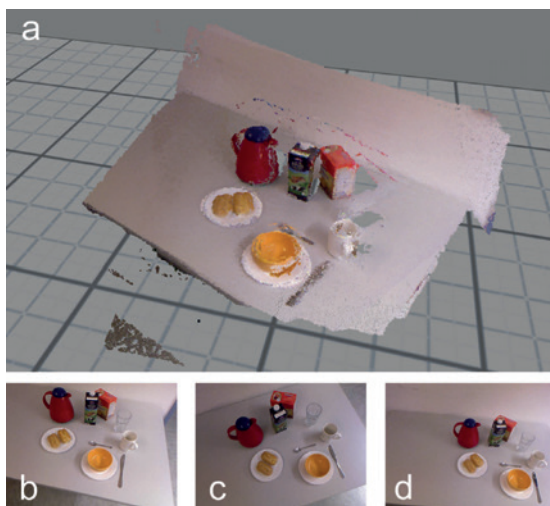
Die Aufnahme einer solchen Szene mit Hilfe des Kinect-Sensors dauert durchschnittlich etwa 5 Minuten. Die Annotierung eines solchen Datensatzes benötigt in etwa 15 Minuten. Innerhalb einer Stunde können demzufolge 2-3 Datensätze digitalisiert werden.

### 8.3.3 Auswertung

Die folgende Auswertung des annotierten Trainingsdatensatzes soll die Struktur des gewonnenen Hintergrundwissens aufzeigen und an Hand von einigen Beispielen die zu erwartenden Ergebnisse diskutieren.

---

<sup>3</sup> <http://www.sfb588.uni-karlsruhe.de>



**Abb. 8.15.** Szene aus dem Trainingsdatensatz der Szenenmodellierung am Beispiel „Frühstückstisch“. a) 3D-Punktwolke b-d) 2D-Bilddaten.

Ein zentraler Aspekt der Szenenmodellierung ist die Frage, welche Objekte mit welcher Wahrscheinlichkeit in der Szene zu erwarten sind. Aus den Annotierungsdaten können diese Wahrscheinlichkeiten leicht berechnet werden. Hierzu wird für jede Objektklasse die Anzahl der gefundenen Instanzen gezählt. Abbildung 8.17 zeigt die konkrete Verteilung für den vorliegenden Datensatz. Insgesamt wurden 466 Objekte aus 30 verschiedenen Objektklassen beobachtet und annotiert.

Neben der Auftretenswahrscheinlichkeit der jeweiligen Objektklassen ist insbesondere die Verteilung der Werte für die in Kapitel 6 eingeführten Relationen innerhalb der Trainingsdaten interessant. Für die Berechnung dieser Verteilungen werden, innerhalb jeder annotierten Szene, alle möglichen Objektpaarungen gebildet und die Relationen für diese Paarungen



**Abb. 8.16.** Szene aus dem Trainingsdatensatz der Szenenmodellierung am Beispiel „Frühstückstisch“. a) 3D-Punktwolke b-d) 2D-Bilddaten.

berechnet. Anschließend werden diese Werte diskretisiert und in ein Histogramm eingetragen. Die Diskretisierung ist notwendig, da die Daten den Raum der möglichen Relationen nur stichprobenartig abdecken und so keine kontinuierlichen Verteilungen erzeugt werden können. In Abbildung 8.18 und 8.19 sind die Verteilungen der Werte für die Relationen „ist auf“ und „ist neben“ für zwei verschiedene Objektklassenpaarungen dargestellt.

In beiden Paarungen ist die Objektklasse „plate“ als sekundäre Klasse enthalten. Zwischen den Verteilungen lassen sich jedoch deutliche Unterschiede feststellen. Es können hier bereits einzelne lokale Maxima beobachtet werden, die der menschlichen Erfahrung entsprechen. So geht eine erhöhte Wahrscheinlichkeit für eine starke Ausprägung der Relation

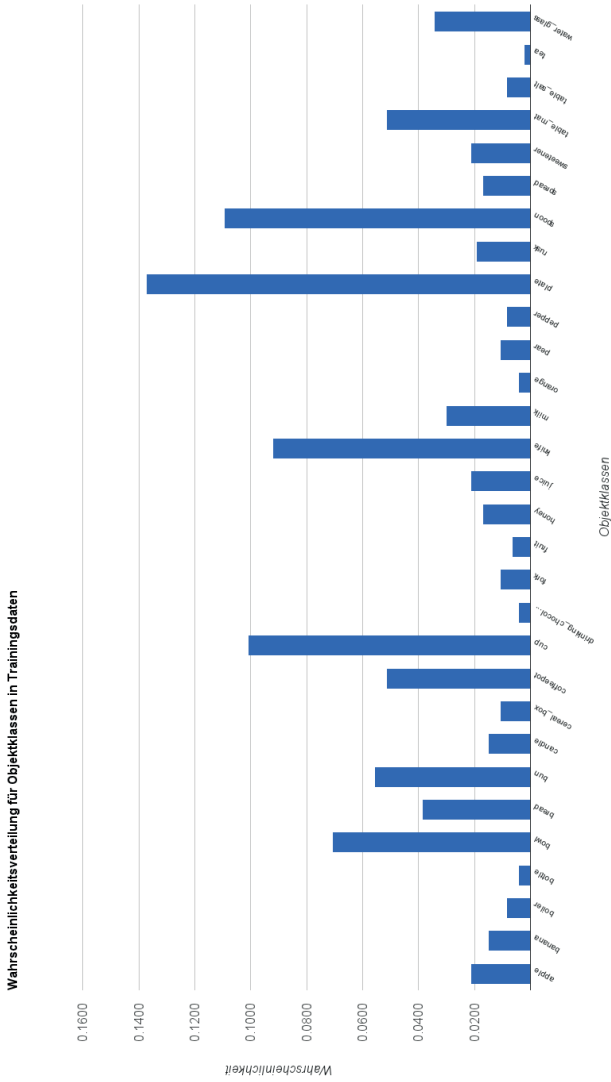
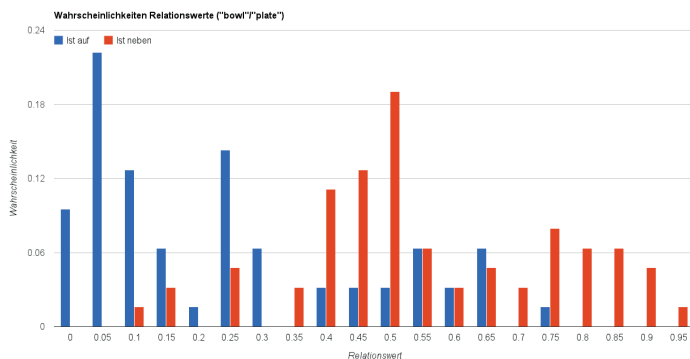
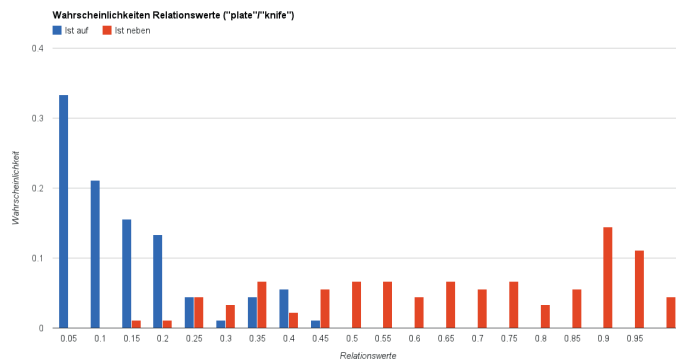


Abb. 8.17. Auftretenswahrscheinlichkeiten der verschiedenen Objekt-Klassen in den Trainingsdaten. Von links: „apple“, „banana“, „beaker“, „bottle“, „bread“, „bun“, „candle“, „cereal\_box“, „coffee\_pot“, „cup“, „drinking\_chocolate“, „fork“, „fruit“, „honey“, „juice“, „knife“, „milk“, „orange“, „pear“, „pepper“, „plate“, „rusk“, „spoon“, „spread“, „sweetener“, „table\_mat“, „table\_salt“, „tea“, „water\_glass“.



**Abb. 8.18.** Wahrscheinlichkeiten für die verschiedenen Relationswerte für die Paarung der Objektklassen „bowl“ und „plate“.



**Abb. 8.19.** Wahrscheinlichkeiten für die verschiedenen Relationswerte für die Paarung der Objektklassen „plate“ und „knife“.

„ist neben“ zwischen Objekten der Klasse „bowl“ und der Klasse „plate“ einher mit der Erfahrung, dass zum Frühstück häufig eine Schale für Müsli o.ä. neben einem Teller vorgefunden werden kann. Die Verteilung der Relationswerte der Klassen „knife“ und „plate“ lässt sich ebenso mit der Beobachtung in Einklang bringen, dass sich das Messer häufiger neben dem Teller befindet als darauf.

Natürlich lässt die Varianz der Positionen der Objekte in einer Alltagsszene quasi keine Rückschlüsse über allgemein gültige Beziehungen zwischen einzelnen Objekten zu. So kann z.B. nie davon ausgegangen werden, dass sich eine Tasse *immer* neben einem Teller und *nie* darauf befindet. Die gewonnenen Verteilungen eignen sich jedoch, um in konkreten Fällen bspw. Plausibilitätsprüfungen durchzuführen oder Kandidaten für mögliche Objektklassenzuordnungen zu ermitteln.

Eine mögliche Anwendung der erzeugten Daten, die bereits in Abschnitt 6.8 angesprochen wurde, ist die Berechnung der Wahrscheinlichkeit für die Zugehörigkeit eines unbekanntes Objekts zu einer Objektklasse, bei gegebenem Bezugsobjekt und zugehörigen Relationswerten. Diese Fragestellung könnte z.B. im Rahmen einer Objekterkennung auftreten, bei der bereits ein oder mehrere Objekte in der Szene identifiziert wurden und nun für lokalisierte, aber nicht erkannte Objekte, mögliche Hypothesen erzeugt werden sollen.

In den Abbildungen 8.20 und 8.21 sind die so berechneten Wahrscheinlichkeiten für die bekannten Objektklassen zu den gegebenen Werten dargestellt. In beiden Diagrammen sind jeweils drei Verteilungen zu sehen, was der Tatsache geschuldet ist, dass die Relation „ist auf“ nicht kommutativ ist. Sei  $U$  das unbekanntes Objekt und  $B$  eine mögliche Objektklasse, gilt es daher nicht nur  $IsOn(U, B)$  ( $U$  befindet sich auf  $B$ ), sondern eben-

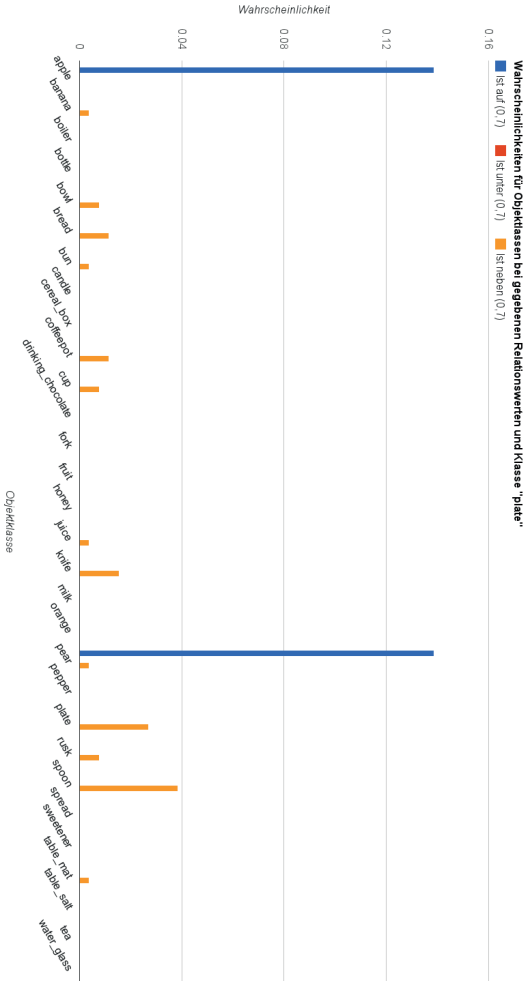
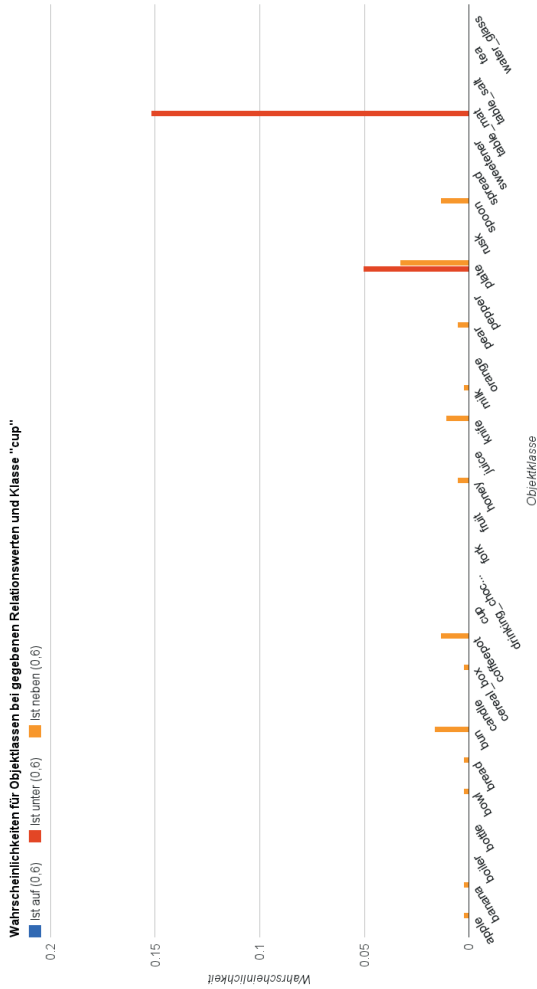


Abb. 8.20. Gegeben ist ein Objekt der Klasse „plate“ und die Relationswerte zu einem Objekt unbekannter Klasse. Berechnet wurden die Wahrscheinlichkeiten für die bekannten Objektklassen.





**Abb. 8.21.** Gegeben ist ein Objekt der Klasse „cup“, und die Relationswerte zu einem Objekt unbekannter Klasse. Berechnet wurden die Wahrscheinlichkeiten für die bekannten Objektklassen.

falls  $IsOn(B, U)$  ( $U$  befindet sich unter  $B$ ) zu betrachten. Die Ergebnisse zeigen deutliche Maxima für bestimmte Objektklassen, die sich wiederum größtenteils mit dem menschlichen Erfahrungsschatz decken.

Neben den Verteilungen der Relationswerte zwischen den Objektklassen und den daraus resultierenden Wahrscheinlichkeiten wurden in Abschnitt 6.7 zusätzlich die Volumina der Objekte betrachtet. Die aus den Trainingsdaten gewonnenen Verteilungen für dieses Objektattribut sind auszugsweise in Abb. 8.22 veranschaulicht. Hierbei ist zu erkennen, dass (zumindest im vorliegenden Datensatz) das Volumen für Vertreter einer Objektklasse nur geringfügig variiert.

Nachdem die Verteilung der Volumenwerte für die Objektinstanzen der Trainingsdaten errechnet wurde, kann mit ihrer Hilfe nach Gleichung 6.17 die Wahrscheinlichkeit für die Zugehörigkeit eines Objekts zu einer Klasse bei gegebenem Volumen bestimmt werden. Abbildung 8.23 zeigt die Ergebnisse dieser Berechnung bei zwei vorgegebenen Volumina.

Insgesamt wird an dieser Stelle jedoch auch deutlich, dass diese Form der Szenenmodellierung (wie alle wahrscheinlichkeitsbasierten Methoden) sehr stark von einem möglichst umfangreichen Satz an Trainingsdaten abhängig ist.

### **8.3.4 Schätzung der Klassenzugehörigkeit**

Nachdem das Hintergrundwissen probabilistisch und statistisch ausgewertet wurde, wird evaluiert wie sich dieses Wissen zur Schätzung der Klassenzugehörigkeit eines Objekts eignet. Zu diesem Zweck wird eine Testszene betrachtet, die einige Vertreter der bereits bekannten Objektklassen

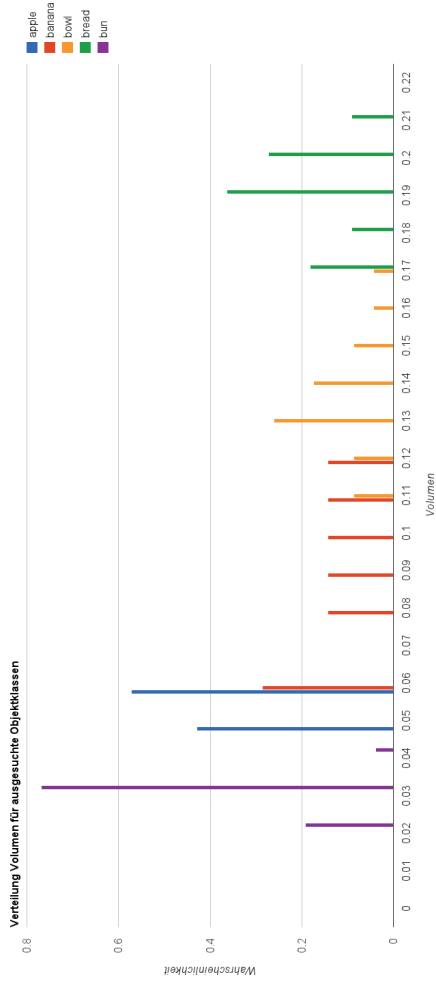


Abb. 8.22. Wahrscheinlichkeiten für das Volumen von gegebenen Objektklassen („apple“, „banana“, „bowl“, „bread“, „bun“).

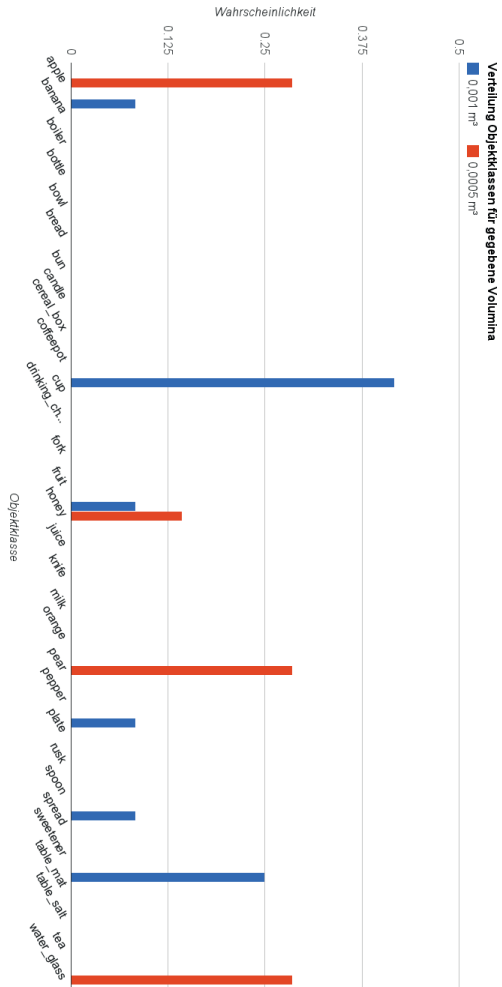
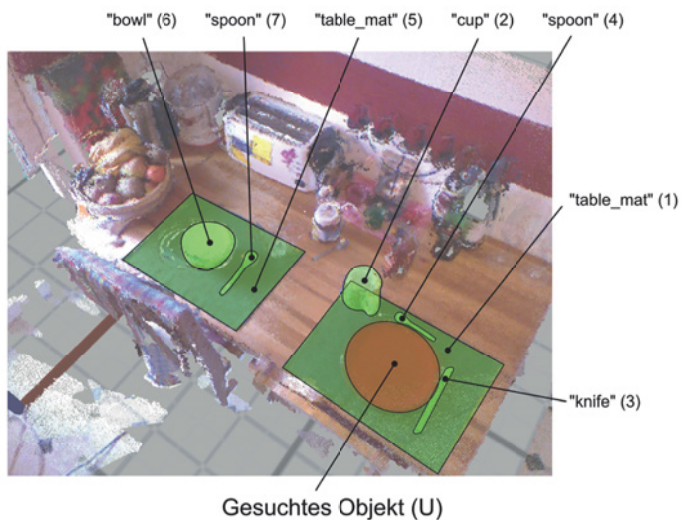


Abb. 8.23. Wahrscheinlichkeiten für die Zugehörigkeit eines Objekts zu den verschiedenen Objektklassen bei gegebenen Volumina.

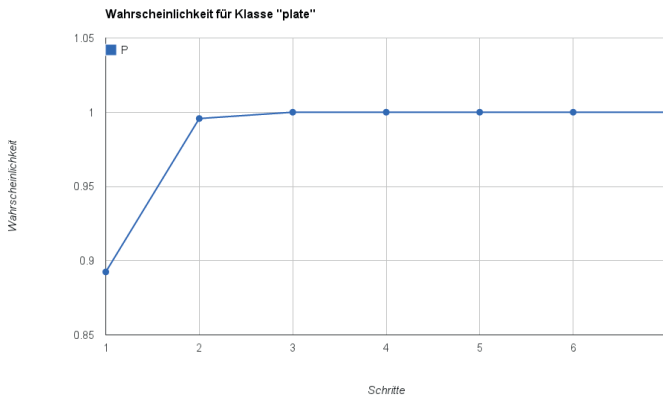
enthält. Für ein annotiertes Objekt (d.h. Größe und Position sind bekannt) soll nun mit Hilfe des Hintergrundwissens die Klassenzugehörigkeit geschätzt werden. Dafür wird zunächst jeweils nur ein Objekt mit bekannter Klassenzugehörigkeit vorgegeben. Anschließend werden unterschiedliche Kombinationen von je zwei vorgegeben Objekten mit bekannter Klassenzugehörigkeit untersucht. Tabelle 8.1 zeigt jeweils die drei Schätzungen für die Klassenzugehörigkeit des gesuchten Objekts U für die einzelnen Fälle. Aufgeführt hierbei sind die normalisierten Wahrscheinlichkeiten für die einzelnen Relationen und das Volumen, sowie die Gesamtwahrscheinlichkeit nach Gleichung 6.18.



**Abb. 8.24.** Testszene zur Evaluierung der Schätzung einer Klassenzugehörigkeit für ein gesuchtes Objekt.

Die Testszene mit dem gesuchten Objekt und den Vorgabeobjekten ist in Abbildung 8.24 dargestellt. Gesucht ist in diesem Fall der Teller rechts unten (U). Es werden nun folgende Objekte in der Szene vorgegeben: Untersetzer („table\_mat“ (1)), Tasse („cup“ (2)), Messer („knife“ (3)), Löffel („spoon“ (4)), Untersetzer („table\_mat“ (5)), Schale („bowl“ (6)), Löffel („spoon“ (7)).

Der Verlauf der Wahrscheinlichkeit für die Zugehörigkeit des untersuchten Objekts U zur Klasse „plate“ in Abhängigkeit der umgebenden Objekte zeit Abb. 8.25.



**Abb. 8.25.** Verlauf der Wahrscheinlichkeit für die Zugehörigkeit zur Klasse „plate“ des gesuchten Objekts in Abhängigkeit der umgebenden Objekte (vgl. Abb. 8.24)

**Tabelle 8.1.** Ergebnisse für die Schätzung der Klassenzugehörigkeit für ein Anfrageobjekt.

Schritt	Geschätzte Klasse	Wahrscheinlichkeit			
		$P$	$P_{Vol}$	$P_{On}$	$P_{Next}$
(1)	„plate“	0,991612	0,470367	0,108736	0,527247
	„spoon“	0,004206	0,008231	0,112763	0,140599
	„knife“	0,002776	0,006585	0,1329	0,105449
(2)	„plate“	0,598424	0,470367	0,043478	0,084723
	„bread“	0,361714	0,235183	0,043478	0,102423
	„bun“	0,007261	0,006114	0,043478	0,079076
(3)	„plate“	0,989052	0,470367	0,091821	0,346971
	„cup“	0,004386	0,007055	0,037953	0,248417
	„banana“	0,002729	0,001646	0,151812	0,165478
(4)	„plate“	0,517336	0,470367	0,049961	0,13546
	„rusk“	0,252078	0,235183	0,065761	0,10029
	„bread“	0,217353	0,235183	0,061063	0,093128
(5)	„plate“	0,947286	0,470367	0,108736	0,083849
	„bowl“	0,014261	0,005409	0,072491	0,100619
	„spoon“	0,010047	0,008231	0,112763	0,055899
(6)	„bread“	0,340177	0,235183	0,045053	0,053029
	„plate“	0,317498	0,470367	0,045053	0,024746
	„rusk“	0,283481	0,235183	0,039421	0,044191
(7)	„bread“	0,492482	0,235183	0,039357	0,111716
	„plate“	0,453526	0,470367	0,041345	0,060935
	„bun“	0,009959	0,006114	0,051018	0,086889
(6)+(7)	„plate“	0,785736	0,666078	0,0430013	0,0570819
	„bread“	0,213311	0,166519	0,0409339	0,0523251
	„bun“	0,000233	0,000112	0,0530625	0,084786
(3)+(4)	„plate“	0,999748	0,666078	0,1208	0,186325
	„spoon“	0,000155	0,000203	0,0149502	0,238458
	„cup“	0,000004	0,000149	0,0376927	0,0681309
(1)+(5)	„plate“	0,99993	0,666078	0,105178	0,588669
	„spoon“	0,000004	0,000203	0,113113	0,104652
	„knife“	0,000001	0,000131	0,157117	0,0588669

## 8.4 Fazit

### 8.4.1 Objektmodelle

Um die erzielten Ergebnisse in diesem Bereich zu beurteilen, wurden verschiedene Resultate vorgestellt. Zur Diskussion dieser Ergebnisse im Hinblick auf die ursprüngliche Fragestellung der Objektmodellierung sollen drei Aspekte betrachtet werden:

#### *Qualität der Modelle*

Um qualitativ hochwertige, digitale Reproduktionen der geforderten Objekte zu erzeugen ist zunächst eine möglichst genaue dreidimensionale Vermessung notwendig. Der im Aufbau verwendete Konica-Minolta Vi-900 3D-Scanner vermag diesen Anforderungen mit Hilfe seiner Spezifikationen, u.a. eine Tiefenaufösung im Submillimeterbereich, gerecht zu werden. Die Weiterverarbeitung der gewonnenen Tiefendaten durch die kommerzielle Softwarebibliothek innerhalb der speziell entwickelten Anwendung gewährleistet die Erzeugung von resultierenden Dreiecksnetzen mit nicht-mannigfaltiger Topologie und hoher Punktdichte. Schließlich ermöglicht dies auch die für viele Anwendungen notwendige Reduktion der Auflösung bei Erhalt der Objektkontur. Schwierigkeiten bei der Erzeugung der Tiefendaten ergeben sich durch spezifische Materialeigenschaften wie etwa hohe Reflektivität oder Transparenz. Diese können zwar durch Verwendung eines Pulversprays abgemildert werden, erschweren die Modellierung jedoch. Bei Objekten mit vielen Vertiefungen und Löchern ergeben sich wegen des Messprinzips des Scanners Artefakte und Unstetigkeiten in der Geometrie, die in schweren Fällen nicht reparabel oder vermeidbar



sind. Insgesamt kann jedoch festgestellt werden, dass sich die verwendete Sensorik und Software für die Vermessung der großen Mehrheit der verarbeiteten Objekte bewährt hat.

Um aus den gewonnenen Kamerabildern eine hochwertige Textur zu erzeugen, müssen die jeweiligen Kamerapositionen relativ zum Objekt möglichst genau bekannt sein. Das vorgestellte Kalibrierungsverfahren stellt diese in der notwendigen Genauigkeit zur Verfügung, wie durch die quantitative und qualitative Analyse der erzeugten Texturen gezeigt werden kann. Schwachstellen des Texturierungsprozesses sind die Nahtstellen zwischen den einzelnen Objektansichten in der finalen Textur, sowie eine Anfälligkeit gegenüber Glanzlichtern bei stark reflektierenden Materialien.

#### *Geschwindigkeit und Robustheit des Aufbaus*

Neben der Qualität der generierten Modelldaten ist der dafür benötigte Aufwand eine wichtige Randbedingung der Objektmodellierung. Hierfür müssen die Abläufe möglichst weit automatisiert und optimiert sein. Die Digitalisierung von 100 Objekten innerhalb von 4 Wochen hat gezeigt, dass der Aufbau des Modellierungscenars äußerst robust und zuverlässig ist. Die Erzeugung eines vollständigen Datensatzes benötigt nur wenig Zeit und verläuft in Teilen voll automatisch. Einzig die Akquisition der Tiefendaten bedarf Interaktion durch den Benutzer um die geforderte Qualität zu erreichen. Während der gesamten Arbeit kam es zu keinen Hardwareausfällen.

*Verbreitung der bereitgestellten Daten*

Die Erzeugung von spezialisierten Modelldaten für die Servicerobotik macht natürlich nur Sinn, wenn die Daten auch nutzbar sind und entsprechende Verbreitung finden. Die gegebene Übersicht über aktuelle Publikationen, die unter Verwendung der erzeugten Daten entstanden sind, zeigt, dass dieses Ziel erreicht werden konnte. Auch die Zugriffszahlen auf die Datenbank über das Webinterface belegen dies. Von Nutzerseite wurde jedoch angemerkt, dass die Auswahl der bereitgestellten Objekte zu homogen ist und mehr Variabilität in Größe und Form gewünscht wird.

**8.4.2 Szenenmodellierung**

Das Ziel im Rahmen der hier vorgestellten Szenenmodellierung ist die Generierung und Repräsentation von Wissen über Objekte, welches aus dem räumlichen Kontext des betreffenden Objekts entsteht. Zu diesem Zweck sollen reale Alltagsszenen mit Hilfe einer Digitalisierung und Annotierung untersucht werden. Das verwendete Verfahren basierend auf 3D-Punktwolken, die mit einem Kinect-Sensor erzeugt werden, erweist sich hierfür als äußerst geeignet, da der Aufbau mobil und einfach zu verwenden ist. Die resultierenden Punktwolken und Bilder stellen eine gute Grundlage für eine semi-automatische Annotierung durch einen menschlichen Benutzer dar. Die speziell entwickelte Applikation für die Annotierung ermöglicht durch die Integration automatischer Methoden zur Ausrichtung und Clusterextraktion eine schnelle Verarbeitung der Tiefendaten. Schwierigkeiten ergeben sich hier lediglich bei Objekten, die vom Tiefensensor nicht erfasst werden können, wie Gläser oder Objekte mit

spiegelnden Oberflächen. Die zusätzliche Anzeige der 2D-Kamerabilder zusammen mit den entstehenden Artefakten ermöglicht jedoch in den allermeisten Fällen eine Annotierung.

Zur Repräsentation und Auswertung der Interobjektbeziehungen stellen die beschriebenen Relationen „ist auf“ und „ist neben“ ein geeignetes Mittel dar. Beide lassen sich allein mit Hilfe einer orientierten Boundingbox und den relativen Objektposen berechnen. Für die aus der Literatur übernommene „ist auf“-Relation wurde dort bereits gezeigt, dass diese für die Perzeption eines Robotersystems geeignet ist. Die neu entwickelte „ist neben“-Relation bildet die Einschätzung menschlicher Benutzer, zumindest für das Haushaltsszenario, sehr gut nach. Aus diesen Relationen lässt sich somit eine probabilistische Szenenbeschreibung erzeugen, die die in der Trainingsphase beobachteten Anordnungen der Objekte nachbildet. Die in den Abbildungen 8.26 und 8.27 dargestellten Graphen visualisieren diese Beschreibung. Dabei korrespondiert eine Ballung von Klassen (bspw. „bread“, „spoon“, etc.) mit dem gehäuften gemeinsamen Auftreten von zu diesen Klassen gehörenden Objekten innerhalb einer Szene. Die Kanten des Graphen repräsentieren die Relationswahrscheinlichkeiten, je dicker ein Pfeil eingezeichnet ist, desto wahrscheinlicher besteht die gegebene Relation zwischen den Objektklassen. Hierfür wurde angenommen, dass für Relationswerte größer 0,5 die Relation erfüllt ist und die entsprechenden Wahrscheinlichkeiten sind im Graph aggregiert.

Schließlich kann mittels einer probabilistischen Modellierung der aus den Trainingsdaten gewonnenen Relationswerte, ein einfach zu verwendendes Hintergrundwissen erzeugt werden. Mit dessen Hilfe kann die Klassenzugehörigkeit eines unbekanntes Objekts anhand der umgebenden Objekte geschätzt werden, bzw. Plausibilitätsprüfungen durchgeführt werden. We-

nigstens innerhalb des untersuchten Szenarios können die gleichen Relationen zwischen typischen Objekten beobachtet und nachgewiesen werden, wie sie dem menschlichen Erfahrungsschatz entsprechen. Eine Aussage über die Performanz der beschriebenen Methodik in einer anderen Art von Szene, wie etwa einem Außenbereich, kann nicht getroffen werden.

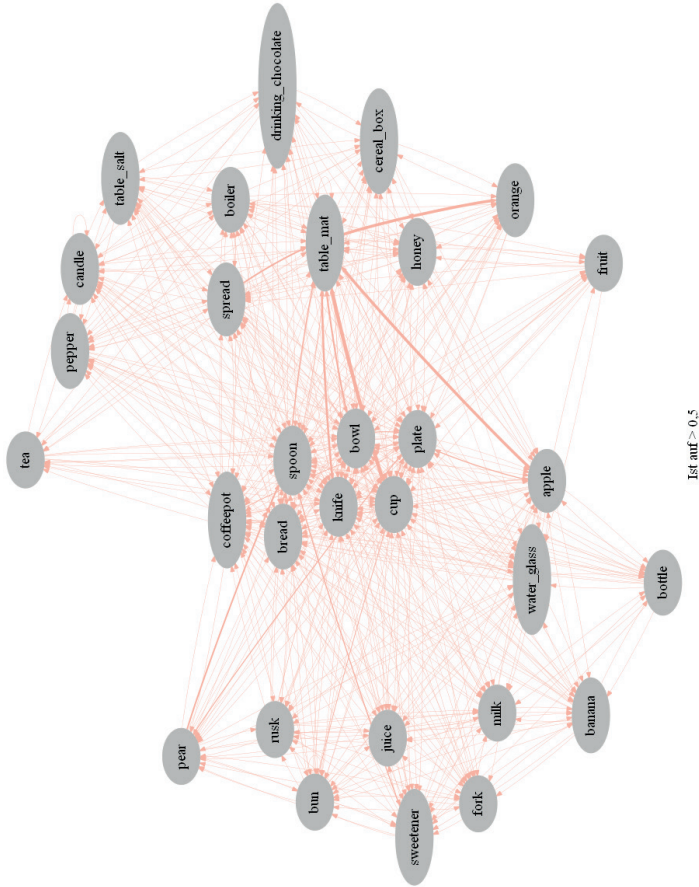


Abb. 8.26. Szenenrepräsentation auf Basis der „Ist auf“-Relation.

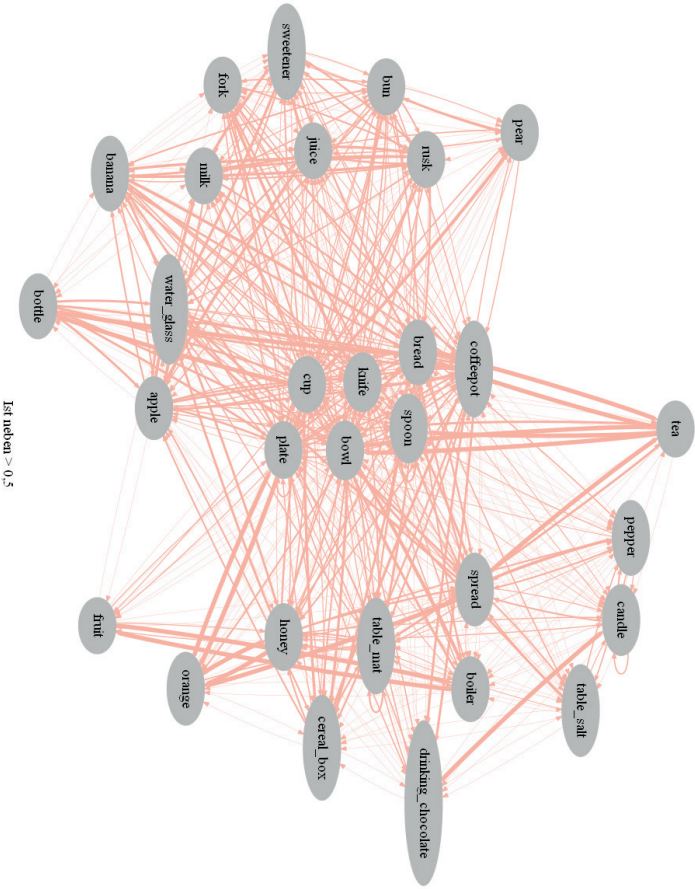


Abb. 8.27. Szenemrepräsentation auf Basis der „Ist neben“-Relation.

## Zusammenfassung und Ausblick

### 9.1 Ergebnisse der Arbeit und Erkenntnisse

#### *Objektmodellierung*

Der vorgestellte Sensoraufbau zur semi-automatischen Erfassung von Alltagsgegenständen ist in seiner Konzeption und Implementierung einzigartig. Die Kombination eines hochgenauen 3D-Sensors mit hochauflösenden und präzise positionierbaren Kameras garantiert qualitativ hervorragende Ergebnisse. Das Stereokamerasystem ist äquivalent zu vielen Kamerasystemen in aktuellen Servicerobotern und ermöglicht daher die schnelle und direkte Übernahme der Ergebnisse aus dem Modellierungscenter in die Anwendung. Durch weitere Bearbeitungsschritte werden mit den resultierenden Datensätzen auch die Anforderungen weiterer Teilgebiete der Servicerobotik an ein Objektmodell erfüllt, wie etwa Greifplanung oder Visualisierung.

Die Geschwindigkeit und Robustheit des Aufbaus wurde durch die beständige Erweiterung des Objektdatensatzes auf derzeit über 130 Objekte gezeigt. Die mit Hilfe dieser Daten erstellten Publikationen innerhalb der Forschung auf dem Gebiet der Servicerobotik, sowie die Zugriffszahlen auf die Webdatenbank, in der die Daten frei zugänglich sind, zeigen, dass die Ergebnisse dieser Arbeit gut angenommen werden und ein entsprechender Bedarf besteht.

Im Verlauf der Arbeit zeigte sich, dass die Nachbearbeitung der 3D-Daten, auch mit spezieller Software, der komplexeste Schritt in Richtung vollständige Automatisierung ist, was zur Folge hat, dass dieser Teil des Aufnahmeprozesses derzeit den zeitlich größten Aufwand erzeugt. Weiterhin erwiesen sich bestimmte Objekte als besonders diffizil, da ihre spezifischen Eigenschaften wie Material oder Form, für die Messverfahren der verwendeten Sensorik ungünstig sind. Schließlich muss die gewählte Beleuchtungsanordnung als suboptimal eingeschätzt werden, da Glanzlichter auf der Objektoberfläche teilweise nicht vermieden werden können und die Homogenität der Beleuchtungsintensität je nach Objekt dadurch teilweise stark variiert.

### *Szenenmodellierung*

Die Erweiterung des Objektmodells von klassen- und instanzspezifischen Attributen auf Beziehungen zu anderen Objekten in Szenen auf Basis realer Sensordaten, ist in dieser Form noch nicht vorgestellt worden. Der gewählte Aufnahmeprozess, mit Hilfe des portablen Tiefensensors unter Verwendung eines markerlosen SLAM-Verfahrens, gewährleistet die schnelle und unkomplizierte Erzeugung der benötigten Datengrundlage. Auch hier könnten die so gewonnenen Daten in ähnlicher Form von einem Service-



roboter erzeugt werden, was die Anwendung der vorgestellten Verfahren und Daten in einem solchen System direkt möglich macht.

Die Annotierung der Punktwolkendaten mittels der Platzierung von einfachen Boundingboxen lässt sich ohne spezielle Vorkenntnisse durchführen. Durch die Integration von State-of-the-Art-Verfahren im Bereich der Verarbeitung von Punktwolken, konnte der Annotierungsprozess beschleunigt und vereinfacht werden. Hierzu trägt auch die Rückführung der bereits gelernten Interobjektrelationen nachhaltig bei.

Die entwickelte „ist neben“-Relation konnte erfolgreich gegen eine Nutzerumfrage validiert werden. Innerhalb der mit Hilfe des Gesamtprozesses erzeugten Trainingsdaten, die aus nachgestellten, realen Szenen generiert wurden, können nach der Berechnung der Relationen Gesetzmäßigkeiten der betrachteten Szenen beobachtet werden, die mit der Alltagserfahrung des Menschen übereinstimmen. Durch die Rückführung dieser Daten in die Annotierung konnte gezeigt werden, dass sich die Klassenzugehörigkeit eines Objekts, allein durch die Relationen zu den umgebenden Objekten bereits mit hoher Güte schätzen lässt.

Zu erwähnen ist an dieser Stelle jedoch die große Menge an Trainingsdaten, die das Verfahren benötigt um großen Mengen unterschiedlicher Objekte und Szenen bewältigen zu können. Weiterhin ist die Zuordnung der Objektinstanzen in einer Szene zu einer Objektklasse, allein durch den annotierenden Benutzer, als potentiell kritisch anzusehen. An dieser Stelle kann eine Fragmentierung der Daten auftreten, für den Fall, dass zwei Benutzer unterschiedliche Granularitäten bei der Klassifizierung anwenden.

## 9.2 Ausblick

### *Mögliche Erweiterungen der Objektmodellierung*

Die aktuelle Implementierung des Modellierungscenters verwendet für die Positionsbestimmung des Stereokamerasystems während der Objektaufnahme eine a-priori erzeugte Kalibrierung. Diese gewährleistet zwar eine schnelle Ermittlung der Kamerapose, hat jedoch den Nachteil, nicht für alle möglichen relativen Posen zwischen Kamera und Objekt vorberechnet vorzuliegen. Eine mögliche Erweiterung an dieser Stelle wäre die Anbringung von Markern an den Rotationsteller, die von den Kameras zu jedem Zeitpunkt erfasst werden können. Auf diese Art ließe sich die Kamerapose direkt während der Objekterfassung relativ zu den Markern ermitteln.

Der aktuell im Modellierungscenter verwendete Tiefensensor bietet zwar eine sehr hohe Genauigkeit, ist jedoch auf Grund seiner Größe und Geschwindigkeit nicht in einem Serviceroboter verwendbar. Um eine weitere Korrespondenz zwischen den im Modellierungscenter erzeugten Objektdaten und denen mit einem Roboter generierbaren Daten herzustellen, könnte zusätzlich ein kleinerer, weniger genauer Sensor (z.B. Kinect) zur Anwendung gebracht werden. Hierdurch können die in Echtzeit erzeugten 3D-Daten mit den hochgenauen 3D-Modellen in Beziehung gesetzt werden.

Momentan liegen die Objektdaten in der Datenbank relativ unbearbeitet vor, um eine möglichst große Zahl an Methoden und Algorithmen der Nachbearbeitung zu ermöglichen. Es könnte jedoch vorteilhaft sein, bestimmte Merkmale im 2D- und 3D-Bereich für die einzelnen Datensätze mit standardisierten Verfahren vorzuberechnen. Die so gewonnen Merk-

male können z.B. als Benchmark Anwendung finden, oder direkt für nachgelagerte Verfahren wie Lokalisierung oder Erkennung verwendet werden.

Bei der Diskussion der Ergebnisse wurde bereits angesprochen, dass die Nachbearbeitung der 3D-Rohdaten einen der kritischsten und komplexesten Schritte des gesamten Prozesses darstellt. Durch die Analyse der bereits vorhandenen Datensätze mit Verfahren des maschinellen Lernens könnten Ansätze entwickelt werden, die eine weiterreichende Automatisierung dieser Nachbearbeitung ermöglichen. So könnten typische Artefakte und Fehler des Sensorsystems etwa schon während der Aufnahme korrigiert werden.

### *Mögliche Erweiterungen der Szenenmodellierung*

Die in der Arbeit vorgestellte Szenenmodellierung greift für die Evaluierung auf eine größere Menge an Rohdaten einer spezifisch ausgewählten Szenenklasse zurück, jedoch könnte in fortführenden Arbeiten dieser Bestand deutlich erweitert werden. Insbesondere die Betrachtung unterschiedlicher Szenenklassen und deren Vergleich ist hierbei von Interesse. Eine menschenzentrierte Umgebung bietet hierfür viele weitere Ansatzpunkte, etwa die Unterschiede zwischen privaten und industriellen Umfeldern (z.B. Haushalt gegenüber Fertigungshalle), oder die Betrachtung von Naturszenarien (z.B. landwirtschaftliche Ernte).

Die vorgestellten Relationen sind in ihrer Berechnung unabhängig von der Orientierung der beteiligten Objekte. Betrachtet man jedoch typische Alltagsszenen, wie bspw. einen Computerarbeitsplatz in einer Büroumgebung, so kann man beobachten, dass bestimmte Objekte wie etwa ein Monitor oder eine Tastatur, bedingt durch ihre Funktion, spezielle Orien-

tierungen aufweisen. Auf Basis dieser zusätzlichen Information könnten weitere, komplexere Objektrelationen entwickelt werden, die die funktionalen Zusammenhänge zwischen Objekten einer Szene modellieren und so zusätzliche Informationen zur Verfügung stellen.

Weiterhin wäre auch eine Weiterentwicklung des Annotierungsprozesses wünschenswert. So könnte etwa ein Robotersystem genutzt werden, das mit seinen Sensoren eine Szene betrachtet und dabei von einem Menschen über die räumliche Positionierung der Objekte unterrichtet wird. Dies könnte mit Hilfe eines klassischen Objekterkennungssystems, unterstützt durch das relationale Hintergrundwissen, realisiert werden. Das System könnte auf diese Art und Weise beständig weiter lernen und damit die Szenenmodelle verfeinern.

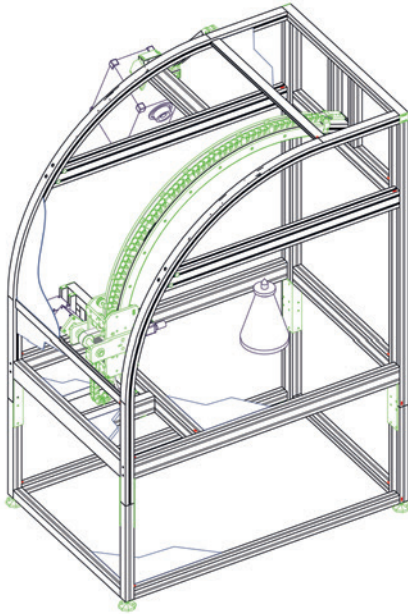
Schließlich könnte dem Problem der unterschiedlichen Granularisierung der Objektklassen begegnet werden, indem keine expliziten Klassen mehr vorgegeben werden, sondern lediglich Ähnlichkeiten zu bereits beobachteten Objekten betrachtet werden. Im Zusammenspiel mit weiteren Objektattributen könnte ein System auf diese Art eine eigenständige Klassifizierung durchführen, die die menschlichen Klassenbegriffe lediglich als Verweis enthält.

**A**

---

## **Konstruktion Modellierungcenter**

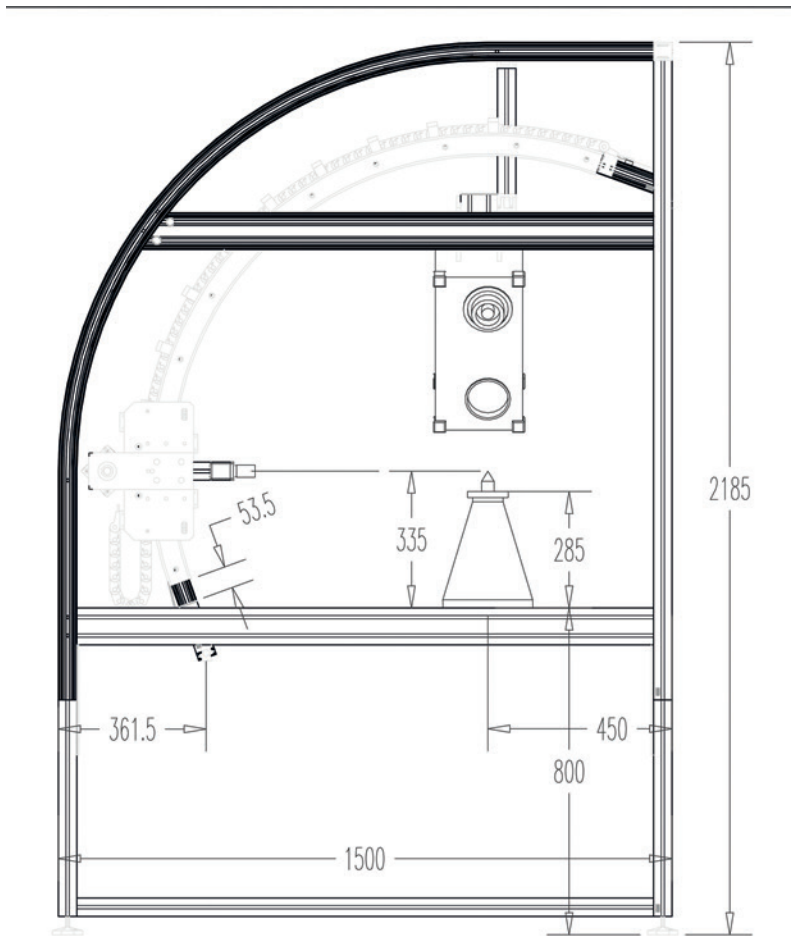
Nachstehend sind die von der Firma Norcan auf Basis der vorgegebenen Auslegung angefertigten Konstruktionszeichnungen des entwickelten Sensoraufbaus.



- Bemerkungen :
- Der Modellierungcenter besteht aus 3 Baugruppen :
  - Das Hauptgestell mit dem Rundbogen und Kamerazylinder
  - Das Untergestell mit den Stellfüßen
  - Der Solliche Trogram für den Scanner
  - Flächenelemente DA bis DC aus Sperrholz CTBX, Stärke 22mm, befestigt mit Befestigungslaschen N1404.
  - Flächenelemente DD aus Polycarbonat farblos, Stärke 3mm, in die Nuten montiert mit Hilfe des Einfassprofilis N0714.
  - Flächenelemente DE & DF aus Polycarbonat farblos, Stärke 5mm, in die Nuten montiert mit Hilfe des Einfassprofilis N0714.

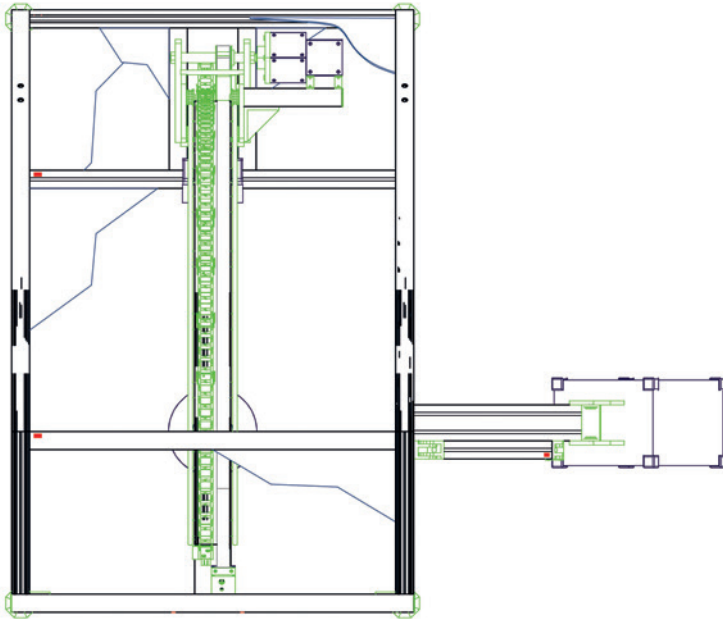
UNIVERSITÄT KARLSRUHE D-76131 KARLSRUHE		<b>NORCAN</b>	
Doc. N°	NORCAN / 13170863	SE des Modèles N° 108	
Modèle n°	05/12/08	Projet	108
<small>Il y a une note de détails de NORCAN 108. Il y a un plan de montage à la main et un détail des éléments de montage.</small>			

**Abb. A.1.** Isometrische Ansicht des Modellierungcenters.



UNIVERSITÄT KARLSRUHE D-76131 KARLSRUHE				<b>NORCAN</b>	
Plan N°	NO8M / 13710BD3	48 rue des Animateurs BP 120 67503 Haguenau Cedex			
Venté le	08/12/08	Par	PAS	Est.	/ AO
Ce plan est la propriété de NORCAN S.A.S. Il ne peut être communiqué à des tiers et/ou reproduit sans autorisation écrite.				Tél 03.88.93.26.26 Fax 03.88.93.30.74	

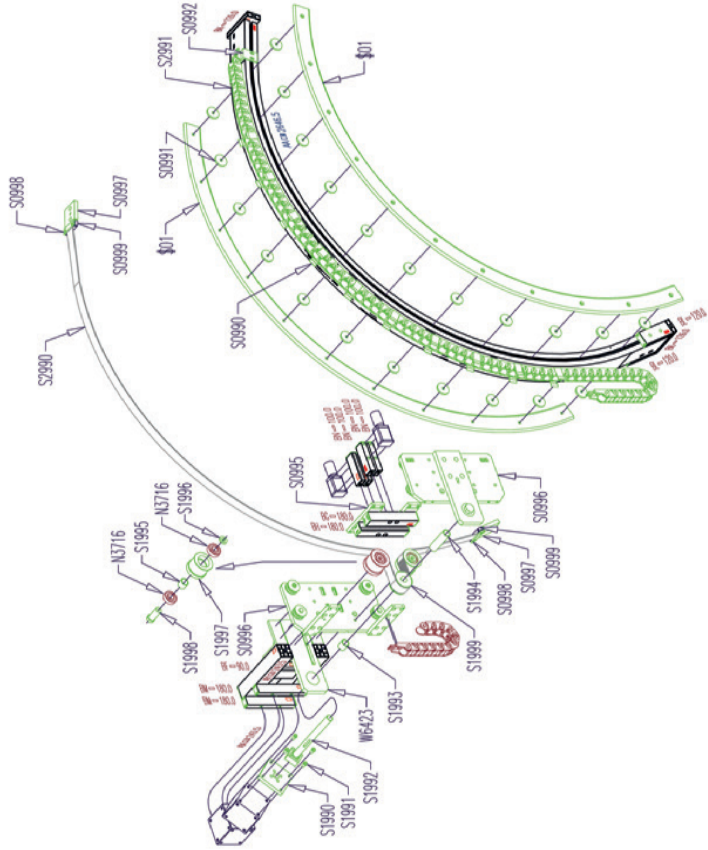
Abb. A.2. Seitenansicht des Modellierungcenters.



UNIVERSITÄT KARLSRUHE D-76131 KARLSRUHE				<b>NORCAN</b>	
Plan N°	NO8M / 137108D3			48 rue des Adolpheurs EP 120 67503 Hognanos Cedex	
Version	08/12/08	Par	PAS	Éch.	/ AO
Ce plan est la propriété de NORCAN S.A.S. Il ne peut être communiqué à des tiers et/ou reproduit sans autorisation écrite.				Tél 03.88.93.26.26 Fax 03.88.93.30.74	

Abb. A.3. Draufsicht des Modellierungszenters.





UNIVERSITÄT KARLSRUHE		NORKAN	
D-76131 KARLSRUHE		48 von der Aulstraße 10 710	
Matr. Nr.	NORM / 137110803	1000	Rechner Code
Matr. Nr.	08/712/08 Nr. PPS	Matr. Nr.	08/712/08 Nr. PPS

Abb. A.4. Detailsicht des Kameraschlittens im Modellierungszenters.



# B

---

## Verwendete Softwareframeworks

In diesem Kapitel werden die für die Implementierung, der in der Arbeit vorgestellten Verfahren verwendeten Softwareframeworks und -bibliotheken vorgestellt.

### B.1 Pointcloud Library (PCL)

Die Pointcloud Library (PCL) - zu deutsch die Punktwolkenbibliothek - wurde von Radu B. Rusu im Rahmen seiner Dissertation entwickelt. Die Bibliothek stellt Methoden zur Verarbeitung von Tiefendaten in Form von Punktwolken zur Verfügung. Sie ist in unterschiedliche Module unterteilt, u.a. finden sich Funktionssammlungen zu den Themen:

- Visualisierung
- Registrierung
- Filterung

- Tracking
- Eingabe/Ausgabe

URL	verwendete Version	Lizenz
<a href="http://www.pointclouds.org">http://www.pointclouds.org</a>	1.5.1	BSD

## B.2 Rapidform.DLL SDK

Dieses Softwarepaket stellt State-of-the-Art-Verfahren zur Bearbeitung von Punktwolken und Dreiecksnetzen, sowie NURBS-Kurven und -Oberflächen zur Verfügung. Die kommerziell vertriebene Bibliothek ist nur unter Windows-Systemen einsetzbar und stellt mehrere APIs bereit, die unter anderem folgende Funktionalitäten enthalten:

- Datei Eingabe/Ausgabe in verschiedenen Formaten
- Filterung von Punktwolken
- Registrierung und Fusion von Dreiecksnetzen
- Reduzierung der Auflösung von Dreiecksnetzen
- Remeshing und Reparatur von Dreiecksnetzen
- Analyse und Vergleich von Scandaten mit Referenzdaten

URL	verwendete Version	Lizenz
<a href="http://www.rapidform.com/products/rapidform-dll/">http://www.rapidform.com/products/rapidform-dll/</a>	1.2.0	kommerziell

### B.3 Object Oriented Rendering Engine (Ogre)

Ogre ist eine der bekanntesten Open-Source 3D-Rendering-Engines der Welt. Das Framework stellt eine Abstraktionsebene zwischen Grafikhardware und Software zur Verfügung, mit spezifischen Implementierungen für diverse DirectX-Versionen sowie OpenGL auf unterschiedlichen Plattformen (Window, Linux, Android, iOS). Weiterhin werden folgende Funktionalitäten geboten:

- Eigene Definitionssprache für Materialien (Fixed Pipeline)
- Unterstützung für Shader (Cg, HLSL, GLSL)
- Skelett- und Shape-Animation
- Szenengraph mit variablen Implementierungen (BSP, Octree)
- Spezialeffekte mit Partikeln und Postprocessing

URL	verwendete Version	Lizenz
<a href="http://www.ogre3d.org">http://www.ogre3d.org</a>	1.7.4	MIT

### B.4 wxWidgets

Im Bereich der GUI-Bibliotheken ist wxWidgets bereits seit vielen Jahren ein weit verbreiteter und bekannter Vertreter. Es bietet alle Merkmale, die man von einem modernen GUI-System erwartet, wie etwa:

- Verschiedene Widget-Klassen wie Fenster, Knöpfe, Dialoge etc.

- Umfangreiches Ereignis-System
- Unterstützung für unterschiedliche Plattformen (Windows, Linux, MacOS) mit nativen Themen

URL	verwendete Version	Lizenz
<a href="http://www.wxwidgets.org">http://www.wxwidgets.org</a>	2.9.1	wxWindows (L-GPL)

## B.5 Ogre Virtual Scene Environment (OVISE)

Dieses am Lehrstuhl Dillmann entwickelte Framework basiert auf wxWidgets und Ogre3D und dient der Visualisierung der Umwelt im Bereich der Servicerobotik. Es beinhaltet ein allgemeines Entitätensystem, welches unterschiedliche Objekte und Agenten in der Umweltrepräsentation eines Roboters darstellen kann. Durch eine Plugin-Schnittstelle können spezifische Visualisierungen der Entitäten realisiert und auch eine netzwerkgesteuerte Manipulation der Entitäten vorgenommen werden. Weitere Merkmale:

- Punktwolkensvisualisierung
- Szenenbeschreibung in XML
- Robotervisualisierung (gelenkwinkelabhängige Animation)
- Multiplattform (Windows, Linux)

URL	verwendete Version	Lizenz
<a href="http://code.google.com/p/ovise">http://code.google.com/p/ovise</a>	0.6	MIT

## B.6 Intergrating Vision Toolkit (IVT)

Die Bildbearbeitungsbibliothek IVT stellt verschiedenste Funktionen zum Ansprechen von Kamerahardware, sowie zur echtzeitfähigen Bearbeitung von Bildern zur Verfügung. Die Multiplattform-Software bietet ein durchgehendes objektorientiertes Design bei hoher Performanz. Merkmale:

- Entzerrung
- Kantendetektion
- Merkmalsberechnung und Matching
- Kalibrierung
- Verschiedene Filter (Sobel, Prewitt, etc.)

URL	verwendete Version	Lizenz
<a href="http://ivt.sourceforge.net">http://ivt.sourceforge.net</a>	1.3.19	BSD

## B.7 Boost

Boost ist eine Sammlung an C++-Bibliotheken mit unterschiedlichsten Anwendungen. Bereitgestellt werden Implementierungen zu Smart-Pointern, spezielle Container, asynchrone Eingabe/Ausgabe, Manipulation von Zeichenketten uvm. In vielen Fällen beinhaltet Boost erste Im-

plementierungen bzw. Referenzimplementierungen für neue Standards der Sprache C++.

URL	verwendete Version	Lizenz
<a href="http://www.boost.org">http://www.boost.org</a>	1.44.0	Boost

## B.8 FreeImage

FreeImage ist eine Bildbearbeitungsbibliothek, mit dem Fokus auf Unterstützung möglichst vieler verschiedener Formate unter einer gemeinsamen Schnittstelle. Ausgewählte Merkmale sind:

- Plugin-Architektur
- Farbkonvertierungen
- Allgemeine Bildbearbeitung (Größe ändern, Spiegelung, etc.)
- Unterstützung für Metadaten (Exif)

URL	verwendete Version	Lizenz
<a href="http://freeimage.sourceforge.net">http://freeimage.sourceforge.net</a>	3.15.1	GPL



---

## Literaturverzeichnis

- [3DModelFree.com 12] 3DModelFree.com. Website. 2012.04.24.
- [Agarwal 02] S. Agarwal, D. Roth. Learning a sparse representation for object detection. Tagungsband: Proceedings of the European Conference on Computer Vision, Band 4, Seiten 113–130, Copenhagen, Denmark, May 2002. Springer-Verlag.
- [Aleotti 11] J. Aleotti, S. Caselli. Part-based robot grasp planning from human demonstration. Tagungsband: Robotics and Automation (ICRA), 2011 IEEE International Conference on, Seiten 4554–4560, may 2011.
- [Allied Vision 11] Allied Vision. Datenblatt marlin 145. [http://www.alliedvisiontec.com/fileadmin/content/PDF/Products/Data\\_sheet/Cameras/Marlin/Marlin\\_DataSheet\\_F-145\\_V4.0.0\\_en.pdf](http://www.alliedvisiontec.com/fileadmin/content/PDF/Products/Data_sheet/Cameras/Marlin/Marlin_DataSheet_F-145_V4.0.0_en.pdf). 2011.11.28.

- [Anand 12] Abhishek Anand, Hema Koppula, Thorsten Joachims, Ashutosh Saxena. Contextually guided semantic labeling and search for 3d point clouds. 2012.
- [Aydemir 10] Alper Aydemir, Kristoffer Sjöo, Patric Jensfelt. Object search on a mobile robot using relational spatial information. Tagungsband: Proc. of the 11th Int Conference on Intelligent Autonomous Systems (IAS-11), 2010.
- [Azad 03] Pedram Azad. Entwurf, aufbau und kalibrierung eines 3d-laser-scanners für medizinische anwendungen. Diplomarbeit, Universität Karlsruhe, 2003.
- [Azad 08a] Pedram Azad. Visual perception for manipulation and imitation in humanoid robots. Dissertation, 2008.
- [Azad 08b] Pedram Azad, Tilo Gockel, Rüdiger Dillmann. Computer Vision : principles and practice. Elektor International, [s. l.], 2008.
- [Azad 10] Pedram Azad. IVT (Integrating Vision Toolkit). <http://ivt.sourceforge.net>. 2010.02.09.
- [Azad 11] Pedram Azad, David Munch, Tamim Asfour, Rüdiger Dillmann. 6-dof model-based tracking of arbitrarily shaped 3d objects. Tagungsband: ICRA, Seiten 5204–5209, 2011.
- [Azad 12] Pedram Azad. Keyetech ar marker recognition. <http://www.keyetech.de/de/produkte.html>. 2012.01.22.

- [Bay 08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. <ce:title>Similarity Matching in Computer Vision and Multimedia</ce:title>.
- [Becher 08] Regine Becher. Semantische Objektmodellierung mittels multimodaler Interaktion. Dissertation, Karlsruhe, 2008. ; Pb.: EUR 29,50.
- [Bentley 75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, Sept. 1975.
- [Berenson 07] Dmitry Berenson, Rosen Diankov, Koichi Nishiwaki, Satoshi Kagami, James Kuffner. Grasp planning in complex scenes. Tagungsband: IEEE-RAS International Conference on Humanoid Robots (Humanoids07), December 2007.
- [Biederman 82] I. Biederman, R.J. Mezzanotte, J.C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982.
- [Blodow 11] Nico Blodow, Lucian Cosmin Goron, Zoltan-Csaba Marton, Dejan Pangercic, Thomas Rühr, Moritz Tenorth, Michael Beetz. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. Tagungsband: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, September, 25–30 2011.

- [Bouguet 99] Jean Y. Bouguet. Visual Methods for three-dimensional modeling. Dissertation, California Institute of Technology, Pasadena, California, Mai 1999.
- [Bronstein 10] A. M. Bronstein, M. M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. A. Godil, L. J. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov, G. Patane, M. Spagnuolo, R. Toldo. Shrec 2010: robust large-scale shape retrieval benchmark. Tagungsband: Eurographics Workshop on 3D Object Retrieval, Seite 8, Norrköping, -1, May 2010.
- [Brook 11] P. Brook, M. Ciocarlie, Kaijen Hsiao. Collaborative grasp planning with multiple object representations. Tagungsband: Robotics and Automation (ICRA), 2011 IEEE International Conference on, Seiten 2851 –2858, may 2011.
- [CGTrader 12] CGTrader. Website. 2012.04.24.
- [Choi 10] Changhyun Choi, H.I. Christensen. Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation. Tagungsband: Robotics and Automation (ICRA), 2010 IEEE International Conference on, Seiten 4048 –4055, may 2010.
- [Cohn 01] A. G. Cohn, S. M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundam. Inf.*, 46(1-2):1–29, Jan. 2001.
- [Computer Vision Department, CMU 12] Computer Vision Department, CMU. List of computer vision test images. 2012.04.24.

- [Costea 11] A.D. Costea, R. Varga, T. Marita, S. Nedevschi. Refining object recognition using scene specific object appearance frequencies. Tagungsband: Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on, Seiten 179 –185, aug. 2011.
- [Creaform 3D 12] Creaform 3D. Handyscan datenblatt. 2012.04.23.
- [Cyberware Inc. 11] Cyberware Inc. Cyberware website. <http://www.cyberware.com>. 2011.11.10.
- [Dalal 05] Navneet Dalal, Bill Triggs. Histograms of oriented gradients for human detection. Tagungsband: In CVPR, Seiten 886–893, 2005.
- [DAVID Vison Systems GmbH 12] DAVID Vison Systems GmbH. David laserscanner website. 2012.04.23.
- [Davis 02] J. Davis, S.R. Marschner, M. Garr, M. Levoy. Filling holes in complex surfaces using volumetric diffusion. Tagungsband: 3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on, Seiten 428 –441, june 2002.
- [Dementhon 95] Daniel F. Dementhon, Larry S. Davis. Model-based object pose in 25 lines of code. *Int. J. Comput. Vision*, 15(1-2):123–141, Juni 1995.
- [Deng 09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. Tagungsband: CVPR09, 2009.

- [Engelhard 11] N. Engelhard, F. Endres, J. Hess, J. Sturm, W. Burgard. Real-time 3d visual slam with a hand-held camera. Tagungsband: Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden, April 2011.
- [Everingham 06a] M. Everingham, A. Zisserman, C. K. I. Williams, L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. <http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf>. 2012.04.24.
- [Everingham 06b] Mark Everingham, Andrew Zisserman, Christopher K. I. Williams, Luc Van Gool, Moray Allan, Christopher M. Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorko, Stefan Duffner, Jan Eichhorn, Jason D. R. Farquhar, Mario Fritz, Christophe Garcia, Tom Griffiths, Frederic Jurie, Daniel Keysers, Markus Koskela, Jorma Laaksonen, Diane Larlus, Bastian Leibe, Hongying Meng, Hermann Ney, Bernt Schiele, Cordelia Schmid, Edgar Seemann, John Shawe-taylor, Amos Storkey, Or Szedmak, Bill Triggs, Ilkay Ulusoy, Ville Viitaniemi, Jianguo Zhang. The 2005 pascal visual object classes challenge. 2006.
- [Everingham 10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, Juni 2010.
- [Fautz 02] M. Fautz. Objekt- und Texturrekonstruktion mit einer robotergeführten Kamera. *Berichte aus der Informatik*. Shaker, 2002.

- [Fei-Fei 04] L. Fei-Fei, R. Fergus, P. Perona. Learning generative visual models from few training examples an incremental bayesian approach tested on 101 object categories. Tagungsband: Proceedings of the Workshop on Generative-Model Based Vision, Washington, DC, June 2004.
- [Fergus 03] R. Fergus, P. Perona, A. Zisserman. Object class recognition by unsupervised scale-invariant learning. Tagungsband: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Band 2, Seiten 264–271, Madison, Wisconsin, June 2003.
- [Fischler 81] Martin A. Fischler, Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, Juni 1981.
- [Fisher 10] Matthew Fisher, Pat Hanrahan. Context-based search for 3d models. Tagungsband: ACM SIGGRAPH Asia 2010 papers, SIGGRAPH ASIA '10, Seiten 182:1–182:10, New York, NY, USA, 2010. ACM.
- [Freksa 92] Christian Freksa. Using orientation information for qualitative spatial reasoning. Tagungsband: Proceedings of the International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning on Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Seiten 162–178, London, UK, UK, 1992. Springer-Verlag.

- [Galleguillos 10] Carolina Galleguillos, Serge Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712 – 722, 2010. <ce:title>Special Issue on Multi-Camera and Multi-Modal Sensor Fusion</ce:title>.
- [Gilbert 88] E. G. Gilbert, D. W. Johnson, S. S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *Robotics and Automation, IEEE Journal of*, 4(2):193–203, 1988.
- [Glover 09] Jared Glover, Daniela Rus, Nicholas Roy. Probabilistic models of object geometry with application to grasping. *The International Journal of Robotics Research*, 28(8):999–1019, 2009.
- [Gockel 06a] T. Gockel. Interaktive 3D-Modellerfassung mittels One-shot-Musterprojektion und schneller Registrierung. Univ.-Verl. Karlsruhe, 2006.
- [Gockel 06b] Tilo Gockel. Logi-scan-3d. 2012.04.23.
- [Goldfeder 09a] C. Goldfeder, M. Ciocarlie, Hao Dang, P.K. Allen. The columbia grasp database. *Tagungsband: Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, Seiten 1710 – 1716, may 2009.
- [Goldfeder 09b] C. Goldfeder, M. Ciocarlie, J. Peretzman, Hao Dang, P.K. Allen. Data-driven grasping with partial sensor data. *Tagungsband: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Seiten 1278 –1283, oct. 2009.



- [Gonzalez-Aguirre 10] D. Gonzalez-Aguirre, T. Asfour, R. Dillmann. Eccentricity edge-graphs from hdr images for object recognition by humanoid robots. Tagungsband: Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, Seiten 144 –151, dec. 2010.
- [Google 12] Google. Google 3d warehouse. 2012.04.24.
- [Grundmann 08] T. Grundmann, R. Eidenberger, R.D. Zoellner, Zhixing Xue, S. Ruehl, J.M. Zoellner, R. Dillmann, J. Kuehnle, A. Verl. Integration of 6d object localization and obstacle detection for collision free robotic manipulation. Tagungsband: System Integration, 2008 IEEE/SICE International Symposium on, Seiten 66 –71, dec. 2008.
- [Grundmann 10a] T. Grundmann, M. Fiegert, W. Burgard. Probabilistic rule set joint state update as approximation to the full joint state estimation applied to multi object scene analysis. Tagungsband: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Seiten 2047 –2052, oct. 2010.
- [Grundmann 10b] Thilo Grundmann, Robert Eidenberger, Martin Schneider, Michael Fiegert. Robust 6d pose determination in complex environments for one hundred classes. Tagungsband: ICINCO (2), Seiten 301–308, 2010.
- [Harada 08] K. Harada, K. Kaneko, F. Kanehiro. Fast grasp planning for hand/arm systems based on convex model. Tagungsband: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, Seiten 1162 –1168, may 2008.

- [Harris 88] Chris Harris, Mike Stephens. A combined corner and edge detector. Tagungsband: In Proc. of Fourth Alvey Vision Conference, Seiten 147–151, 1988.
- [He 11] Li He, Hui Wang, Hong Zhang. Object detection by parts using appearance, structural and shape features. Tagungsband: Mechatronics and Automation (ICMA), 2011 International Conference on, Seiten 489–494, aug. 2011.
- [Holroyd 10] Michael Holroyd, Jason Lawrence, Todd Zickler. A coaxial optical scanner for synchronous acquisition of 3D geometry and surface reflectance. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010), 2010.
- [Huebner 09] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, R. Dillmann. Grasping known objects with humanoid robots: A box-based approach. Tagungsband: Advanced Robotics, 2009. ICAR 2009. International Conference on, Seiten 1–6, june 2009.
- [Inus Technology 10] Inus Technology. Rapidform DLL. [http://www.rapidform.com/Contents/Product/Skin/ProductETC/category\\\_id/53](http://www.rapidform.com/Contents/Product/Skin/ProductETC/category\_id/53). 2010.02.15.
- [Izadi 11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, Andrew Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. Tagungsband: Proceedings of the 24th annual ACM symposium on User interface software and technology, UIST '11, Seiten 559–568, New York, NY, USA, 2011. ACM.

- [Jeromin 12] A.J. Jeromin. Kinect turntable 3d scanner. 2012.04.23.
- [Kasper 11] A. Kasper, R. Jäkel, R. Dillmann. Using spatial relations of objects in real world scenes for scene structuring and scene understanding. Tagungsband: Proceedings of the 15th International Conference on Advanced Robotics (ICAR '11), Tallinn, Estonia, 2011.
- [Kasper 12a] Alexander Kasper, Zhixing Xue, Rüdiger Dillmann. The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. The International Journal of Robotics Research, 2012.
- [Kasper 12b] Alexander Kasper, Zhixing Xue, Rüdiger Dillmann. Towards Service Robots for Everyday Environments, Band 76 of Springer Tracts in Advanced Robotics (S.T.A.R.), Kapitel 5.1 Semi Automatic Object Modeling for a Service Robot, Seiten 167–179. Springer, 2012.
- [Kochenderfer 03] Mykel J. Kochenderfer, Rakesh Gupta. Common sense data acquisition for indoor mobile robots. Tagungsband: In Nineteenth National Conference on Artificial Intelligence (AAAI-04, Seiten 605–610. AAAI Press / The MIT Press, 2003.
- [Kollar 09] T. Kollar, N. Roy. Utilizing object-object and object-scene context when planning to find things. Tagungsband: Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, Seiten 2168 –2173, may 2009.
- [Koppula 11] H.S. Koppula, A. Anand, T. Joachims, A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. Tagungsband: NIPS, 2011.

- [Kragic 06] D. Kragic, V. Kyrki. Initialization and system modeling in 3-d pose tracking. Tagungsband: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, Band 4, Seiten 643 –646, 0-0 2006.
- [Kragic 09] Danica Kragic, Markus Vincze. Vision for robotics. Found. Trends Robot, 1(1):1–78, Jan. 2009.
- [Kuehnle 09] J. Kuehnle, A. Verl, Zhixing Xue, S. Ruehl, J.M. Zoellner, R. Dillmann, T. Grundmann, R. Eidenberger, R.D. Zoellner. 6d object localization and obstacle detection for collision-free manipulation with a mobile service robot. Tagungsband: Advanced Robotics, 2009. ICAR 2009. International Conference on, Seiten 1 –6, june 2009.
- [Lai 10] Kevin Lai, Dieter Fox. Object recognition in 3d point clouds using web data and domain adaptation. The International Journal of Robotics Research, 29(8):1019–1037, 2010.
- [Lai 11] Kevin Lai, Liefeng Bo, Xiaofeng Ren, Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. ICRA. 2012.04.24.
- [Lecun 04] Yann Lecun, Fu Jie Huang, L?on Bottou. Learning methods for generic object recognition with invariance to pose and lighting. Tagungsband: In Proceedings of CVPR 04. IEEE Press, 2004.
- [Leibe 04] B. Leibe, A. Leonardis, B. Schiele. Combined object categorization and segmentation with an implicit shape model. Tagungsband: Proceedings of the Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic, May 2004.
- [Leibe 12] Bastian Leibe. Website mit testdatensätzen. 2012.04.24.

- [Levoy 10] Marc Levoy. Stanford spherical gantry. <http://graphics.stanford.edu/projects/gantry>. 2010.10.15.
- [Li 08] Wenjing Li, G. Bebis, N.G. Bourbakis. 3-d object recognition using 2-d views. *Image Processing, IEEE Transactions on*, 17(11):2236–2255, nov. 2008.
- [Liu 11] Qingqian Liu. 3d-szenendigitalisierung. Diplomarbeit, Karlsruhe Institute of Technology (KIT), 2011.
- [Lowe 04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.
- [Luebke 01] D.P. Luebke. A developer’s survey of polygonal simplification algorithms. *Computer Graphics and Applications, IEEE*, 21(3):24–35, may/jun 2001.
- [Malisiewicz 09] Tomasz Malisiewicz, Alexei A. Efros. Beyond categories: The visual memex model for reasoning about object relationships. Tagungsband: NIPS, December 2009.
- [Meilinger 10] Tobias Meilinger, Gottfried Vosgerau. Putting egocentric and allocentric into perspective. Tagungsband: Proceedings of the 7th international conference on Spatial cognition, SC’10, Seiten 207–221, Berlin, Heidelberg, 2010. Springer-Verlag.
- [MESA Imaging AG 12] MESA Imaging AG. Swissranger sr4000 datenblatt. [http://www.mesa-imaging.ch/dlm.php?fname=pdf/SR4000\\_Data\\_Sheet.pdf](http://www.mesa-imaging.ch/dlm.php?fname=pdf/SR4000_Data_Sheet.pdf). 04.03.2012, 2012.03.04.

- [Miller 03] A.T. Miller, S. Knoop, H.I. Christensen, P.K. Allen. Automatic grasp planning using shape primitives. Tagungsband: Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, Band 2, Seiten 1824 – 1829 vol.2, sept. 2003.
- [Moratz 04] Reinhard Moratz, Dr. Thomas Barkowsky. Qualitative spatial reasoning about oriented points. 2012.05.15.
- [Muldoon 12] Matthew Muldoon. Blendswap website. 2012.04.24.
- [Nayar 96] S.K. Nayar, S.A. Nene, H. Murase. Real-time 100 object recognition system. Tagungsband: Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, Band 3, Seiten 2321 –2325 vol.3, apr 1996.
- [Oliva 01] Aude Oliva, Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision, 42:145–175, 2001.
- [Opelt 04] A. Opelt, M. Fussenegger, A. Pinz, P. Auer. Generic object recognition with boosting. Technischer Bericht TR-EMT-2004-01, EMT, TU Graz, Austria, 2004. Submitted to the IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [PASCAL 12] PASCAL. The pascal object recognition database collection. 2012.04.24.
- [Pauly 01] Mark Pauly, Markus Gross. Spectral processing of point-sampled geometry. 2012.01.22.

- [Polygon Technology GmbH 12] Polygon Technology GmbH. Qtsculp-  
tor datenblatt. 2012.04.23.
- [Popovic 11] Mila Popovic, Gert Kootstra, Jimmy Alison Jørgensen, Da-  
nica Kragic, Norbert Krüger. Grasping unknown objects using an early  
cognitive vision system for general scene understanding. Tagungsband:  
Proceedings of the IEEE/RSJ International Conference on Intelligent  
Robots and Systems (IROS), Seiten 987–994. IEEE, 2011. ? 2011 IE-  
EE. Personal use of this material is permitted. Permission from IEEE  
must be obtained for all other uses, in any current or future media, inclu-  
ding reprinting/republishing this material for advertising or promotio-  
nal purposes, creating new collective works, for resale or redistribution  
to servers or lists, or reuse of any copyrighted component of this work  
in other works.
- [PrimeSense, Ltd. 12] PrimeSense, Ltd. Ps1080 soc produktweb-  
site. [http://www.primesense.com/en/technology/115-the-  
primesense-3d-sensing-solution](http://www.primesense.com/en/technology/115-the-primesense-3d-sensing-solution). 08.03.2012, 2012.03.08.
- [Profactor GmbH 12] Profactor GmbH. Reconstructme website.  
2012.04.23.
- [Quelhas 07] Pedro Quelhas, Florent Monay, Jean-Marc Odobez, Daniel  
Gatica-Perez, Tinne Tuytelaars. A thousand words in a scene. IEEE  
Transactions on Pattern Analysis and Machine Intelligence, 29:1575–  
1589, 2007.

- [Ramanan 11] Amirthalingam Ramanan, Mahesan Niranjan. A review of codebook models in patch-based visual object recognition. *Journal of Signal Processing Systems*, Seiten 1–20, 2011. 10.1007/s11265-011-0622-x.
- [Rosenbrock 60] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.
- [Rusinkiewicz 01] Szymon Rusinkiewicz, Marc Levoy. Efficient variants of the ICP algorithm. Tagungsband: Third International Conference on 3D Digital Imaging and Modeling (3DIM), Juni 2001.
- [Russell 08] Bryan Russell, Antonio Torralba, Kevin Murphy, William Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008. 10.1007/s11263-007-0090-8.
- [Rusu 09] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. Dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.
- [Saito 11] Manabu Saito, Haseru Chen, Kei Okada, Masayuki Inaba, Lars Kunze, Michael Beetz. Semantic object search in large-scale indoor environments. Tagungsband: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World, San Francisco, CA, USA, September, 25–30 2011.



- [Santosh K. Divvala 09] James H. Hays Alexei A. Efros Martial Hebert Santosh K. Divvala, Derek Hoiem. An empirical study of context in object detection. Tagungsband: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 2009.
- [Saut 12] Jean-Philippe Saut, Daniel Sidobre. Efficient models for grasp planning with a multi-fingered hand. Robotics and Autonomous Systems, 60(3):347 – 357, 2012. <ce:title>Autonomous Grasping</ce:title>.
- [Schall 05] Oliver Schall, Alexander Belyaev, Hans-Peter Seidel. Robust filtering of noisy scattered point data. In Mark Pauly, Matthias Zwicker, Hrsg., Tagungsband: IEEE/Eurographics Symposium on Point-Based Graphics, Seiten 71–77, Stony Brook, New York, USA, 2005. Eurographics Association.
- [Scharstein 02] Daniel Scharstein, Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vision, 47:7–42, April 2002.
- [Schulze 12] Jørgen Schulze, Daniel Tenedorio. Kinect 3d object and people scan. 2012.04.23.
- [Shi 94] Jianbo Shi, Carlo Tomasi. Good features to track. 2012.12.11.
- [Shilane 04] Philip Shilane, Patrick Min, Michael Kazhdan, Thomas Funkhouser. The princeton shape benchmark. Tagungsband: Shape Modeling International, Juni 2004.

- [SICK AG 11] SICK AG. Produktwebsite. [http://www.sick.com/group/DE/home/products/product\\_news/laser\\_measurement\\_systems/Seiten/lms5xx\\_laser\\_measurement\\_sensors.aspx](http://www.sick.com/group/DE/home/products/product_news/laser_measurement_systems/Seiten/lms5xx_laser_measurement_sensors.aspx). 2011.11.06.
- [Sinapov 11] Jivko Sinapov, Taylor Bergquist, Connor Schenck, Ugonna Ohiri, Shane Griffith, Alexander Stoytchev. Interactive object recognition using proprioceptive and auditory feedback. *The International Journal of Robotics Research*, 30(10):1250–1262, 2011.
- [Sjöo 10] Kristoffer Sjöo, Alper Aydemir, T. Mörwald, K. Zhou, Patric Jensfelt. Mechanical support as a spatial abstraction for mobile robots. Tagungsband: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2010.
- [Sjöo 11] Kristoffer Sjöo, Andrzej Pronobis, Patric Jensfelt. Functional topological relations for qualitative spatial representation. Tagungsband: Proceedings of the 15th International Conference on Advanced Robotics (ICAR'11), Tallinn, Estonia, Juni 2011.
- [Speth 08] J. Speth, A. Morales, P.J. Sanz. Vision-based grasp planning of 3d objects by extending 2d contour based algorithms. Tagungsband: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, Seiten 2240–2245, sept. 2008.
- [Stark 98] Michael M. Stark, Richard F. Riesenfeld. Wordnet: An electronic lexical database. Tagungsband: Proceedings of 11th Eurographics Workshop on Rendering. MIT Press, 1998.

- [Steinbichler Optotechnik GmbH 12] Steinbichler Optotechnik GmbH. Produktwebsite. [http://www.steinbichler.de/de/main/3d\\_digitalisierung.htm](http://www.steinbichler.de/de/main/3d_digitalisierung.htm). 2012.04.22.
- [Stricker 12] Didier Stricker. Orbital camera system produktwebsite, technische universität kaiserslautern, ag prof. didier stricker. [http://www.nek-kl.de/de\\_DE/produkte/orcam-orbital-camera-system/](http://www.nek-kl.de/de_DE/produkte/orcam-orbital-camera-system/). 2012.04.22.
- [Suppa 07] M. Suppa, S. Kielhofer, J. Langwald, F. Hacker, K.H. Strobl, G. Hirzinger. The 3d-modeller: A multi-purpose vision platform. Tagungsband: Robotics and Automation, 2007 IEEE International Conference on, Seiten 781 –787, april 2007.
- [Tenorth 10] Moritz Tenorth, Lars Kunze, Dominik Jain, Michael Beetz. KNOWROB-MAP – Knowledge-Linked Semantic Object Maps. Tagungsband: 10th IEEE-RAS International Conference on Humanoid Robots, Seiten 430–435, Nashville, TN, USA, December 6-8 2010.
- [The Ponce Group 12] The Ponce Group. Datasets for computer vision research (website). 2012.04.24.
- [Torralba 03] Antonio Torralba, Kevin P. Murphy, William T. Freeman, Mark A. Rubin. Context-based vision system for place and object recognition. Computer Vision, IEEE International Conference on, 1:273, 2003.

- [Torralba 04] A. Torralba, K. P. Murphy, W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. Tagungsband: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Band 2, Seiten 762–769, Washington, DC, June 2004.
- [Tsai 87] R Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal on Robotics and Automation, 3(4):323–344, 1987.
- [TurboSquid 12] TurboSquid. Website. 2012.04.24.
- [Ulbrich 11] S. Ulbrich, D. Kappler, T. Asfour, N. Vahrenkamp, A. Bierbaum, M. Przybylski, R. Dillmann. The opengrasp benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation. Tagungsband: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, Seiten 1761 –1767, sept. 2011.
- [Viola 01] Paul Viola, Michael Jones. Robust real-time object detection. Tagungsband: International Journal of Computer Vision, 2001.
- [Vogt 00] Stefan Vogt. Reduktion optischer 3D-Meßdaten mittels Multi-Resolution. Dissertation, Universität Karlsruhe (TH), Mainz, 2000.
- [Wächter 11] Mirko Wächter. Automatisiertes, modellbasiertes training und erkennung von 3d-objekten mit mehrdeutigen punktmerkmalen. Diplomarbeit, 2011.

- [Wikipedia, E. - User:Kolossos 12] Wikipedia, E. - User:Kolossos. <http://en.wikipedia.org/wiki/File:Kinect2-ir-image.png>. Wikimedia Commons. 04.03.2012, 2012.03.04.
- [Wikipedia, E. 12] Wikipedia, E. Artikel zu kinect <http://en.wikipedia.org/wiki/Kinect>. 28.02.2012, 2012.02.26.
- [Willow Garage 11] Willow Garage. Opencv wiki - camera calibration and 3d reconstruction. [http://opencv.willowgarage.com/documentation/camera\\_calibration\\_and\\_3d\\_reconstruction.html](http://opencv.willowgarage.com/documentation/camera_calibration_and_3d_reconstruction.html). 2011.10.08.
- [Willow Garage 12] Willow Garage. Ros household objects sql database. 2012.04.24.
- [Wohlers Associates 10] Wohlers Associates. 3d scanning & reverse engineering overview. <http://www.wohlersassociates.com/scanning.html>. 2010.10.15.
- [Xue 09a] Zhixing Xue, A. Kasper, J.M. Zoellner, R. Dillmann. An automatic grasp planning system for service robots. Tagungsband: Advanced Robotics, 2009. ICAR 2009. International Conference on, Seiten 1–6, june 2009.
- [Xue 09b] Zhixing Xue, P. Woerner, J.M. Zoellner, R. Dillmann. Efficient grasp planning using continuous collision detection. Tagungsband: Mechatronics and Automation, 2009. ICMA 2009. International Conference on, Seiten 2752–2758, aug. 2009.

- [Yao 09] B. Yao, Xiong Yang, Tianfu Wu. Image parsing with stochastic grammar: The lotus hill dataset and inference scheme. Tagungsband: Computer Vision and Pattern Recognition Workshops, 2009. CV-PR Workshops 2009. IEEE Computer Society Conference on, Seite 8, june 2009.
- [ZALEVSKY 07] Rosh Ha'ayin 48560 IL); SHPUNT Alexander (10/7 Berlovich Street Petach Tikvah 49742 IL); MAIZELS Aviad (14 Revivim Street Tel Aviv 69354 IL); GARCIA Javier (C/Rodriguez de Cepeda 48 1 Rodriguez De Cepeda Street Valencia E-46021 ES) ZALEVSKY, Zeev (1 Ha'Hermon Street. Method and system for object reconstruction. 2012.03.08.
- [Zhang 92] Zhengyou Zhang. Iterative point matching for registration of free-form curves. 2012.03.15.
- [Zhang 99] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. Computer Vision, IEEE International Conference on, 1:666, 1999.