

# Masterarbeit

Hosnia Najem

Kamerabasierte Suche von Objekten in einer  
Smart Home Umgebung

Hosnia Najem  
Kamerabasierte Suche von Objekten in einer Smart  
Home Umgebung

Masterarbeit eingereicht im Rahmen der Masterprüfung  
im Masterstudiengang Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Professoren :  
Prof. Dr. rer. nat. Kai von Luck  
Prof. Dr.-Ing. Gunter Klemke

Abgegeben am 23. Mai 2013

**Hosnia Najem**

**Thema der Arbeit**

Kamerabasierte Suche von Objekten in einer Smart-Home-Umgebung

**Stichworte**

Merkmale, Referenzmodell, SURF-Algorithmus, Objektsuche im Bild und Raum, Positionsbestimmung, sprachliche Beschreibung der Pose im Raum

**Kurzzusammenfassung**

Der Living Place Hamburg ist ein Laboratorium zur praktischen Erprobung von „Smart Home“-Techniken in einer realistischen Umgebung. Dort sind neben vielen anderen Sensorsystemen auch flexible Kameras installiert, deren Einbindung in das Gesamtsystem und erste Nutzung in dieser Arbeit in Angriff genommen wurde.

Als Testanwendung wurde ein Objektsuch-System mit Sprachausgabe ausgewählt. Dazu wurden die Grundlagen der Abbildung und Erkennung von Objekten sowie Anwendungsszenarien und -fälle studiert und ein Prototyp konzipiert und implementiert. Besondere Aufmerksamkeit wurde der Formulierung sprachlicher Mitteilungen an den Benutzer/Bewohner des Living Place Hamburg gewidmet. Der Prototyp wurde erfolgreich in die Zielumgebung eingebunden, getestet und erste Einstellungen wurden optimiert.

**Hosnia Najem**

**Title of the paper**

Camera based object search in a smart-home environment

**Keywords**

feature points, key points, SURF algorithm, object searching, object recognition, smart home, localization

**Abstract**

The Living Place Hamburg is a laboratory which provides a realistic environment to test smart home techniques. In addition other sensor systems it is equipped with cameras. This thesis intends to integrate these cameras into the existing smart home system and to develop a prototype application to use them.

The application chosen was an object search system with speech output. For that purpose the foundations of object representation and identification with cameras were studied and scenarios and use cases developed. A prototype was designed and implemented. Particular emphasis was given to the problem of verbalizing the responses. The prototypical application was integrated into the smart home environment and successfully tested. First improvements were possible.

## **Danksagung**

Wo ein Wille ist, ist auch ein Weg. Dies ist ein Satz, der mich seit langem begleitet.

Ich widme diese Arbeit meinen Eltern, die mir als Vorbild dienen und mir immer gezeigt haben, dass der Weg, den ich gehe, der richtige ist. Die mir immer Mut und Kraft gegeben und mich auf meinem Weg immer bestärkt haben. Ein ganz großer Dank geht an meine betreuenden Professoren, die mir unterstützend zur Seite standen. Nicht zu vergessen meine Studienfreunde und die Wegbegleiter in der Phase meines Masterstudiengangs, die meine Freude am Studium aufblühen ließen.

Das Ende ist der Beginn von etwas Neuem.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>8</b>
<b>Abbildungsverzeichnis</b>	<b>9</b>
<b>1. Einleitung</b>	<b>11</b>
1.1. Motivation der Masterarbeit . . . . .	13
1.2. Living Place Hamburg . . . . .	14
1.2.1. Aufbau des Living Place Hamburg . . . . .	15
1.2.2. Sensorik im Living Place Hamburg . . . . .	17
1.3. Ziel der Masterarbeit . . . . .	19
1.4. Gliederung . . . . .	20
<b>2. Verfahren der Objekterkennung</b>	<b>22</b>
2.1. Mögliche Bildmerkmale . . . . .	23
2.2. Merkmalsbasierte Objekterfassungs- und -modellierungsverfahren . . . . .	23
2.2.1. SIFT-Verfahren . . . . .	25
2.2.2. SURF-Verfahren . . . . .	30
2.2.3. Deskriptoren-Matching . . . . .	32
2.3. Positionsbestimmung im Raum . . . . .	34
2.3.1. Stereo Vision . . . . .	38
<b>3. Anforderungsanalyse</b>	<b>41</b>
3.1. Anwendungsszenarien . . . . .	41
3.1.1. Sals Morgen-Routine . . . . .	42
3.1.2. Sal vorm Schlafen gehen . . . . .	42
3.1.3. Sal hinterlegt das Buch im System . . . . .	42
3.2. Analyse der Szenarien . . . . .	43
3.2.1. Szenarien mit Suchanfrage an das System . . . . .	44
3.2.2. Szenario: Objekt im System hinterlegen . . . . .	53
3.3. Anforderungen an die Benutzeroberfläche . . . . .	57
3.4. Anforderungen . . . . .	59
3.4.1. Technische Anforderungen . . . . .	59
3.4.2. Qualitative Anforderungen . . . . .	60

---

3.4.3. Funktionale Anforderungen . . . . .	60
3.5. Related work / Aktuelle Forschungsprojekte . . . . .	71
3.5.1. Masterarbeiten mit der SURF-Implementierung . . . . .	71
3.5.2. Beziehung zwischen Raum und Sprache . . . . .	72
3.6. Fazit . . . . .	80
<b>4. Konzeption</b>	<b>82</b>
4.1. Systemarchitektur . . . . .	83
4.1.1. Präsentationsschicht . . . . .	84
4.1.2. Kamera-Komponente . . . . .	84
4.1.3. Datenverwaltung . . . . .	85
4.1.4. Anbindung an Living Place Hamburg . . . . .	86
4.1.5. Anwendungsschicht . . . . .	87
4.1.6. Training-Komponente . . . . .	88
4.1.7. Benutzeroberfläche: Training-GUI . . . . .	90
4.2. Subsystem Search: Orten von Objekt in Bildern . . . . .	95
4.2.1. Search-Komponente . . . . .	96
4.2.2. Benutzeroberfläche: Search-GUI . . . . .	98
4.3. Subsystem Location: Orten von Objekten im Raum . . . . .	101
4.3.1. Location-Komponente . . . . .	102
4.3.2. Sprachliche Beschreibung der Pose im Raum . . . . .	104
<b>5. Umsetzung</b>	<b>109</b>
5.1. Verwendete Tools und Bibliotheken . . . . .	109
5.1.1. Programmiersprache . . . . .	110
5.1.2. OpenCV . . . . .	110
5.1.3. SURF-Implementierung . . . . .	111
5.1.4. Microsoft Speech API . . . . .	112
5.2. Klassenaufbau . . . . .	112
5.3. Modellierung und Suche von Objekten . . . . .	114
5.3.1. Training/Teaching Objekt . . . . .	116
5.3.2. Test des SURF-Algorithmus . . . . .	116
5.3.3. Search/ Search Objekt . . . . .	120
5.4. Sprachliche Beschreibung der Pose . . . . .	122
5.4.1. Vereinbarte Konventionen . . . . .	125
5.5. Offene Realisierungen/Themen . . . . .	129
5.5.1. Lokalisierung des Objekts im Raum . . . . .	130
5.5.2. Die Kamerasteuerung . . . . .	131
5.6. Fazit . . . . .	133
<b>6. Schlussbetrachtung</b>	<b>135</b>

---

6.1. Zusammenfassung . . . . .	135
6.2. Ausblick . . . . .	136
<b>Literaturverzeichnis</b>	<b>138</b>
<b>A. Kameras im Living Place Hamburg</b>	<b>144</b>
A.0.1. Kamera Eigenschaften . . . . .	146
A.0.2. Spezifikation der Kamera . . . . .	148
A.0.3. Kamera – Funktionalitäten . . . . .	154
A.0.4. PTZ Konfiguration . . . . .	156

# Tabellenverzeichnis

3.1. Anwendungsfall: Service-Anfrage an das System stellen . . . . .	48
3.2. Anwendungsfall: Objektsuche in Bildern . . . . .	49
3.3. Anwendungsfall: Objekt anzeigen . . . . .	50
3.4. Anwendungsfall: Pose ermitteln . . . . .	51
3.5. Anwendungsfall: Position ansagen . . . . .	52
3.6. Anwendungsfall: Referenzbilder hinterlegen . . . . .	55
3.7. Anwendungsfall: Referenzmodell erstellen . . . . .	56
5.1. Zeitmessung bei der Extraktion der Referenzpunkte der Kamerabilder mit unterschiedlicher Einstellung des analogen Zooms [Quelle:Hosnia Najem] . . . .	118
5.2. Zeitmessung bei der Extraktion der Merkmale durch manuelle Berechnung der Auflösung eines Kamerabildes [Quelle:Hosnia Najem] . . . . .	120

# Abbildungsverzeichnis

1.1.	3-D Modell Living Place Hamburg [Quelle: ?]	14
1.2.	Living Place Hamburg Aufteilung[Quelle: <a href="#">Karstaedt (2012)</a> ]	15
1.3.	Living Place Hambur Kontrollraum [Quelle: Living Place Hamburg ]	16
2.1.	Ablauf einer Merkmalsextraktion[Quelle: <a href="#">Fries (2011)</a> ]	26
2.2.	Gaußpyramide	27
2.3.	Bestimmung der Extrema[Quelle: <a href="#">Bay u. a. (2006)</a> ]	28
2.4.	Merkmals-Orientierung [Quelle: <a href="#">Lowe (2004)</a> ]	29
2.5.	Bestimmung des Deskriptors auf Basis von Gradienten. Quelle: <a href="#">Lowe (2004)</a>	30
2.6.	SIFT vs. SURF [Quelle: <a href="#">Bay u. a. (2006)</a> ]	32
2.7.	Kameramodell [Quelle: <a href="#">Meisel (2012)</a> ]	35
2.8.	Abbildung der 3D zu 2D	36
2.9.	Kameraparameter im Weltkoordinatensystem [Quelle: <a href="#">Meisel (2012)</a> ]	37
2.10.	Achsen-parallele Anordnung [Quelle: <a href="#">Meisel (2012)</a> ]	39
2.11.	Epipolarebenen und Epipolarlinien Quelle:( <a href="#">Meisel, 2012</a> )	39
3.1.	Analyse der Szenarien	43
3.2.	Anwendungsdiagramm: Objektsuche im Raum	45
3.3.	Anwendungsdiagramm: Objekt im System hinterlegen	53
3.4.	Kameragestützte Suche des Buches	80
4.1.	Systemarchitektur in Gesamtüberblick	83
4.2.	Datenbank-Entwurf	86
4.3.	Anwendungs-komponente Binnenansicht	87
4.4.	Dynamischer Ablauf: Serviceanfrage verstehen	89
4.5.	Konzept Training-GUI: Referenzmodell erstellen	91
4.6.	Orten von Objekten in Bilder	95
4.7.	Dynamischer Ablauf : Orten in Bildern	97
4.8.	Interface Search-GUI	98
4.9.	Orten im Raum	101
4.10.	Ablauf Orten im Raum	105
4.11.	Interaktion zwischen Anwender, Kamerasystem und ISIS (erstellt von. <a href="#">Karstaedt, 2011a</a> )	106

---

4.12. Ablauf sprachliche Beschreibung . . . . .	107
5.1. Ausschnitt der Klassendiagramme für Surfmodel, Teaching-, Search Objekt . . . . .	113
5.2. Benutzerschnittstelle . . . . .	115
5.3. Referenzbilder mit den extrahierten Merkmalen . . . . .	119
5.4. Erstellung des Referenzmodells „Buch“ aus vier Perspektiven mit jeweils drei unterschiedlichen Auflösungen . . . . .	121
5.5. Wiederfinden des Buches anhand des Referenzmodells im Living Place Hamburg . . . . .	122
5.6. Möbel im Living Place Hamburg [Quelle: <a href="#">Karstaedt (2012)</a> ] . . . . .	123
5.7. Living Place Hamburg Objekt IDs [Quelle: <a href="#">Karstaedt (2012)</a> ] . . . . .	123
5.8. Kommunikation der Kamerasystem mit ISIS . . . . .	124
5.9. Applikation zur Steuerung der PTZ-Kamera [Quelle: Hosnia Najem] . . . . .	130
5.10. Systemarchitektur in Gesamtüberblick . . . . .	133
A.1. Die Kameras im Wohnraum: [Quelle: <a href="#">Karstaedt (2012)</a> ] . . . . .	144
A.2. Kameras Postion im Schlafbereich [Quelle: <a href="#">Karstaedt (2012)</a> ] . . . . .	145
A.3. Kameras vom Esszimmer [Quelle: <a href="#">Karstaedt (2012)</a> ] . . . . .	145
A.4. Vernetzung der Kameras im Living Place Hamburg . . . . .	147
A.5. Encoder-Einstellung in PTZ-Domekamas . . . . .	154
A.6. Bezeichnung der Auflösungen . . . . .	155
A.7. Kamera Steuerung über den Webbrowsers . . . . .	157
A.8. PTZ-Steuerung über die PView Software . . . . .	157

# 1. Einleitung

„Humans out of the loop“<sup>1</sup> lautet häufig die Devise von Wissenschaftlern und Entwicklern in dem heutigen digitalen Zeitalter, wo Computer immer mehr in den Hintergrund treten, aus dem Fokus verschwinden, immer kleiner, schneller und günstiger werden; wo sie sich immer mehr im Alltag integriert haben, in den Hintergrund rücken und im Verborgenen ihre Aufgaben erledigen, ohne dass der Mensch davon Kenntniss nimmt.

Mit zunehmender Komplexität der Funktionen und Technik in den Alltagsgegenständen, möchte der Mensch der Generation *Digital Natives*,<sup>2</sup> der von klein auf mit den Technologien des digitalen Zeitalters aufgewachsen ist, auch nichts davon merken, sondern deren Funktionalitäten nur nutzen. Die Digital Natives leben in eine Welt der Vernetzung, in der E-Mail, Internet, Mobiltelephone, *instant messages* und *social media networks* integrale Bestandteile ihres Leben sind.

Computer sind allgegenwärtig (ubiquitous computing) und „verschwinden“ gar (computers disappear), und die immer weiter fortschreitende Entwicklung der Informations- und Kommunikationstechniken in zahlreichen Bereichen unseres Lebens und die massive Interaktion und Vernetzung, die Möglichkeit, immer online zu sein, führen zu einer anderen Denkweise und neuen Denkmustern.

Während die *Digital Immigrants*, Menschen, die nicht von klein auf mit diesen Technologien aufgewachsen sind, E-Mails oder einen Text zur Bearbeitung noch ausdrucken müssen, sind es die Digital Natives gewohnt, Informationen schnell in digitaler Form zu empfangen und lieben es, alles digitalisiert zu haben, so dass es für sie von überall zugreifbar ist. Sie leben in einer Welt totaler Vernetzung, in dem Jeder mit Jedem und Alles mit Allem vernetzt ist.

Mit der zunehmenden Geschwindigkeit von Veränderungen in unserem Leben und der daraus entstehenden Komplexität unseres Alltags werden Lösungen gesucht, die den „heutigen“ Menschen im Alltag unterstützen. Die Informatik unternimmt wissenschaftliche Anstrengungen und kreiert Projekte, die sich mit computergestützten Anwendungen zur effizienteren Gestaltung unseres Alltags (zu Hause und im Beruf) beschäftigen, ausgedrückt in Begriffen wie *ubiquitous*, *pervasive* oder *mobile computing*.

---

<sup>1</sup>Referenzierte Quellen: [Mattern (2003a), Mattern (2003b), Bohn u. a. (2003)]

<sup>2</sup>Der Begriff *Digital Natives* wurde von Marc Prensky geprägt (Prensky, 2001a,b).

Der einflussreiche Aufsatz „The computer for the 21st century“ von Mark Weiser (1991), der damals als ein visionärer Artikel oder gar als Science-Fiction angesehen wurde, entpuppt sich heute in Wissenschaft und Forschung als treffende Vorausschau auf die Entwicklung der neuesten Technologien.

Weiser prägte damals den Begriff des „ubiquitous computing“, des allgegenwärtigen Computers, der dem Menschen im Alltag und Beruf unsichtbar und unaufdringlich im Hintergrund dient mit der Zielsetzung, ihn bei seinen Arbeiten und Tätigkeiten zu unterstützen bzw. ihn von lästigen Routineaufgaben weitestgehend zu befreien.

Nach Weiser sollte der PC als solcher verschwinden, seine informationsverarbeitende Funktionalität aber überall verfügbar sein. Er sah die Zukunft darin, dass der PC durch viele kleine Prozessoren und mikroelektronische Sensoren im Alltag ersetzt wird, die untereinander vernetzt sind und unsichtbar im Hintergrund agieren. Er verstand diese Technik als reines Mittel zum Zweck, das im Hintergrund bleiben soll, um die Aufmerksamkeit der Anwender auf die Sache an sich zu lenken und nicht zu stören.

As technology becomes more imbedded and invisible, it calms our lives by removing the annoyances . . . The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. [Want \(2000\)](#)

Michael Dertouzos, Direktor des Laboratory for Computer Science am MIT, meint jedenfalls, dass man bei der Verwendung von Ubiquitous-Computing-Techniken die humane Komponente in den Mittelpunkt stellen muss.

Ubiquity and pervasiveness are simply not the important drivers of change for tomorrow . . . If our technologists are driven by the prospects of pervasive computing alone, they are likely to continue along the familiar path that increases the complexity and inadequacy that users face, even though the designers never had this intent . . . The important quest for the balance of this century is to make our information systems human-centered.

Die Kritiker der „Ubiquitous Computing“-Theorien behaupten nun, es werde versucht, die reale Welt in eine computergesteuerte Welt zu verwandeln. Aber auch wenn das Paradigma des Ubiquitous Computing so erscheinen mag, dass versucht wird, die reale Welt in eine computergesteuerte Welt zu bringen, ist dem nicht so.

Bei der Vision des Ubiquitous Computing geht es gerade nicht darum, sich von der realen Welt abzukapseln und eine künstliche Welt aufzubauen, sondern im Gegenteil darum, unser Leben in der „einzig wahren“ Welt und der natürlichen, dem Menschen vertrauten Umgebung durch diskret in den Hintergrund tretende Technik angenehmer zu gestalten.

## 1.1. Motivation der Masterarbeit

Die Motivation dieser Masterarbeit besteht darin, in dem Living Place Hamburg , dem Smart-Home-Projekt an der Hochschule für Angewandten Wissenschaften Hamburg (HAW Hamburg) mitzuwirken, also daran mitzuwirken, das Smart Home an der HAW Hamburg smarter zu gestalten, die Alltagsgegenstände „smart“ zu machen; und insbesondere natürlich auch, an der Forschung und Entwicklung der neuesten Technologien in diesen Bereichen teilzuhaben.

Auf Grund einiger Erfahrung mit einer Bachelorarbeit im Bereich Computer Vision wurde diese Masterarbeit mit dem Titel „Kamerabasierte Suche von Objekten in einer Smart Home Umgebung“ gewählt.

Mark Weisers Vision dient als die Grundlage für die Forschung im Living Place Hamburg , dem Smart Home an der HAW Hamburg.

In ihm wird so die Idee und Vision von Mark Weiser verfolgt, die ihren Ursprung in den Theorien haben, die das junge Forschungsgebiet des „Ubiquitous Computing“ geprägt haben.

„Smart Devices“, „Smart Environments“, „Smart Living“ „Companion Systems“ sind nur einige Begriffe, die die im Living Place Hamburg entwickelten Technologien und Anwendungen begleiten und kennzeichnen. Laut Mark Weiser ist eine Anwendung smart (intelligent), wenn deren Gegenwart dem Anwender verborgen bleibt. Smart wird also verstanden als Kennzeichen intelligenter Alltagsgegenstände, die ihre Umgebung erfassen und den Anwender bei der Bewältigung ihrer Aufgaben intuitiv und unauffällig unterstützen, deren Gegenwart also i. d. R. dem Anwender verborgen bleibt.

Im Living Place Hamburg werden bereits seit langem Informations- und Kommunikationstechniken erprobt, um eine Umgebung zu schaffen, die sich adäquat an die Gewohnheiten der Bewohner und die aktuelle Situation in der Wohnung anpasst.

In den folgenden Abschnitten soll erklärt werden, wie der Living Place Hamburg aufgebaut ist, welche Technologien eingesetzt und wie die Allgegenwärtigkeit erreicht werden soll sowie welche Technologien bereits im Living Place Hamburg integriert sind, um darzustellen, wie sie durch diese Masterarbeit ergänzt werden.

## 1.2. Living Place Hamburg



Abbildung 1.1.: 3-D Modell Living Place Hamburg [Quelle: ?]

Es stellt eine Laborumgebung für Realexperimente im Bereich intelligente Wohnung dar. Es ist eine Musterwohnung auf der Gelände der HAW Hamburg, in der Ergebnisse insbesondere aus den Forschungsbereichen „Ubiquitous Computing“ und „Ambient Intelligence“ implementiert werden.

Das Labor „Wohnen der Zukunft“ wurde im Oktober 2011 eröffnet. Es wird von der Wirtschaftsbehörde gefördert und dient als eine Kooperationsplattform für die Forschung und die Wirtschaft.

Bereits vor der Gründung des Living Place Hamburg wurden im Rahmen vieler Projekte und Abschlussarbeiten in verschiedenen Forschungslaboren die verschiedensten Systeme und Technologien unter dem Aspekt des Ubiquitous Computing entwickelt und später in den Living Place Hamburg integriert.

Im Living Place Hamburg wird von den Studierenden der Fachbereiche Informatik und Design versucht, die Paradigmen des Ubiquitous Computing, „Human Computing“, „Ambient Intelligence“, „Smart Environment“, „Smart Living“ und „Companion-Systeme“ in die Realität umzusetzen. So soll die Frage geklärt werden, wie Wohnen in der Zukunft aussieht und wie und welchen Einfluss die Technik nehmen wird.

Die Vorgehensweise ist es dabei, alle möglichen Systeme zu entwickeln und einzusetzen, um unter realen Bedingungen Forschungsfragen zu klären wie z .B.:

- Bieten Anwendungen und Services den Bewohner Unterstützung oder bevormundet die neue Technologie den Bewohner und stellt ihn vor vollendete Tatsachen, ohne ihm eine Auswahlmöglichkeit zu geben?
- Welche der Anwendungen stören im Alltag und welche bieten eine Unterstützung?
- Wo kann dem Bewohner der Alltag erleichtert werden und den Anwendungen mehr Verantwortung übertragen werden?
- Auch Forschungsfragen wie z. B., inwieweit der Mensch die intelligenten Anwendungen akzeptiert, können durch Usability-Tests in dieser Umgebung analysiert und erforscht werden. Denn laut Mattern (2003) setzt nicht die Umsetzung einer Anwendung Grenzen, sondern die Akzeptanz durch den Menschen.

In Folgenden wird erklärt, welche Konzepte und Technologien eingesetzt werden, um „Intelligenz in den Anwendungen“ zu schaffen und den Living Place Hamburg somit smarter zu gestalten und ein Überblick über die bereits integrierten und implementierten Anwendungen im Living Place Hamburg gegeben.

### 1.2.1. Aufbau des Living Place Hamburg

Das Living Place Hamburg repräsentiert einen Ein-Personen-Haushalt. Die Musterwohnung ist ein 140 m<sup>2</sup> Loft, in dem die Forschungsergebnisse implementiert werden.

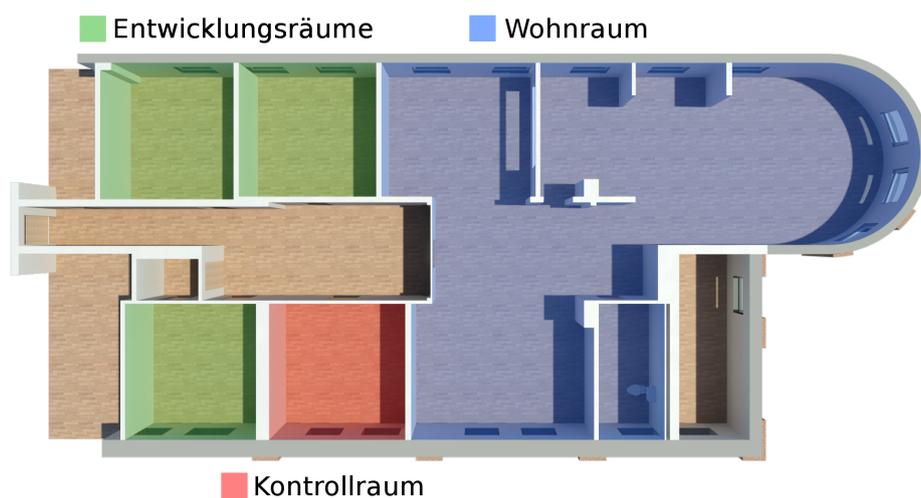


Abbildung 1.2.: Living Place Hamburg Aufteilung[Quelle: [Karstaedt \(2012\)](#) ]



Abbildung 1.3.: Living Place Hambur Kontrollraum [Quelle: Living Place Hamburg ]

Das Living Place Hamburg enthält einen Kontrollraum, den Wohnraum und drei Entwicklungsräume, die im 2-D-Gebäudemodell in der Abbildung 1.2 farbig markiert dargestellt sind. Der Wohnbereich ist unterteilt in den Bereiche Küche und Essbereich, Wohnbereich und Schlafbereich, in denen jeweils eine Kamera montiert ist. Die Eigenschaften und Funktionalitäten der integrierten Kameras im Living Place Hamburg werden im Anhang A genannt.

Der Kontrollraum, dargestellt im der Abbildung 1.3, ist das Herzstück der Wohnung. Von hieraus wird die eingesetzte Technik in der Wohnung gesteuert und überwacht.

Der Vorteil der Living Place Hamburg gegenüber andern Smart Homes liegt darin, dass er einen komplett funktionierenden Haushalt darstellt. Die Wohnung ermöglicht es so Personen in realistischen Experimenten mehrere Tage auf Probe zu leben. So können die entwickelten Ideen, Konzepte und Techniken in einer Alltagsumgebung getestet werden.

Durch die Vielzahl integrierter Sensor-Technologien im Living Place Hamburg ist ein Sensor-Netzwerk entstanden, das in manchen Arbeiten bereits als „Sensor Wolke“ bezeichnet wird.

Im Living Place Hamburg gibt es bereits viele Anwendungen, die einen Service für den Bewohner zur Verfügung stellen. Die Kommunikation der Anwendungen miteinander findet durch Austauschen von Informationen über die Umgebung und den Bewohner statt, die sie über ihre Sensoren wahrnehmen.

Da sich in der Regel die Komponenten in großen Sensor-Netzwerken nicht untereinander kennen, aber ihre Informationen anderen zur Verfügung stellen wollen, ist auch eine Kommunikations-Architektur integriert, nämlich ein ActiveMQ Message-Broker [vgl. [Otto und Voskuhl \(2010\)](#)], der es den Komponenten ermöglicht, ihre Daten im Netz zur Verfügung zu stellen und Informationen untereinander auszutauschen, ohne sich zu kennen. Jede Anwendung stellt so ein eigenständiges System dar und ist von den übrigen Systemen entkoppelt.

### **1.2.2. Sensorik im Living Place Hamburg**

Die Wohnung und das Mobiliar ist mit einer Vielzahl an Sensoren ausgestattet, welche in verschiedenen Gegenständen und Geräten wie z. B. Audio- und Videosystemen, Single- und Multitouch-Tischen und vielen weiteren Gerätschaften integriert sind. Dabei kommen sowohl aktive als auch passive Sensoren zum Einsatz.

Nachdem eine adäquate und optimale Kommunikationsschnittstelle im Living Place Hamburg integriert wurde, konnten die einzelnen Anwendungen durch den Austausch von Informationen neue Services im Living Place Hamburg anbieten und bestehende Dienste smarter gestalten.

#### **Ubisens Sensoren**

Im Kontext der Living Place Hamburg Projekte wurden die Ubisens Sensoren im Living Place Hamburg angebracht. Es handelt sich um RFID-Tags (Transponder), die in einem bestimmten Zeitintervall Informationen über ihren Ort in Form einer 3D-Koordinate an den Ubisens-Empfänger senden. Jedem Tag kann eine Funktionalität zugeordnet werden, um so eine Aktion auszulösen.

Die Integration der Ubisens-Technik eröffnete neue Möglichkeiten der Integration von neuem Ubiquitous Computing oder smarten Anwendungen im Living Place Hamburg. Mit den Ubisens-Sensoren entwickelte Sören Voskuhl im Rahmen seiner Masterarbeit (vgl. [Voskuhl, 2011](#)) eine Anwendung zur Analyse des Verhaltens der Bewohner.

Die Wohnung „weiß“ auf Grundlage einer solchen Verhaltensanalyse des Bewohners, wie lange er in der Regel im Bad benötigt, welche Gewohnheiten der Bewohner im Alltag hat, usw.

## Lichtsteuerung auf Basis der Tags

Die Lichtsteuerung im Living Place Hamburg ist ein weiteres Anwendungsgebiet, bei dem auf Basis von Sensor-Daten bestimmte Lichteinstellungen in einem bestimmten Bereich der Wohnung vorgenommen werden. Der Benutzer ordnet einem (Ubisens-)Tag eine Farbe zu. Sobald der Software-Agent der Lichtsteuerung die Position des Tags erhalten hat, nimmt der Raum, in dem er sich befindet, die dem Tag zugeordnete Farbe an.

## Kapazitiver Sensor im LP

Im Rahmen der Masterarbeit, „Das intelligente Bett – Sensorbasierte Detektion von Schlafphasen“ (vgl. [Hardenack, 2011](#)) wurde das „intelligente Bett“ implementiert. Anhand von kapazitiven Sensoren im Bett wird analysiert, wie der Bewohner geschlafen hat und ob er bereits das Bett verlassen hat oder noch im Bett liegt. Diese Informationen werden dem Sensor-Netzwerk zur Verfügung gestellt.

In Kooperation mit dem Studiengang *Bekleidung – Technik und Management* am Department Design wurde eine Anwendung implementiert, die es ermöglicht, die Sitzposition des Bewohners auf dem Sofa zu analysieren (vgl. [Weiss \(2011\)](#), [Dreschke \(2011\)](#)), so kann man feststellen, ob der Bewohner auf dem Sofa liegt oder sitzt. Dabei erfassen die kapazitiven Sensoren auch die Bewegungen des Bewohners auf dem Sofa. Auf Basis der kapazitiven Sensordaten und weiterer Information aus dem Sensor-Netz können dann sogar Annahmen darüber getroffen werden, ob der Bewohner im Wohnbereich relaxt oder arbeitet.

Jedes dieser Geräte sendet Zustandsinformationen in das Sensor-Netzwerk. Auf Basis der gesammelten Sensorinformationen werden bestimmte Aktionen ausgeführt, um die Wohnung adäquat an die Bedürfnisse des Bewohners anzupassen.

Die oben genannten integrierten Technologien und daraus entstanden Ubiquitous-Computing-Anwendungen im Living Place Hamburg sind nur einige Beispiele dafür, wie die Kommunikation zwischen den Anwendungen stattfindet und bestimmte Services dem Bewohner zur Verfügung gestellt werden.

### 1.3. Ziel der Masterarbeit

Ausgangspunkt sind die im Living Place Hamburg integrierten Full High-Definition (HD) Pan-Tilt-Zoom (PTZ) High Speed (HS) Dome-Kameras der Firma Dallmeier,<sup>3</sup> die als die optischen Sensoren in der intelligenten Wohnung fungieren. Sie sollen nun in das Sensor-Netzwerk integriert werden, bildlich ausgedrückt: der Wohnung Augen verleihen.

Die Vision im Hintergrund dieser Arbeit liegt also darin, die integrierten sensorbasierten Information- und Kommunikationstechnologien des Smart Homes um die optischen Sensordaten zu ergänzen. Anwendungen sollen Kontextanalysen auf Basis von optischen Sensoren durchführen können, mit der Zielsetzung, die Services durch die Ergebnisse der optischen Sensoren adaptiv und adäquat, dem Kontext entsprechend an die Bedürfnisse der Bewohner im Living Place Hamburg anzupassen. Somit sollen die aus der Integration der Kameras im Living Place Hamburg gewonnenen Informationen anderen Anwendungen für ihre Zwecke zur Verfügung gestellt werden.

Auf der Grundlage dieser Anforderung, die bereits entwickelten Technologien des Living Place Hamburg mit einem System auf Basis der integrierten Kameras zu erweitern, erschließt sich auch das Ziel dieser Masterarbeit, nämlich die Suche nach Gegenständen zu unterstützen.

Um ein Konzept dafür zu erarbeiten, wie dies mit den montierten Kameras ermöglicht werden kann, wurden Szenarien herangezogen, bei denen eine Objektsuche mit diesen Kameras durchgeführt werden soll, und um die Umsetzbarkeit des Konzepts zu zeigen, wird ein Prototyp entwickelt.

**Abgrenzung** Hier wird aufgeführt was diese Masterarbeit nicht umfasst.

- Diese Masterarbeit soll nur den Grundstein für weitere Arbeiten darüber legen, wie ein Service mit den montierten Kameras für den Bewohner des Living Place Hamburg integriert wird.
- Sie soll analysieren, welche Grundvoraussetzungen geschaffen sein müssen, um die optischen Sensordaten im Living Place Hamburg in einen Service zu integrieren. Dies soll nur ein Konzept darstellen, wie die optischen Sensoren im Living Place Hamburg integriert werden können.
- Sie soll nur einen Service anbieten, der den Bewohner bei der Suche nach beliebigen Objekten mit den Kameras unterstützt. Wie dieser Service mit den bereits integrierten Services zusammenspielen soll, wird in den im Abschnitt 3.1 aufgeführten Szenarien erläutert.

---

<sup>3</sup>Quelle: ?

- Das in dieser Arbeit entwickelte System wird keine Schnittstelle anbieten, um die Kameras in der Wohnung zu steuern, und auch nicht, um die Kamerasteuerung für das gesamte Living Place Hamburg zur Verfügung zu stellen.
- Es bietet keine Lösung an, um eine Kontextanalyse auf Basis der optische Sensoren durchführen zu können; es wird auch keine Lösung anbieten, wie auf Basis der optischen Sensoren bestimmte Kontextanalysen stattfinden könnten.

Die Spracherkennung ist nicht Gegenstand dieser Masterarbeit. Die Einschränkungen, die sich daraus ergeben, sind folgende:

- Es wird eine System-Architektur für die Integration einer Spracherkennung im Rahmen der Living Place Hamburg vorgestellt. Eine prototypische Anwendung für die Spracherkennung wurde hierfür entwickelt.
- Im Kontext des Living Place Hamburg wurde bereits die Einbindung einer Sprachsteuerung in alle Aspekte der Interaktion mit dem Smart-Home in der Masterarbeit „Kontextabhängige multimodale Interaktion mit Schwerpunkt Spracherkennung im Smart-Home Umfeld“ (vgl. Witt, 2011) vorgestellt. Diese Umsetzung kann genutzt werden, um die Spracheingabe zu gestalten. Es wird aber nicht näher auf die Spracherkennung eingegangen.

## 1.4. Gliederung

Das Ziel der Masterarbeit soll in mehreren Schritten erreicht werden:

Zuerst werden die Grundlagen von Objekterkennung und Positionsbestimmung geklärt, wie in Kapitel 2 beschrieben. Dann wird ein Blick auf Vision und Kontext geworfen und erste Anforderungen in Abschnitt 3.4 abgeleitet, bevor dann im folgenden Abschnitt 3.1 die Szenarien vorgestellt und daraus die Anwendungsfälle abgeleitet werden.

Darauf gestützt können die Anforderungen selbst in Abschnitt 3.3 dargestellt werden.

Bei dieser Arbeit wurde auf zahlreiche Projekte zurückgegriffen, die an der HAW Hamburg in diesem Zusammenhang durchgeführt wurden, und über die Abschnitt 3.5 einen Überblick geben soll. Nach einem Fazit der Analyse in Abschnitt 5.6 wird das Konzept für eine Pilotanwendung in Kapitel 4 entwickelt. Dafür wird eine Systemarchitektur entworfen, wie in Abschnitt 4.1 zu sehen ist, um dann die Subsysteme Training (??), Searching (4.2) und Location (4.3) abzuhandeln, einschließlich der Modellierung der räumlichen Situation und deren Verbalisierung.

Die Umsetzung wird in Kapitel 5 geschildert. Neben Überlegungen zur Programmiersprache, den zur Verfügung stehenden Bibliotheken und Details zu den einzelnen Komponenten

umfasst sie auch Ergebnisse erster Versuche zur Ermittlung günstiger Parameter. Zusammengefasst wird dies in Abschnitt 5.6, nicht ohne darzulegen, welche Entwicklungen noch nicht möglich waren und warum, in Abschnitt 5.5.

Zu guter Letzt findet sich noch eine abschließende Diskussion des Erreichten in Kapitel 6.

## 2. Verfahren der Objekterkennung

Diese Arbeit beschäftigt sich mit der Erkennung von Objekten in Bildern und damit verbunden der Bestimmung der „Pose“ oder räumlichen Lage des Objektes im Raum, d. h. der Kombination von Position und Orientierung im dreidimensionalen Raum. Anhand bekannter Positionen von (Pass-)Marken im Raum kann die Pose einer Kamera bestimmt werden. Umgekehrt kann mit Hilfe von Passmarken, die einen Gegenstand definieren, dessen Position mit einer Kamera bestimmt werden.. in Bildern zu erkennen und deren Pose im Raum zu ermitteln.

In dieser Masterarbeit wird davon ausgegangen , dass sich die Objekte nicht bewegen. Aus diesem Grund liegt das Augenmerk nicht auf dem Verfolgen von Objekten in Bildern, dem sogenannten Tracking, sondern auf der Posenbestimmung der Objekte in einem Bild.

Im Abschnitt 2.3 wird skizziert, was an mathematischen Grundlagen benötigt wird, um einen Raumpunkt zu bestimmen. Es werden auch gängige Verfahren der Photogrammetrie angeführt, mit denen man den zu einem Bildpunkt passenden Raumpunkt bestimmt. Zunächst aber geht es darum, wie man einen Punkt in einem Bild überhaupt identifizieren kann.

Zur Bestimmung der räumlichen Lage eines Objektes gibt es verschiedene Methoden und Verfahren, die sich zunächst auf Grund der eingesetzten Sensortechniken unterscheiden, wie z. B. magnetische, akustische oder auch funkbasierte Sensoren. In dieser Masterarbeit dienen Kameras als optische Sensoren im Living Place Hamburg . Somit kommen die bildbasierten Methoden zum Einsatz.

Grundsätzlich berechnen die bildbasierten Verfahren eine Pose anhand von Punktkorrespondenzen in perspektivisch unterschiedlichen Bildern. Dabei unterscheiden sich die Verfahren darin, ob sie künstliche oder natürliche Bildmerkmale verwenden, sowie in den Methoden der Korrespondenzsuche.

Bei den bildbasierten Methoden unterscheidet man markerbasierte und markerlose Verfahren und solche, die eine oder mehrere Kameras verwenden.

## 2.1. Mögliche Bildmerkmale

Bei markerbasierten Verfahren wird zur Positionsbestimmung von Objekten in der realen Umgebung mit speziellen Markierungen gearbeitet, die allgemein Marker genannt werden. Diese Marker können noch weiter anhand des Informationsgehalts und der verwendeten Codierungen unterteilt werden. In der Regel haben sie die Form eines Quadrates. In der Mitte befindet sich ein eindeutiges Muster, um sie identifizieren und ihre Ausrichtung bestimmen zu können. Ein besonderes Merkmal dieser Markierungen ist, dass sie besonders gut durch Bildanalyseverfahren extrahiert werden können und dass sich ihre Position anhand ihrer speziellen Form sehr leicht bestimmen lässt.

Um ein Objekt zu erfassen, muss man die Marker in der Szene platzieren. Die Szene wird dann mitsamt den Markern aufgenommen. Aufgrund der bekannten Koordinaten der Marker im Raum und im Bild kann dann die Position des erfassten Objektes berechnet werden.

Die Verwendung von Markern ist aber in dieser Masterarbeit unerwünscht. Zu einen, da diese die natürliche Umgebung des Living Place Hamburg stören würden, zum anderen weil das Anbringen der Marker in der Umgebung an den Objekte mit einem hohen Aufwand verbunden wäre. Es müsste ja jedes Objekt mit einem eindeutigen Marker versehen werden.

Jede Kameraaufnahme enthält nun abhängig von Aufnahme und Auflösung eine Vielzahl von Bildinformationen, bei dem einige Punkte und Informationen eine Art von natürlichem Marker darstellen. In dieser Masterarbeit soll also die Posenbestimmung mit solchen merkmalsbasierten Verfahren stattfinden.

## 2.2. Merkmalsbasierte Objekterfassungs- und -modellierungsverfahren

Merkmale können markante Punkte oder auch Kanten von Objekten in Form von Linien, Kurven, Ecken, Kanten, Flächen, Konturen der Nachbarschaftsinformationen bzgl. eines Pixels sein. Das Ziel dieser Masterarbeit ist es, ein Verfahren aus dem Bereich der Computer Vision (Bildverarbeitung und Bildbearbeitung) zu verwenden, das aus den Bildinformationen die Merkmale eines Objektes aus dem Bild extrahiert. Die Merkmale sollen das Objekt eindeutig identifizieren können und das Objekt soll zugleich einen hohen Wiedererkennungswert aufweisen.

In dieser Masterarbeit findet die Posenbestimmung mit merkmals- oder feature-basierten Verfahren statt. Dies sind markerlose Verfahren, die also auf die Anbringung von künstlichen Markern in der Umgebung verzichten. Sie extrahieren stattdessen Merkmale eines Objektes

aus den Bildern. Solche Merkmale werden in der Literatur auch als Feature- oder Keypoints bezeichnet.

Im Gegensatz zur bildbasierten Ansätzen und Verfahren, wird im modellbasierten Ansatz aus den extrahierten Bildmerkmalen des Objekte ein Modell erstellt; anstatt ganzer Bilder werden also nur bestimmte Bildmerkmale des Objektes gespeichert. Mit Hilfe des Modelles werden dann korrespondierende Punkte in Bildern gefunden.

Die anfängliche Kritik an dem modellbasierten Ansatz, dass dieser nämlich rechenintensiver sei als der bildbasierte Ansatz, hat angesichts des Fortschritts bei der Entwicklung und Verbesserung von Bildbearbeitungsverfahren keinen Bestand gehabt. Mittlerweile existieren Implementationen, die Echtzeit-Ansprüchen genügen.

Weiterhin hat der modellbasierte Ansatz im Gegensatz zum bildbasierten Ansatz die Vorteile robust gegenüber Verdeckung zu sein, viel weniger Speicherplatz für die Bildinformationen zu benötigen und eine genauere Bestimmung der Position des Objektes im Raum zu ermöglichen.

Die Problematik dieses Ansatzes liegt in der eindeutigen Zuordnung von Merkmalen, dem sogenannten Korrespondenzproblem. Zahlreiche Forschungsergebnisse haben jedoch gezeigt, dass eine falsche Zuordnung von natürlichen Markern mit adäquatem Berechnungsaufwand fast immer vermieden werden kann [Lowe (2004),Lowe, Hoffmann (2009)].

Die skaleninvarianten Merkmalsextraktion (engl. Scale-Invariant Feature Transform, kurz SIFT) ist eines der am meisten verwendeten Verfahren der Computer Vision zum Auffinden von Merkmalen, die invariant bezüglich Rotation, Translation, Skalierung, Änderung der Lichtverhältnisse und teilweiser affiner Verzerrung in Bildern sind. Es wurde erstmals 1999 von David Lowe in (Lowe (1999)) veröffentlicht, in der Veröffentlichung [Lowe (2004)] wurde das Verfahren optimiert.

SIFT war einer der ersten featurebasierten Algorithmen. Das Besondere des SIFT-Verfahrens ist seine Fähigkeit, Objekte in Bildern unabhängig von der Umgebungsbeleuchtung, ihrer Lage oder ihrer Skalierung wiederzufinden. Der Algorithmus kommt freilich in seiner ursprünglichen Version nicht in vielen Anwendungen zum Einsatz. Zum einen, weil er von Lowe patentiert wurde, aber vor allem, weil das Verfahren beim Wiedererkennen der Merkmale in Bildern sehr rechenintensiv ist.

Es wurden viele Anstrengungen unternommen, andere Algorithmen mit gleichen Funktionalitäten und Eigenschaften zu entwickeln, um ähnlich gute Ergebnisse bei geringerer Laufzeit zu erhalten. Einer davon ist das SURF-Verfahren. Es hat sich wegen seiner Schnelligkeit und Zuverlässigkeit als bestens bewährter Suchalgorithmus zur Wiedererkennung von Objekten etabliert und findet in vielen Bereichen seinen Einsatz.

Die Arbeitsweise der SIFT- und SURF-Algorithmen gliedert sich in vier Schritte.

- Im ersten Schritt werden potentielle skalierungsinvariante markante Punkte (Keypoints) in den Bildern ermittelt.
- Anschließend werden die Keypoints auf Stabilität geprüft, um daraus die potentiellen Merkmale zu bestimmen.
- Im dritten Schritt wird für jeden markanten Punkt (Keypoint) die Hauptorientierung berechnet.
- Im letzten Schritt wird zu jedem Keypoint ein Merkmalsvektor (Deskriptor) ermittelt.

Der genaue Ablauf der beiden Verfahren zur Merkmalsextraktion wird im Folgenden vorgestellt.

### 2.2.1. SIFT-Verfahren

Nachfolgend werden die vier Schritte des SIFT-Algorithmus ausführlich beschrieben. Die genaue mathematische Berechnung der aufgeführten Methoden und Verfahren findet sich in der Veröffentlichung von [Lowe (1999)] und [Lowe (2004)] und auch in verschiedenen Arbeiten von [vgl. Brauer (2010), Fries (2011)];].

**Ermittlung potentieller Merkmale** Aus dem Eingangsbild sollen markante Punkte, die sogenannten key points extrahiert werden, das sind die Bildpunkte, die eine Ecke, Kante oder T-Kreuzung eines Objekts darstellen. Dazu wird eine sogenannte Gaußpyramide gebildet.

**Gaußpyramide** Die Berechnung der Gaußpyramide erfolgt in folgenden Schritten:

- Beim ersten Schritt wird ein Gauß-Filter auf das Eingangsbild angewandt. Dadurch wird das Eingangsbild geglättet und die Detailinformationen des Eingangsbildes reduziert. Damit wird das Bildrauschen vermindert; kleinere Strukturen gehen verloren, wohingegen größere erhalten bleiben.
- Im zweiten Schritt wird das geklärte Eingangsbild auf die Hälfte der Bildgröße reduziert.
- Anschließend wird auf das reduziertes Bild der Gauß-Filter erneut angewendet. Diese liefert die zweite Stufe der Pyramide.

Aufgrund der vorherigen Gaußglättung beinhaltet jedes Pixel Details von seinen Nachbarpixeln. Dadurch muss beim Verkleinern der Bilder keine zusätzliche Pixelinterpolation mehr durchgeführt werden.

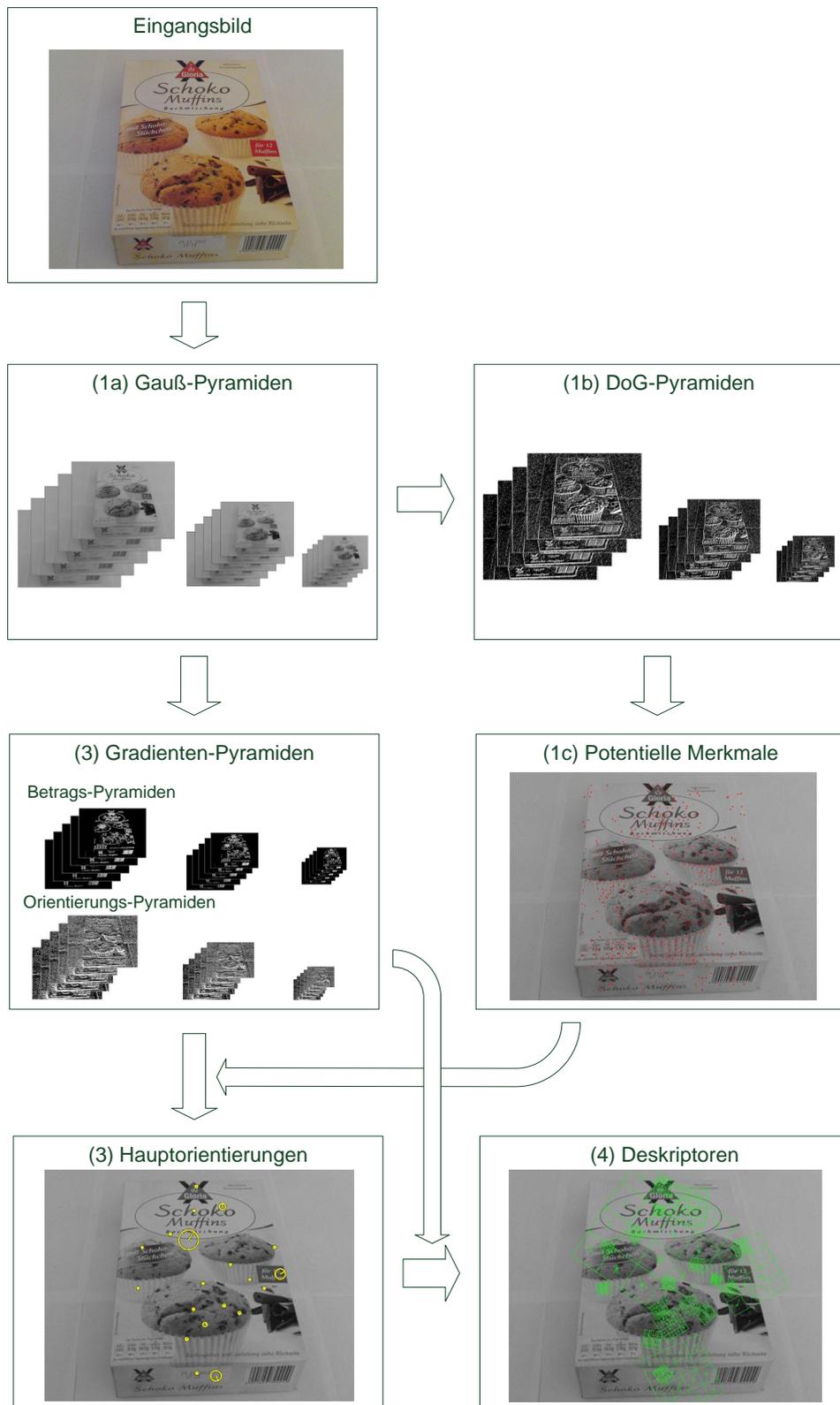


Abbildung 2.1.: Ablauf einer Merkmalsextraktion[Quelle:Fries (2011)]

- Nach mehrfacher Anwendung der Gaußglättung ergibt sich die sogenannte Gaußpyramide. Die Reduktion der Bildinformationen (somit auch die Größe der Gaußpyramide) ist abhängig von der Anwendung und kann selbst bestimmt werden.

Durch die Verwendung von Gaußpyramiden wird die Rechenzeit deutlich reduziert und das Verfahren erlangt neben Laufzeitverbesserungen auch Skalierungsinvarianz. In der Abbildung 2.1 ist die Gauß-Pyramide in 1a dargestellt.

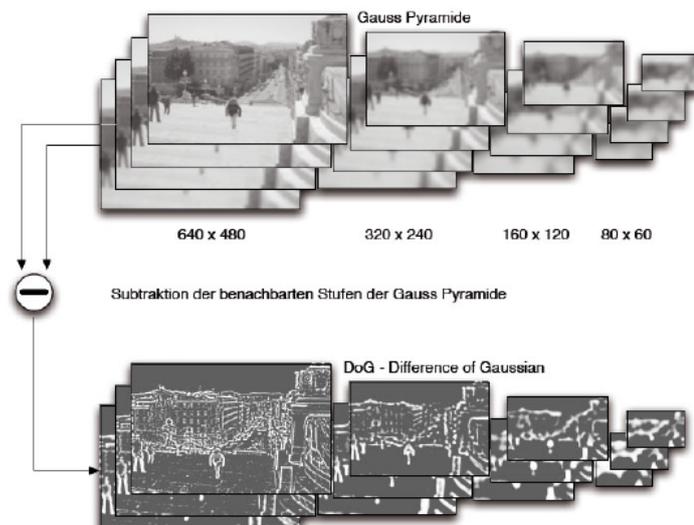


Abbildung 2.2.: Gaußpyramide

**DOG-Pyramide** Aus der Gaußpyramide wird nun die DOG-Pyramide berechnet, indem auf jeder Ebene (Oktave) der Gaußpyramide zwei benachbarte Gaußbilder voneinander subtrahiert werden. Das Ergebnis der Subtraktion wird als Difference of Gaussian (DOG) bezeichnet und die Zusammenfassung aller Stufen der DOG-Bilder als die DOG-Pyramide. In der Abbildung 2.1 (1b) und Abbildung 2.2 wird sie dargestellt.

Die Extremstellen dieser Bilder werden dann als Merkmalskandidaten, d. h. potentielle Merkmale angesehen. Für jedes Bild der DOG-Pyramide wird dafür ein Pixel mit seinen Nachbarn verglichen, acht Nachbarn auf der gleichen Stufe, neun Nachbarn in der darüber und darunter liegenden Stufe. Weist dieses Pixel ein echtes Maximum oder ein echtes Minimum auf, ist es ein Kandidat für einen Keypoint.

Ein Maximum ist gefunden, wenn alle benachbarten Pixel kleiner als es selbst sind. Ein Minimum ist im umgekehrten Fall gefunden. Die Grafik 2.3 veranschaulicht den Vergleich des Pixels  $m$  mit seinen Nachbarn.

Der Kontrast der Extrema (Minima, Maxima) wird durch Differenz mit der Nachbarschaft bestimmt. Liegt der Kontrast unter einer bestimmten Schwellenwert, wird dieses Extremum verworfen. Weiterhin werden solche Extrema nicht weiter berücksichtigt, deren Position auf der Kante infolge von Rauschen nicht eindeutig bestimmt werden kann.

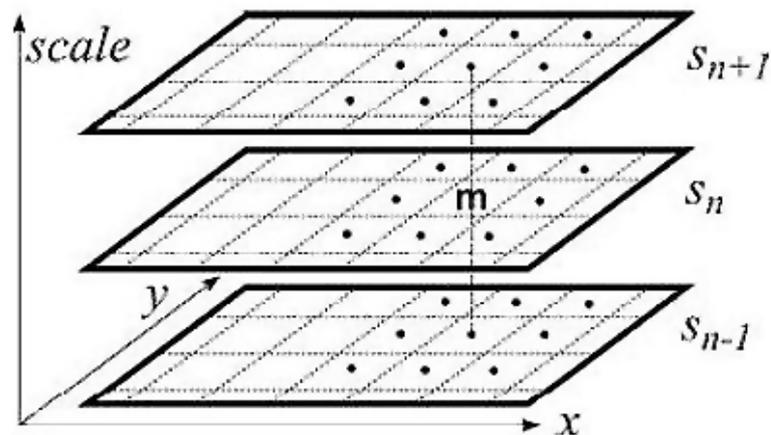


Abbildung 2.3.: Bestimmung der Extrema[Quelle:Bay u. a. (2006)]

Die potentiellen Keypoints werden anschließend einer Stabilitätsanalyse unterzogen. Erfüllt ein Keypoint die festgelegten Stabilitätsbedingungen, ist noch seine exakte Position zu bestimmen. Anderenfalls wird er verworfen.

Stabilitätsmerkmale sind unter anderem Kontrast (Rauschunterdrückung), Klassifikation in Ecken, homogene Regionen und Kanten. Denn da die Umgebungen unterschiedlicher Punkte einer Kante, einer Ecke oder einer homogenen Region sich relativ wenig unterscheiden, würden die errechneten Deskriptoren gleich sein, deshalb sollen diese Punkte verworfen werden. Durch einen vordefinierten Schwellenwert wurden die Keypoints mit geringem Kontrast bereits beim Aufbau der DOG-Pyramide eliminiert.

Ein Kantenpunkt kann anhand seiner zwei Hauptkrümmungen erkannt werden. Jeder Kantenpunkt besitzt eine große Hauptkrümmung quer über die Kante. Die zweite Hauptkrümmung verläuft entlang der Kante und ist deswegen sehr klein. Anhand des Verhältnisses zwischen den beiden Hauptkrümmungen wird also ermittelt, ob sich ein Punkt auf einer Kante befindet. Dafür wird mit Hilfe einer  $2 \times 2$ -Hesse-Matrix das Verhältnis der Hauptkrümmungen berechnet. Hierfür wird die  $2 \times 2$ -Hesse-Matrix zu den Keypoints aufgestellt und analysiert. Dabei muss das Verhältnis des größeren Eigenwerts zum kleineren unter einem bestimmten Schwellenwert liegen.

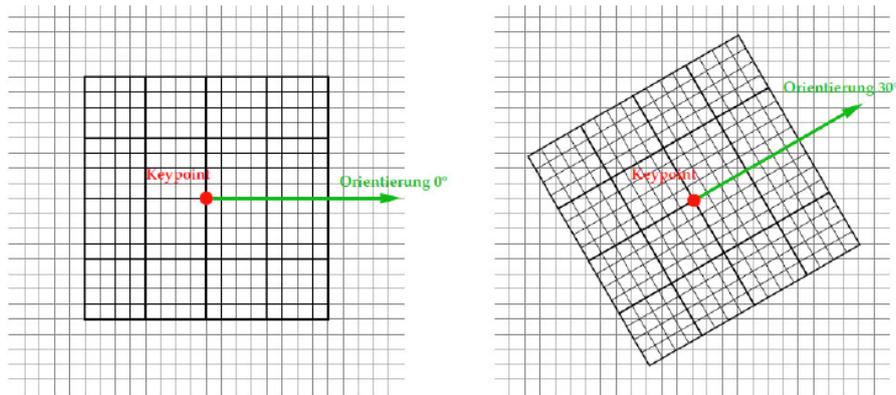


Abbildung 2.4.: Merkmals-Orientierung [Quelle: Lowe (2004)]

**Positionsbestimmung** Nachdem die Keypoints aus der DoG-Pyramide bestimmt sind, werden die exakte Position des Keypoints und die Größe jedes Pixels ermittelt. Dies geschieht mittels Subpixellokalisierung anhand der DOG-Pyramide. Die genaue Vorgehensweise der Subpixellokalisierung ist in [Lowe (2004)] aufgeführt.

**Berechnung der Hauptorientierung** Ein Keypoint ist in SIFT als ein markanter Punkt  $(x, y)$  in einem Bild definiert. Wie jedes Merkmal (Feature) wird er durch seine Bildposition, seine Hauptorientierung und einen Merkmalsvektor bzw. Deskriptor gekennzeichnet.

Der Deskriptor wird relativ zur Hauptorientierung berechnet. Wird ein Bild beispielsweise rotiert, dann haben übereinstimmende Merkmale den gleichen Deskriptor, aber eine unterschiedliche Hauptorientierung, weswegen das Verfahren rotationsinvariant ist.

Im Folgenden wird ausgeführt, wie der Deskriptor relativ zu der Hauptorientierung berechnet wird.

Die Orientierung eines Keypoints relativ zum Bild wird durch seine Nachbarschaft bestimmt. Für jedes Pixel  $(x, y)$  aus dem gaußgefilterten Bild  $I$  wird der Gradient wie folgt berechnet:

$$m(x, y) = \sqrt{(I(x+1, y) - I(x-1, y))^2 + (I(x, y+1) - I(x, y-1))^2}$$

$$\theta(x, y) = \arctan((I(x+1, y) - I(x-1, y)), (I(x, y+1) - I(x, y-1)))$$

Dabei wird der Gradient in Polarkoordinaten durch seine Größe  $m$  und seine Richtung  $\theta$  beschrieben.

Um die Orientierung eines Keypoints an der Stelle  $(x, y)$  zu berechnen, wird ein Histogramm über die Gradienten der umliegenden Region aufgestellt. Dazu werden die Nachbarschafts-

punkte in ein Histogramm eingebracht, und dann wird aus ihnen der SIFT-Deskriptor berechnet.

**SIFT-Deskriptor** Der SIFT-Deskriptor setzt sich aus 128 Vektoren zusammen und wird unter Verwendung des Histogrammes der lokalen Gradientenorientierung um die Koordinaten jedes Keypoints herum berechnet (s. Abb. 2.2.1).

Das Histogramm besteht jeweils aus 36 Einträgen, die mit jeweils  $0^\circ$ ,  $10^\circ$ , ...,  $350^\circ$  bezeichnet werden. Für jedem Eintrag wird dann der stärkste Gradient gesucht und dessen entsprechende Richtung berechnet.

Der Deskriptor beinhaltet Informationen über das Umfeld des Keypoints. Er beschreibt den Keypoint eindeutig, so dass es möglich ist, den gleichen Punkt in zwei oder mehr Bildern wiederzufinden. Jeder Eintrag in diesem Histogramm bildet ein spezifisches Muster, das einem Keypoint eine hohen Wiedererkennungswert gibt.

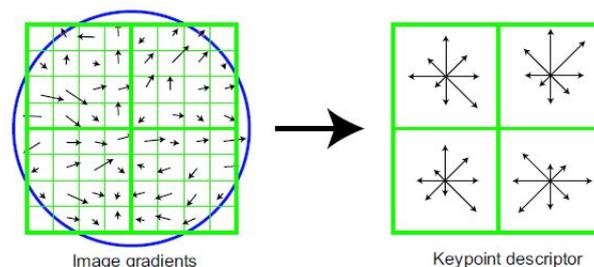


Abbildung 2.5.: Bestimmung des Deskriptors auf Basis von Gradienten. Quelle: [Lowe \(2004\)](#)

Durch die Generierung des Deskriptors wird ein Merkmal invariant gegenüber Skalierung, Drehung und Verschiebung und darüber hinaus auch noch bezüglich Helligkeits- und Kontrastveränderungen eines Objekts auf.

## 2.2.2. SURF-Verfahren

Speeded Up Robust Features (SURF) [[Bay u. a. \(2006\)](#)] ist wie SIFT eine Methode, um Keypoints aus Bildern zu berechnen und diese Punkte durch einen Deskriptor zu beschreiben.

Der SURF-Algorithmus wird als Weiterentwicklung des SIFT-Algorithmus angesehen mit dem Ziel, das Verfahren zu beschleunigen. SURF ersetzt die in SIFT verwendeten Gaußfilter durch Mittelwertfilter. Dadurch wird die Extraktion der markanten Punkte aus den Bildern beschleunigt.

In der Veröffentlichung [Bay u. a. (2006)] ist aufgeführt, wie die Mittelwertberechnung funktioniert. Hierfür nutzt man Integralbilder. Ein Integralbild dient der schnellen Berechnung von Pixelsummen innerhalb rechteckiger Ausschnitte von Bildern. Lediglich vier Rechenschritte sind dafür nötig. Diese Eigenschaft macht sich [Bay u. a. (2006)] zunutze, indem er vergleichsweise große Mittelwertfilter nutzt. Laut Aussagen des Verfassers ist der SURF-Algorithmus somit siebenmal schneller als SIFT.

**SURF-Deskriptor** Der SURF-Algorithmus weist auch bei der Berechnung des Deskriptors eine andere Vorgehensweise auf als der SIFT-Algorithmus.

Zur Berechnung des SURF-Deskriptors wird im ersten Schritt die dominante Orientierung im Umkreis eines gefundenen Keypoint durch eine Wavelet-Transformation<sup>1</sup> bestimmt.

In weiterem wird um den Merkmal eine Quadrat, der wiederum in in  $4 \times 4$  gleich große Sub-Region aufgeteilt wird. Diese wird in der Abbildung ?? veranschaulicht. In jeder Region wird das Haar-Wavelet Transformationen angewendet. Anschließend werden die Ergebnisse aus jeder Sub-Region in horizontal  $dx$  und vertikal  $dy$  Richtung gesammelt. Zusätzlich werden die Absolutwerte jeder Sub-Region addiert:  $|dx|$  und  $|dy|$  und relativ zu der Orientierung aufsummiert.

Jede Region wird damit beschrieben durch:

$$v = \sum dx_i \sum dy_i \sum |dx_i| \sum |dy_i|$$

SURF-Merkmale haben genauso wie SIFT-Merkmale eine hohe Wiedererkennungsrates und sind skalierungs- und rotationsinvariant.

Die Abbildung ?? zeigt, dass aufgrund der Berechnung der Integrale zur Erstellung der Surf-Deskriptor, sind die Keypoints von Surf robuster als die SIFT Keypoints. Die Abbildung zeigt, dass die Gradient Richtung in der Homogene Bereich der Deskriptor weniger aussagekräftiger sind, als der SURF-Deskriptor.

Die Abbildung 2.6 zeigt, dass Berechnung der Integrale bei Bild rauschen genauso stabil sind, als in Bildern mit viel Bildstörungen.

Die genaue Berechnung der Deskriptoren ist in der Veröffentlichung [Bay u. a. (2006)] aufgeführt. Dieser wird auch in der Masterarbeiten [Brauer (2010), Fries (2011)] wiedergegeben.

<sup>1</sup> Wavelet-Transformation wird eine bestimmte Familie von linearen Zeit-Frequenz-Transformationen in der Mathematik bezeichnet. Das Haar-Wavelet ist das erste in der Literatur bekannt gewordene Wavelet und wurde 1909 von Alfred Haar vorgeschlagen. Es hat eine Rechteckform.

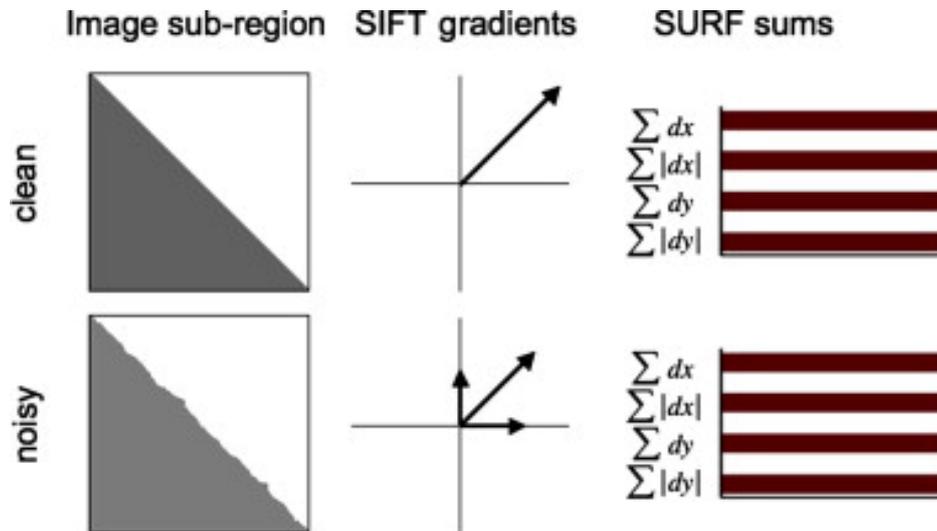


Abbildung 2.6.: SIFT vs. SURF [Quelle: Bay u. a. (2006)]

### 2.2.3. Deskriptoren-Matching

In diesem Abschnitt geht es darum, die korrespondierenden Punkte beim Abgleich von keypoints in Bildern zu finden. Hier geht es also um die Wiederfindung eines Objekts bzw. seiner Merkmale im Raum.

Es wird eine Korrespondenzanalyse durchgeführt. In Bildern gleicher oder unterschiedlicher Szenen wird nach korrespondierenden Merkmalen gesucht.

In der Literatur hat sich der Matching als ein Verfahren verstanden, bei dem ein Vergleich von Merkmalen, Bildsequenzen und Bildinformationen, in diesem Fall der Vergleich von Deskriptoren, stattfindet.

Der Deskriptor-Match hat sich in der Literatur für das Auffinden von keypoints etabliert. Diese Methode vergleicht jeweils zwei Deskriptoren vieler keypoints miteinander, um eine Korrespondenz von keypoints im Bild zu finden.

Der Deskriptor-Matcher vergleicht Deskriptoren auf Ähnlichkeit. Die Ähnlichkeit der Deskriptoren wird durch deren Abstand bestimmt.

Zwei Deskriptoren sind gleich, wenn sie keinen Abstand zueinander haben, er somit gleich null ist. Wenn der Abstand der beiden Deskriptoren also null ist, dann sind die Merkmale gleich.

Zum Vergleich der Deskriptoren verwendet man den euklidischen Abstand  $d(x, x')$ , auch euklidische Norm  $\|x - x'\|$  des Differenzvektors genannt.

Für den euklidischen Abstand zwischen zwei n-dimensionalen Vektoren  $x$  und  $x'$ :

$$d(x_1, x_2) = \|x - x'\|_2 = \sqrt{x_1 - x'_1 + \dots + x_n - x'_n}$$

Mit Hilfe des euklidischen Abstandes lässt sich der nächste Nachbar zwischen einem Deskriptor  $d_1$  und einer Menge von Deskriptoren  $D' = \{d'_1, \dots, d'_n\}$  bestimmen.

Je geringer die Distanz zwischen zwei Deskriptoren ist, umso ähnlicher sind die Merkmale einander. Dieser Eigenschaft wird genutzt, um die zu einem Keypoint gehörenden Nachbarpunkte zu einem Objekt zu bestimmen und Ausreißer zu entfernen. Ein vordefinierter Schwellenwert definiert dabei den bestimmten Abstand der Punkte zueinander.

Die Nachbarpunkte eines markantes Punktes werden miteinander verglichen, und wenn nun ein Punkt ähnlich genug ist, dann gehört dieser Punkt zu dem einen Objekt im Bild, sonst wird er verworfen.

Die lineare Suche bietet eine Möglichkeit, den Abstand von allen Punkten aus der Menge  $D'$  zum Deskriptor  $d_1$  zu berechnen und anschließend zu vergleichen.

Der Nachteil der linearen Suche ist, dass sie eine Komplexität von  $O(n^2)$  aufweist und somit die Zahl der Rechenschritte mit dem Quadrat der Anzahl der Merkmale steigt. Zur Optimierung des Verfahrens können Abbruchbedingungen eingeführt werden, aber dies kompensiert nicht die schlechte Performance.

Um die Suche nach nächsten Nachbarn zu optimieren werden K-d-Bäume eingesetzt, diese sind eine Erweiterung von binären Bäumen.

Für die Suche nach den korrespondierenden Punkten wird ein k-dimensionaler Baum (kd-Baum) verwendet, der aus der Datenstruktur (Deskriptoren) des Eingangsbildes (Suchbildes, das Bild im dem das Modell gefunden werden soll) gebildet wird.

Es handelt sich hierbei um einen Binär-Suchbaum mit Sortierschlüssel aus einer Menge von mehrdimensionalen Datenpunkten.

Jeder Baum besitzt eine bestimmte Anzahl an Knoten, die wiederum null oder zwei Blätter, d. h. dann ein linkes und ein rechtes Blatt besitzen.

Aus den Merkmalen des Eingangsbildes wird der kd-Baum aufgestellt und mit den gespeicherten Merkmalen des Modells verglichen.

Die Geschwindigkeit des Vergleichs ist abhängig von der Dimension des Deskriptors. Je niedriger sie ist, desto schneller ist der Vergleich. Aber niedrige Dimensionen erlauben nur wenige Spezifikationen des Keypoints.

Das Verfahren von Beis und Lowe ? erzeugt einen kd-Baum, bei dem die Verzweigungen stets die gleiche Tiefe annehmen, somit handelt es sich um einen optimal balancierten kd-Baum (balanced binary tree). Durch diese Optimierung reduziert sich die Zugriffszeit auf ein Blatt von der Ordnung  $O(N)$  auf die Ordnung  $O(\log N)$ .

Lowe verwendet den kd-Baum, um in dem 128-dimensionalen Vektor den nächsten Nachbarn mit dem Nearest-Neighbour-Verfahren (NN) zu finden. Für die Suche wird ein Best-Bin-First-Verfahren angewendet, das sich gut für die Suche in höheren Dimensionen eignet.

Lowe vermeidet eine langwierige Suche in einem Baum dadurch, indem er die Zahl der Blattknoten auf einen bestimmten Wert beschränkt, der eine Annäherung über die Entfernung an den nächsten Nachbarn angibt. Das Best-Bin-First-Verfahren berechnet die minimale Strecke der vom Eingangsknoten aus zu durchlaufenden Knoten im Baum.

Um die Entfernung zu berechnen bedient man sich einer priorisierten Warteschlange, in der noch nicht begangene Weg mit der gegenwärtigen Baumposition und deren Distanz zu dem Eingangsknoten gespeichert wird. Nachdem der Blattknoten überprüft wurde, wird das oberste Element der Warteschlange entnommen und für die weitere Suche verwendet.

Die Geschwindigkeit des Vergleichs ist abhängig von der Dimension des Deskriptors. Je niedriger seine Dimension desto schneller ist der Vergleich. Aber niedrige Dimensionen enthalten wenig Spezifikation des Punktes.

In der Literatur werden noch viele weitere Verfahren genannt, um den Suchaufwand im Baum zu reduzieren. In der Computer Vision Literatur bieten sich z. B. für das Finden von nächsten Nachbarn einige Verfahren an (vgl. [OpenCVbruteforcematcher](#)). Bei der Implementation des SURF-Verfahren in der OpenCV werden der Brute-Force-Algorithmus und die Fast Library for Approximate Nearest Neighbors (FLANN ) für den Vergleich von Deskriptoren benutzt, um die nächsten Nachbarn zu finden (vgl. [Lowe, Lowe und Muja \(2009\)](#)).

### 2.3. Positionsbestimmung im Raum

In diesem Abschnitt soll erklärt werden, was die Grundlagen dafür sind, um aus einem Bildpunkt den passenden Raumpunkt zu bestimmen.

Für die die Positionsbestimmung von Raumpunkten werden in der Literatur viele Methoden und Verfahren vorgestellt, die eine genaue mathematische Vorgehensweise beschreiben.

Die Ermittlung eines 3D-Raumpunktes ist eine weit verbreitete Problematik der Photogrammetrie, für die in der Literatur die einzelnen mathematischen Vorgehensweise und Methoden vorgestellt werden. Daher wird hier nicht auf die detaillierten mathematischen Verfahren und Berechnungen eingegangen werden.

Um die Raum-Position eines Objekts in einer Abbildung zu bestimmen, muss ein Raumpunkt auf einen Bildpunkt abgebildet werden. Mit Hilfe der extrahierten Keypoints einer Abbildung ist dafür die Position und Lage jeder Kamera relativ zur Position des Objekts im Raum zu bestimmen und in Weltkoordinaten zu transformieren.

Die mathematische Umrechnung einer 2D-Bildkoordinate in die 3D-Weltkoordinate wird durch das 3D-Kameramodell (siehe Abb. 2.7) ausgedrückt. In einem Kameramodell stecken insgesamt 13 verschiedene physikalische und geometrische Parameter, die durch die sogenannte Kamerakalibrierung bestimmt werden. Die Parameter bestehen aus sechs externen (extrinsischen) Kameraparametern und sieben internen (intrinsischen) Parametern.

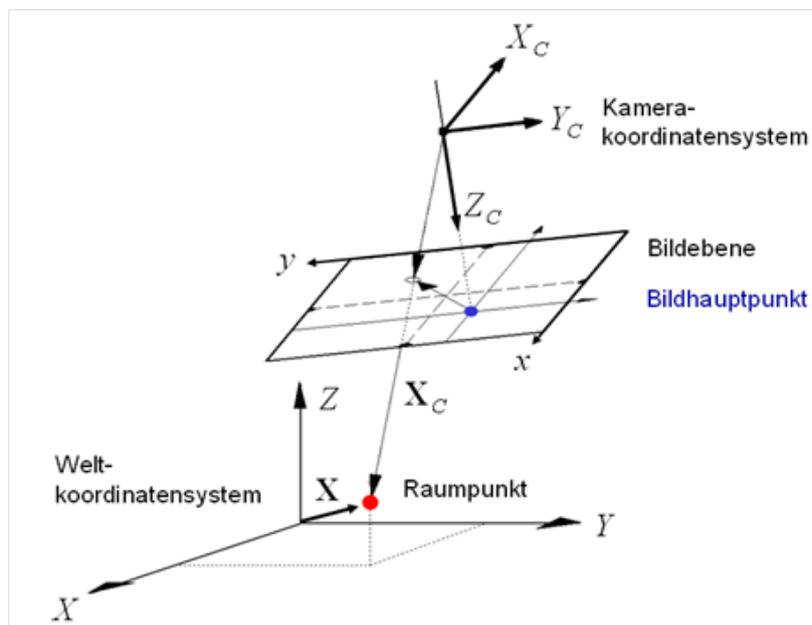


Abbildung 2.7.: Kameramodell [Quelle: Meisel (2012)]

Das Kameramodell besitzt 13 Parametern, die sich aus sechs externen (extrinsischen) Kameraparametern und sieben internen (intrinsischen) Parametern zusammensetzen, die zuvor bestimmt und berechnet werden müssen.

Die sieben internen Parameter der Kamera bestehen aus der Bildhauptpunktverschiebung, Bildweite  $C$  und

Die intrinsischen Parameter sind . . .

Die intrinsischen Parameter beschreiben die interne Geometrie der Kamera. Es definiert die Abbildung zwischen dem 3D-Kamerakoordinatensystem und dem 2D-Bildkoordinatensystem und beschreibt die Orientierung.

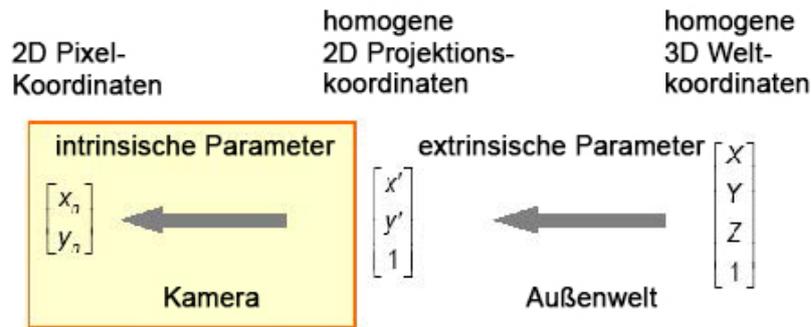


Abbildung 2.8.: Abbildung der 3D zu 2D

- die Brennweite  $f$ ,
- den Verzerrungskoeffizienten  $k_1, k_2$  für die radiale und tangentielle Verzerrung,
- den Skalierungsfaktor,
- die Koordinaten des Zentrums der Verzerrung bzgl. des verzerrten Kamerabildes.

Die externen Parameter liefern die Position und die Orientierung der Kamera, welche durch die drei Winkel [vgl. Abb. 2.9]

- $\alpha$ : Winkel um die  $x$ -Achse (Rollwinkel),
- $\beta$ : Winkel um die  $z$ -Achse (Gierwinkel),
- $\gamma$ : Winkel um die  $y$ -Achse (Nickwinkel)

in Bezug zum Weltkoordinatensystem vorliegt. Diese Parameter beschreiben den Zusammenhang zwischen dem 3D-Kamerakoordinatensystem und dem 3D-Weltkoordinatensystem. Es beschreibt die Transformation des Weltkoordinatensystems auf die Kamerakoordinaten. Die Transformation von Weltkoordinatensystem in das Kamerakoordinatensystem besteht aus einer Rotation und einer Translation. Diese Transformation hat dementsprechend 6 Freiheitsgrade, 3 für die Rotation und 3 für die Translation.

Es folgt dann die Berechnung der Kameraposition  $(x,y,z)$  und -orientierung  $(\alpha,\beta,\gamma)$  im Raum. Die Parameter können nicht direkt abgelesen werden, sondern müssen manuell durch Kalibrierung der Kamera mit einer Photogrammetrie-Software bestimmt werden.

Um die Position berechnen zu können, werden mindestens drei Bezugspunkte benötigt, deren Weltkoordinaten bekannt sind bzw. die manuell durch Kalibrierungsverfahren bestimmt werden. Dazu werden Passmarken in der Szene angebracht, deren Weltkoordinaten bekannt sind, und dienen als Bezugspunkte. Diese Passmarken sind unabhängig von der Betrachtungsperspektive (d. h. sie weisen keine Verzerrungen auf) und sind eindeutig identifizierbar.

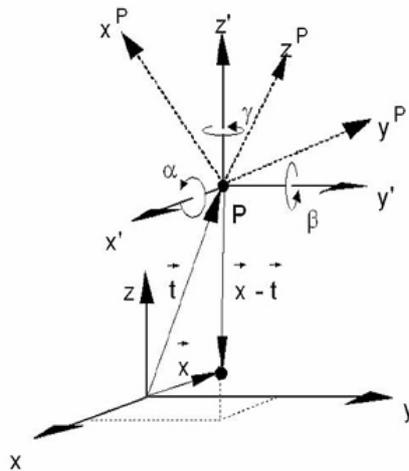


Abbildung 2.9.: Kameraparameter im Weltkoordinatensystem [Quelle: Meisel (2012)]

Dann werden Kalibrierungsverfahren angewendet, die durch Lösung von Matrixgleichungen den Raumpunkt ermitteln.

In dem Buch *Computer vision* [vgl. Hartley und Zisserman (2004)] ist beschrieben, wie durch die Lösung der Rotations- und Translations-Matrizen und die daraus berechnete Fundamentalmatrix der Raumpunkt bestimmt wird.

Ein Verfahren ist der Direct Linear Transformation (DLT) algorithm (vgl. Hartley und Zisserman, 2004, S. 88), welche sich durch das ausschließliche Lösen von linearen Gleichungen auszeichnet.

Das Verfahren ist dadurch gekennzeichnet, dass für die Messung und die Durchführung der Kalibrierung nur 4 Passpunkte benötigt werden. Es ist ein einfaches Verfahren, den mathematischen Zusammenhang zwischen Bild- und Weltkoordinaten zu bestimmen, mit dem man beliebig viele Punkturechnungen durchführen kann. Der Preis dieser Einfachheit ist, dass die radiale Linsenverzerrung nicht berücksichtigt wird. Dieser hat die Auswirkung, dass die Messpunkte nicht exakt bestimmt werden können und einen leichten Messfehler enthalten.

Für diese Masterarbeit ist eine Messung auf Millimeter-Genauigkeit nicht erforderlich und somit kann DLT ein mögliches Verfahren für die Bestimmung der Raumpunkte sein.

Ein anderes in der Praxis häufig verwendetes Kamerakalibrierungsverfahren ist die Tsai-Kalibrierung [vgl. Tsai (1986)][R. Kurazume und Ikeuchi (January 2002)]. In vier Schritten werden die Matrizen zur Berechnung des Raumpunktes bestimmt. Das Verfahren nach Tsai modelliert zusätzliche Verzerrungen der Kamera und ist somit ein Vermessungs-Verfahren, bei dem die Punkte subpixelgenau bestimmt werden können.

Sind nun die Kalibrierungsparameter bestimmt, können beliebige Raumpunkte (3D-Koordinaten) zu einem Bildpunkt (2D-Koordinaten) berechnet werden.

### 2.3.1. Stereo Vision

Angelehnt an das menschliche räumliche Sehen, wird auch in der digitalen Bildverarbeitung das Stereoanalyseverfahren verwendet, um die Position des Objektes im Raum anhand geometrischer Beziehungen zu bestimmen.

**Stereoanalyse** In dieser Masterthesis sollen nur statische Stereoanalyseverfahren betrachtet werden, da hier von der Annahme ausgegangen wird, dass keine Objekt- oder Kamerabewegung während der Aufnahme der Bilder stattfindet.

Im Folgenden werden die Arbeitsschritte des Verfahrens und der Lösungsweg erläutert.

1. Bildaufnahme:

In einer Szene wird das Objekt von zwei unterschiedlichen Standorten bzw. von zwei verschiedene Kameras aufgenommen.

Der Lösungsansatz in der Masterthesis ist wie folgt:

Sobald das Objekt von einer Kamera (CAM-1) in einem Bereich der Wohnung erkannt wird, soll eine zweite Kamera beauftragt werden, den Bereich der CAM-1 zu durchsuchen und dasselbe Objekt wiederzuerkennen.

2. Merkmalsextraktion:

Es werden signifikante Bildmerkmale (hier SURF-Feature-Points) in den Bildern bestimmt.

3. Korrespondenzanalyse:

Bestimmung von korrespondierenden Elementen in beiden Bildern. Dies wird mit Hilfe des SURF-Matching-Verfahrens realisiert.

4. Tiefenwertbestimmung:

Die Berechnung der Disparität, des Differenzvektors zweier korrespondierender Merkmale liefert die Tiefeninformation eines Objektes in der Szene. Hierfür werden zwei eindeutige Matching-Points gewählt und daraus die Disparität berechnet.

Grundvoraussetzungen für das Stereoanalyseverfahren sind die Kamerakalibrierung und die prinzipiell parallele Anordnung der Kameras in der Szene.

**Annahmen und Einschränkung der Stereo Vision** Im Allgemeinen stellt die Korrespondenzanalyse das schwierigste Teilproblem im Stereoverfahren dar, da beim Bestimmen der korrespondierenden Bildmerkmale in Stereobildern Mehrdeutigkeiten bei der Zuordnung in einer Szene vorkommen können.

Das Problem wird durch das eingesetzte Match-Verfahren [vgl. Deskriptor Match 2.2.3 und (Reinhard Klette, 1996, Seite 156–165)] reduziert und teilweise behoben.

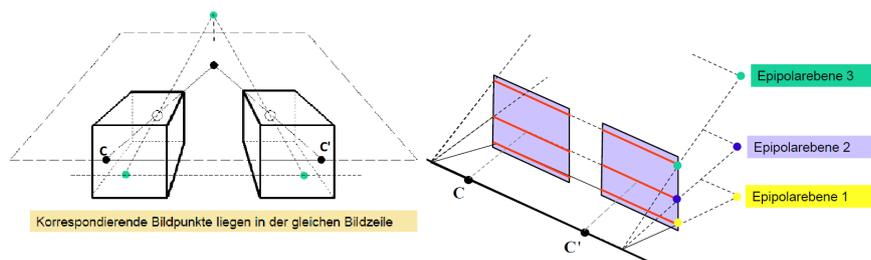


Abbildung 2.10.: Achsen-parallele Anordnung [Quelle: Meisel (2012)]

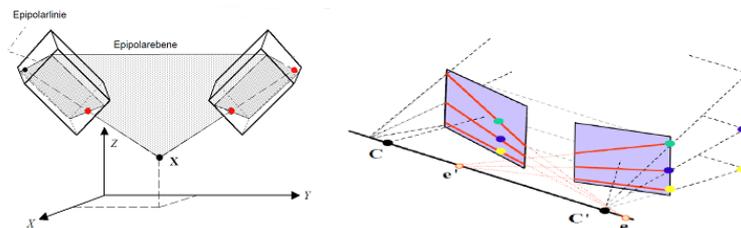


Abbildung 2.11.: Epipolarebenen und Epipolarlinien Quelle:(Meisel, 2012)

Das Matching-Verfahren ist eine merkmalsbasierte Korrespondenzanalyse. Hierdurch sind die Mehrdeutigkeiten bei Korrespondenzen geringer und die Bestimmung der Disparitäten erfolgt genauer, denn die Position der Merkmale kann subpixelgenau berechnet werden.

Dennoch bietet das statische Stereoanalyseverfahren nicht in allen Bereichen der Wohnung eine Lösung zur Gewinnung der Tiefeninformation an. Zum einem sind nicht alle Bereiche der Wohnung mit zwei Kameras ausgestattet, zum anderem kann die aufwendige Steuerung (Justierung) der Kameras nur in einigen Bereichen der Wohnung realisiert werden, da nicht überall die optischen Achsen ( $C_l$  und  $C_r$ ) der Kameras parallel verlaufen und die Basislinie ( $x$ - und  $y$ -Achse) richtungsmäßig mit der Bildebene übereinstimmt, wie es in der Abbildung 2.10 dargestellt ist.

Sind die Kameras beliebig zueinander ausgerichtet, wie in der Abb. 2.11 dargestellt, müssen

die Epipolarlinien und Epipolarebenen bestimmt werden, was einen höheren Rechenaufwand mit sich bringt (vgl. [Meisel \(2012\)](#), [Reinhard Klette \(1996\)](#), [Xiaoyi Jiang](#)).

Im weiteren Verlauf ist zu untersuchen, welche weiteren Verfahren für die Bestimmung eines Punktes im dreidimensionalen Raum auch mit nur einer Kamera und evtl. mit Hilfe eines aktiven Lichtstrahls realisierbar sind und sich als effizient für die Umgebung im Living Place Hamburg erweisen. Hierfür kommen z. B. die Verfahren triokulares Stereo, aktives Stereo und axiales Stereo (vgl. [Jiang und Bunke, 1997](#)) in Frage.

## 3. Anforderungsanalyse

Gegenstand dieses Kapitels ist die Analyse der Problemstellung und Erfassung der Anforderungen an das System. Es wird ausgeführt, welche Fragestellungen sich aus den Aufgabenbereichen ergeben, und in Abschnitt 3.5 auch, welche Forschungsarbeiten Lösungswege für die Probleme bieten. Vor allem aber stellt dieses Kapitel vor, wie diese Ziele umgesetzt werden sollen.

Zuerst werden „Sals Morgen-Routine“ und andere Szenarien im Living Place Hamburg an die Aufgabenstellung des kamerabasierten Objektsuch-Systems angepasst (3.1). Hieraus werden dann im folgenden Abschnitt 3.2 die Dienste extrahiert, die das System den Bewohnern des Living Place Hamburg anbieten soll. Damit wird diese Arbeit die Services im Living Place Hamburg um neue Möglichkeiten erweitern.

Der Abschnitt 3.4 der Analyse beschreibt schließlich die Anforderungen, die sich aus den Szenarien und den Anwendungsfällen für das kamerabasierte Objektsuche-System ergeben.

### 3.1. Anwendungsszenarien

Für die Implementierung von Services im Living Place Hamburg wurde bereits das Szenario „Sals Morgen-Routine“ entwickelt, welches an das „Sal“-Szenario in dem Aufsatz „The computer for the 21st century“ von Mark Weiser (1991) angelehnt ist. Es wurde bereits in den Masterarbeiten von (Karstaedt, 2012; Ellenberg, 2011; Voskuhl, 2011) im Kontext des Living Place Hamburg verwendet.

Für diese Masterarbeit wurde das Szenario adaptiert, um zwei Abwandlungen ergänzt und dem kamerabasierten Objektsuch-System angepasst.

Die folgenden Szenarien sollen verdeutlichen, wie das Zusammenspiel der Services in dem Gesamtkonzept des Living Place Hamburg aussehen soll und wie das kamerabasierte Objektsuch-System von dem Bewohner im Living Place Hamburg genutzt werden kann.

### 3.1.1. Sals Morgen-Routine

Sal wird geweckt durch sanfte Musik und einen simulierten Sonnenaufgang, denn der Wetterdienst-Agent meldet, dass es regnerisch und bewölkt ist.

Sie verlässt das Bett und die Lichter der Wohnung passen sich ihrer Stimmung an. Während Sal noch im Bad verweilt, startet die Kaffeemaschine und die neuesten Nachrichten auf den Monitoren werden für sie vorbereitet. Beim Verlassen des Bades ertönt aus den Lautsprecher der Wohnung „Du hast eine Nachricht auf deinem Handy“.

Sals ruft in die Wohnung „Wo ist mein Handy?“. Aus der Wohnung ertönt „Dein Handy liegt im Essbereich auf dem Multitouch neben dem Buch“. Auf ihrem Monitor erscheint das Kamerabild aus dem Essbereich. Darin ist das Buch auf dem Multitouch markiert.

### 3.1.2. Sal vorm Schlafen gehen

Sal ist beim Essen und denkt sich, dass sie doch gerne vor dem Zu-Bett-Gehen ein Buch lesen möchte. Darum möchte sie wissen, ob das Buch bereits am Bett liegt oder ob sie es erst suchen und dorthin mitnehmen muss.

Da fragt sie die Wohnung. „Liegt das Buch ‚Du schaffst es‘ auf dem Bett?“ Die Kamera im Schlafbereich wird beauftragt, die Suche aufzunehmen. Das kamerabasierte Objektsuch-System kann das Buch aber nicht im Schlafbereich finden, daraufhin entdeckt es das Buch auf dem Fernsehtisch im Wohnbereich. Aus den Lautsprechern ertönt: „Das Buch ist im Wohnbereich, auf dem Fernsehtisch neben der Fernbedienung.“

### 3.1.3. Sal hinterlegt das Buch im System

Da Sal ein neues Buch gekauft hat, möchte sie dieses im kamerabasierten Objektsuch-System hinterlegen und somit in der Wohnung bekannt machen.

Hierfür ruft sie die Anwendung auf und startet das Erfassen des Objektes. Sie ruft den Service „Objekt im System hinterlegen“ auf, macht von allen Seiten Fotos von dem Buch und hinterlegt die Bilder im System. Zudem gibt sie noch den Namen des Buches und einige Schlagworte und ergänzende Informationen zu dem gekauften Buch ein. Dann betätigt Sal nur noch einen Button zum Erfassen des Buches im System und prüft die Angaben.

## 3.2. Analyse der Szenarien

Jedes Szenario beschreibt einen bestimmten Service, der dem Bewohner des Living Place Hamburg angeboten werden kann.

Analysiert man die Szenarien, so lassen sich folgende Services extrahieren:

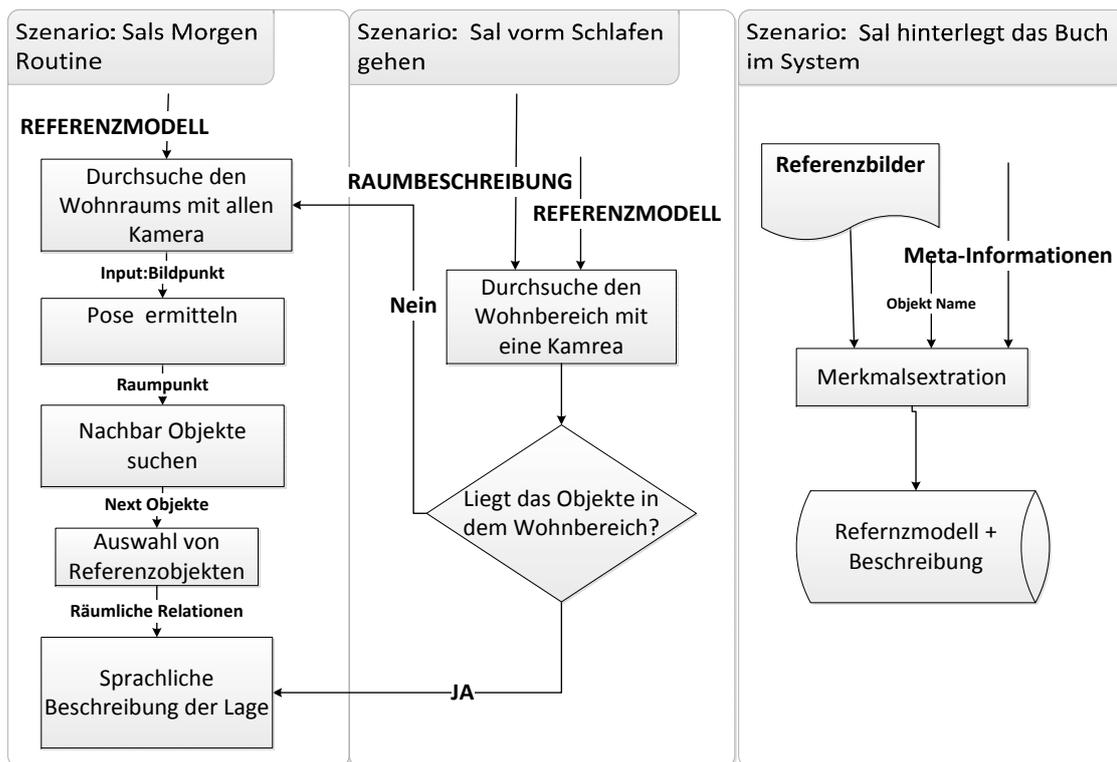


Abbildung 3.1.: Analyse der Szenarien

Aus den Szenarien „Sals Morgen-Routine“ 3.1.1 und „Sal vorm Schlafen gehen“ 3.1.2 kann man die Frage nach den Handy und die Frage nach dem Buch entnehmen.

Die Frage „Wo ist das Handy“ bzw. „Wo ist das Buch“ wird an das kamerabasierte Objektsuch-System gestellt, dabei dienen das Handy und das Buch als Eingabe. Als Ausgabe liefert der Service die Position des Objektes im Raum.

In dem Szenario 3.1.3 nutzt der Benutzer einen Service, um eine bestimmtes Objekt im System zu hinterlegen. Es zeigt, wie dieser Service Objekte im System zu hinterlegen erlaubt. Dabei handelt es sich um eine Art von Datensammlung.

Aus den beschriebenen Szenarien lassen sich bestimmte Abläufe extrahieren, wie in Abbildung 3.1 zu sehen ist. Im folgenden werden die einzelnen Schritte und die daraus extrahierten Services des Systems erläutert.

### 3.2.1. Szenarien mit Suchanfrage an das System

In den Szenarien „Sals Morgen-Routine“ 3.1.1 und „Sal vorm Schlafen gehen“ 3.1.2 findet sich die Frage nach dem Handy bzw. dem Buch.

#### Szenario: Sals Morgen-Routine

Der Benutzer benennt ein Suchobjekt, der Service erhält daraufhin das dazugehörige Referenzmodell als Eingabe. Das System beauftragt daraufhin alle Kameras der Wohnung, gleichzeitig alle Wohnbereiche nach dem angegebenen Objekt zu durchsuchen. Liefert die Suche ein positives Ergebnis, dann wird die Position des Objektes zurückgeliefert. In der Abbildung heißt dieser Prozess „Pose ermitteln“.

Das System liefert als Ergebnis der Suche zunächst eine Raumkoordinate, die aus einer  $x$ -, einer  $y$ - und einer  $z$ -Ordinate besteht. Der Mensch kann dem nicht ohne weiteres entnehmen, wo sich ein Objekt nun befindet, daher muss die Koordinate noch in einen Satz umgewandelt werden, der die Lage des Objektes adäquat und dem Kontext angepasst beschreibt.

Für diese sprachliche Beschreibung werden weitere Referenzobjekte als Bezugspunkte benötigt, welche in unmittelbarer Nähe dieses Raumpunktes liegen.

Dafür liefert der Prozess „Nachbar-Objekte suchen“ der Abbildung 3.1 eine Liste von Objekten, aus denen im Schritt „Auswahl des Referenzmodells“ die passenden Objekte als Referenzobjekte gewählt und deren räumliche Relationen zueinander bestimmt werden.

#### Szenario: Sal vorm Schlafen gehen

Das Szenario „Sal vorm Schlafen gehen“ 3.1.2 verdeutlicht den Bedarf an einem Service, der gezielt einen bestimmten Bereich der Wohnung nach einem bestimmten Objekt absucht.

Als Ergebnis teilt er dann mit, ob sich das gesuchte Objekt am vermuteten Platz befindet oder nicht. Sollte sich die vermutete Position nicht bestätigen, so soll dem Benutzer noch die tatsächliche Position des Objekts genannt werden.

In diesem Szenario ist die Eingabe für den Service zum einen das Referenzmodell, zum anderen eine räumliche Beschreibung der Lage des Objektes in der Wohnung bzw. eine Vermutung hinsichtlich des Ortes, an dem es sich befinden könnte. Das System entscheidet an Hand der räumlichen Informationen, welche Kamera in der Wohnung mit der Suche beauftragt werden soll.

Um eine Antwort auf diese Service-Anfrage zu liefern, muss eine gezielte Suche durchgeführt werden. Ist die Suche erfolgreich, so wird dem Benutzer seine Anfrage bestätigt. Als Ergebnis des Szenarios kann z.B. folgender Satz an den Benutzer ausgegeben werden: „Ja das Buch befindet dort.“ Oder: „Ja, das Buch befindet sich im Schlafbereich auf der Kommode.“

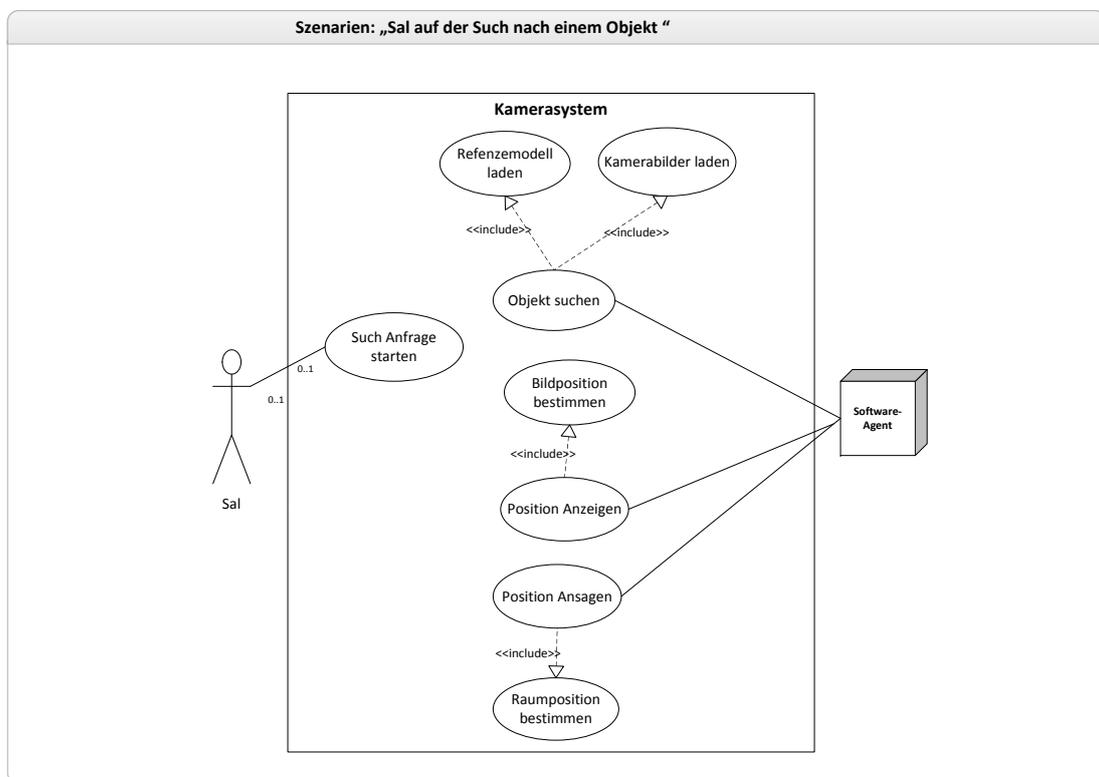


Abbildung 3.2.: Anwendungsdiagramm: Objektsuche im Raum

Ist die Suche der Kamera in dem genannten Wohnbereich erfolglos, so wird die Suche mit den anderen Kameras auf weitere Bereiche der Wohnung ausgedehnt. Dies entspricht den Prozess „Durchsuchen der Wohnung mit allen Kameras“ in der Abbildung 3.1. Sobald eine der Kameras ein positives Ergebnis liefert, wird der Suchauftrag für alle Kameras gestoppt. Als Ausgabe ist in diesem Fall eine negative Antwort denkbar, verbunden mit der Angabe der tatsächlichen Position des Objektes. So kann z. B. der Antwortsatz wie folgt lauten: „Nein,

das Buch liegt nicht im Schlafbereich. Das Buch befindet sich im Wohnbereich auf dem Fernsehtisch.“.

### **Anwendungsfälle: Suchanfragen an das System**

In diesen Szenarien wurden nun in Rahmen der Masterarbeit die Anwendungsfälle identifiziert, wie sie in dem Anwendungsfalldiagramm 3.2 dargestellt sind.

In dem System gibt es zwei Akteure. Einmal den Bewohner des Living Place Hamburg , der als ein Strichmännchen auf der linken Seite darstellt ist, und dann einen Software-Agenten, der das Objektsuchsystem darstellt und sich auf der rechten Seite des Diagramms befindet. Die einzelnen Anwendungsfälle werden durch Ellipsen repräsentiert.

Bei genauerer Betrachtung der beiden Szenarien können Gemeinsamkeiten entdeckt werden, so unterscheiden sie sich nur bezüglich der Eingabedaten und somit in einem Detail der Service-Anfrage an das System. So soll in dem Szenario „Sals Morgen-Routine“ 3.1.1 nach dem Objekt gezielt in einem Bereich gesucht werden, während in dem Szenario „Sal vorm Schlafen gehen “ 3.1.2 die Suche nicht auf einem Bereich eingeschränkt ist. Das Szenario 3.1.2 kann somit als eine Modifikation des Szenarios 3.1.1 angesehen werden.

Für die obigen Szenarien wurden folgende Anwendungsfälle festgelegt: Sal, der Bewohner, führt als aktiver Akteur folgende Anwendungsfälle durch:

- „Suchanfrage starten“, beschrieben in Tabelle 3.2.1,
- „Suchanfrage stoppen“ ist eine alternative Aktion, die der Akteur zu jeder Zeit durchführen kann.

Der Software-Agent ist ein weiterer Akteur des Anwendungsfalldiagramms 3.2, der für folgende Anwendungsfälle verantwortlich ist:

- „Objekt suchen“ (Tabelle 3.2.1 ),
- „Position anzeigen“ (Tabelle 3.2.1 ) und
- „Position ansagen“ (Tabelle 3.2.1),

wobei die beiden Anwendungsfälle „Position anzeigen “ und „Position ansagen“ die Basis für weitere Anwendungsfälle darstellen, denn um die Komplexität der Anwendungsfälle zu reduzieren, wurden diese abgetrennt und dann von den Basisanwendungsfällen inkludiert. Dementsprechend inkludiert der Anwendungsfall „Position anzeigen“ den Anwendungsfall „Bildposition bestimmen“ und vor dem Anwendungsfall „Position ansagen“ muss der Raumpunkt bestimmt werden, was durch den Anwendungsfall „Raumposition bestimmen“ dargestellt ist.

In folgenden werden die Anwendungsfallbeschreibungen für die Basisanwendungen dargestellt, wobei in den Beschreibungen die einzelnen Prozessabläufe dargestellt werden sollen.

### **Anwendungsfall: Service-Anfrage an das System stellen**

<b>Anwendungsfall</b>	Suchanfrage starten
<b>Hauptakteur</b>	Sal, als Bewohner des Living Place Hamburg .
<b>Nebenakteur</b>	Der passive Akteur ist ein Software-Agent , der ein Teilsystem des kamerabasierten Objektsuch-Systems darstellt.
<b>Vorbedingung</b>	Im kamerabasierten Objektsuch-System ist das zu suchende Referenzmodell bereits hinterlegt. Es existiert eine Benutzerschnittstelle (Suchmaske), die eine Auswahlliste mit den Namen gespeicherter Objekte enthält.
<b>Nachbedingung</b>	Suche starten in den Bildern der Kamera. Objektsuche im Raum auslösen.
<b>Normaler Ablauf</b>	<ol style="list-style-type: none"> <li>1. Sal öffnet die Suchmaske</li> <li>2. In der Auswahlliste werden alle Namen von Referenzmodellen im System aufgeführt</li> <li>3. Sal wählt den Namen des Referenzmodells aus der Liste aus.</li> <li>4. Der Software-Agent zeigt das Bild des Objektes in der Suchmaske.</li> <li>5. Sal betätigt den Button „Start/Stop“ einmal</li> </ol>
<b>Alternativablauf</b>	<p>Objektauswahl über einen Typen einschränken</p> <p>Sal schränkt die Liste mit den Referenzmodell-Namen über die Auswahl eines Typen ein</p> <ol style="list-style-type: none"> <li>2. Sal gibt einen Typ in die Suchmaske ein.</li> <li>3. Alle Namen des Referenzmodell mit dem gewählten Typen erscheint in der Auswahlliste auf der Suchmaske</li> </ol>
<b>Alternativablauf</b>	Objektauswahl über Metainformationen einschränken. Sal schränkt die Liste mit den Referenzmodell-Namen durch Eingabe von Metainformationen ein.

2. Sal gibt Metainformationen in der Suchmaske ein.
3. Alle Namen des Referenzmodells, die passende Metainformationen enthalten, erscheinen in der Auswahlliste in der Suchmaske.

Tabelle 3.1.: Anwendungsfall: Service-Anfrage an das System stellen

**Anwendungsfall: Objektsuche in Bildern**

<b>Anwendungsfälle</b>	„Objekt suchen“; „Referenzmodell laden“; „Kamerabilder laden“; „Bildposition bestimmen“
<b>Hauptakteur</b>	Ein Software-Agent , der ein Teilsystem des Kamerabasierte Objektsuche System darstellt.
<b>Vorbedingung</b>	Im System ist das zu suchende Referenzmodell bereits hinterlegt. Es existiert ein Eingabemaske für die Suche. Die Suchmaske stellt eine Auswahl der Objekte zu Verfügung.
<b>Trigger</b>	Anwendungsfall „Servicefrage starten“.
<b>Ergebnis</b>	Die Position des Objektes im Bild (Bildposition).
<b>Hauptablauf</b>	<ol style="list-style-type: none"> <li>1. Der Softwareagent lädt das Referenzmodell.</li> <li>2. Der Softwareagent ruft die Live-Bider aller Kameras ab.</li> <li>3. Der Softwareagent übergibt das Referenzmodell an die internen Kamera-Suchagenten CAM1 bis CAM6.</li> <li>4. Der Softwareagent stellt für jede der Kamera (CAM1 bis CAM6) eine Standardkonfiguration für die Suche ein.</li> <li>5. Der Softwareagent ruft neue Frame-Video-Streams auf.</li> <li>6. Der Softwareagent starte eine Parallele suche nach den Merkmalen des Referenzmodells in jedes Frame einer Kamras</li> <li>7. Jeder Kamera-Suchagent durchsucht seinen Bereich nach dem Refernezmodell.</li> </ol>

	8. Die Suche war Erfolgreich.
<b>Alternativablauf:</b>	7a. Die Such war in einem Bereich der Living Place Hamburg Erfolglos. 8a1. Einstellung der Kameras ändern und Suche erneut starten. 8a2. Der Softwareagent ruft neues Frame von der Kamera auf und durchsucht jedes Frame nach dem Referenzmodell. Weiter mit 7. Die Schritte 8a1 und 8a2 können beliebig wiederholt werden.
<b>Alternativablauf</b>	8b.Die Such war in allen Bereichen Erfolglos. 8b1. Der Softwareagent bricht die Suche ab.

Tabelle 3.2.: Anwendungsfall: Objektsuche in Bildern

**Anwendungsfall: Objekt anzeigen**

	Anwendungsfall] Position anzeigen
<b>Hauptakteur</b>	Ein Software-Agent , der ein Komponente des kamerabasierten Objektsuch-Systems darstellt.
<b>Nebenakteur</b>	Sal, als Bewohner des Living Place Hamburg .
<b>Trigger</b>	Der Anwendungsfall „Objektsuche“
<b>Vorbedingung</b>	Der Software-Agent hat das Objekte in den Bildern wiedergefunden. Der Bildpunkt des Objektes ist dem Kamerasystem bekannt. Es besteht eine Verbindung zum Live-Stream der Kamera, mit der das Objekt erkannt wurde. Die Position des Bewohners ist bekannt.
<b>Ergebnis</b>	Anzeige des Objektes auf einem der Monitore im Living Place Hamburg .
<b>Hauptablauf</b>	Der Software-Agent hat das Objekte in der Wohnung wiedergefunden und führt als nächstes folgende Schritte aus:

	<ol style="list-style-type: none"> <li>1. Aufruf der Video-Stream-Kamera (aus CAM1 bis CAM6), in dem das Referenzmodell lokalisiert wurde.</li> <li>2. Markieren und Anzeigen des gefundene Objektes in dem Video-Stream.</li> <li>3. Verbindungsaufbau mit den dem Bewohner nächstgelegenen Monitoren im Living Place Hamburg .</li> <li>4. Markieren und Anzeigen des Objektes im Live-Stream der Kamera.</li> <li>5. Anzeigen der Bildposition auf der Benutzeroberfläche.</li> </ol>
<b>Alternativablauf</b>	<ol style="list-style-type: none"> <li>4. Aufruf des 2D-Modells des Living Place Hamburg auf dem Monitor</li> <li>5. Anzeigen der Position des Objektes im 2D-Raummodell.</li> </ol>

Tabelle 3.3.: Anwendungsfall: Objekt anzeigen

**Anwendungsfall: Pose ermitteln**

<b>Anwendungsfall</b>	Raumposition bestimmen
<b>Akteur</b>	Ein Software-Agent .
<b>Trigger</b>	Objekt wurde von dem Software-Agent gefunden. Dies wird getriggert von dem Anwendungsfall „Position ansagen“
<b>Vorbedingung</b>	Ein Objekt wurde vom Software-Agent in den Bildern wiedergefunden. Der Bildpunkt des Objektes ist bekannt. Es besteht eine Verbindung zum Live-Stream der Kamera, von der das Objekt gefunden wurde.
<b>Ergebnis</b>	Ermittlung des Raumpunkts (3D) zu dem 2D-Bildpunkt.
<b>Standardablauf</b>	<ol style="list-style-type: none"> <li>1. Aufruf der Video-Stream-Kamera, in der das Referenzmodell lokalisiert wurde.</li> </ol>

2. Analyse, wieviel weitere Kameras das Referenzmodell in ihrem Sichtbereich haben.
3. Bestimme das Verfahren zur Ermittlung des Raumpunktes.
4. Ermittle den Raumpunkt.

Tabelle 3.4.: Anwendungsfall: Pose ermitteln

**Anwendungsfall: Position ansagen**

<b>Anwendungsfall</b>	Position ansagen
<b>Akteure</b>	Ein Software-Agent , der ein Komponente des Kamerabasierte Objektsuche System darstellt ist der aktive Akteur. Sal, als Bewohner des Living Place Hamburg ist der passive Akteur. Ein externer Service-Anbieter, der Referenzobjekte für das Kamerabasierte Objektsuche System zu Verfügung stellt.
<b>Trigger</b>	Der Anwendungsfall „Orten von Objekten im Raum“
<b>Vorbedingung</b>	<p>Der Raumposition des Objektes ist bekannt.</p> <p>Das Software-Agent ist mit der Technologie und Living Place Hamburg Systeme verbunden</p> <p>Das Software-Agent hat einen Zugriff auf den externen Service-Anbieter, der weitere Referenzobjekte zur Verfügung stellt.</p> <p>Das Software-Agent hat einen Zugriff auf einen Text-to-Speech-Agenten.</p>
<b>Ergebnis</b>	Sprachliche Beschreibung der Position des Objekte im Living Place Hamburg .
<b>Standard-Ablauf</b>	<ol style="list-style-type: none"> <li>1. Der Software-Agent sendet eine Service-Anfrage „Was befindet sich in unmittelbare Umgebung der 3D-Raumkoordinaten?“ an den externen Service-Anbieter.</li> <li>2. Der Service-Anbieter liefert eine Liste von Objekten in unmittelbarer Umgebung der Koordinaten.</li> </ol>

- 
3. Der Software-Agent selektiert aus den gelieferte Objekten die „prominenten Objekte“.
  4. Der Software-Agent bildet Relationen in Bezug zu weiteren dem Objekt im Raum
  5. Der Software-Agent formuliert aus den Relationen und räumlichen Informationen einen Satz, der die räumliche Lage des Objekte beschreibt.
  6. Der Software-Agent ruft einen internen Agenten für die Sprachausgabe auf.
  7. Der Software-Agent übergibt den formulierten Satz in Textform an den Sprachausgabe-Agenten.
  8. Die Position des Objektes im Raum wird für Sal über den Lautsprecher angesagt.
- 

Tabelle 3.5.: Anwendungsfall: Position ansagen

### 3.2.2. Szenario: Objekt im System hinterlegen

In dem Szenario 3.1.3 nutzt Sal einen Service, um ein Objekt im System zu hinterlegen. Dies wurde in der Abbildung 3.1 „Szenario: Sal hinterlegt das Buch im System“ modelliert. Wie die Abbildung zeigt, werden hier Daten erfasst.

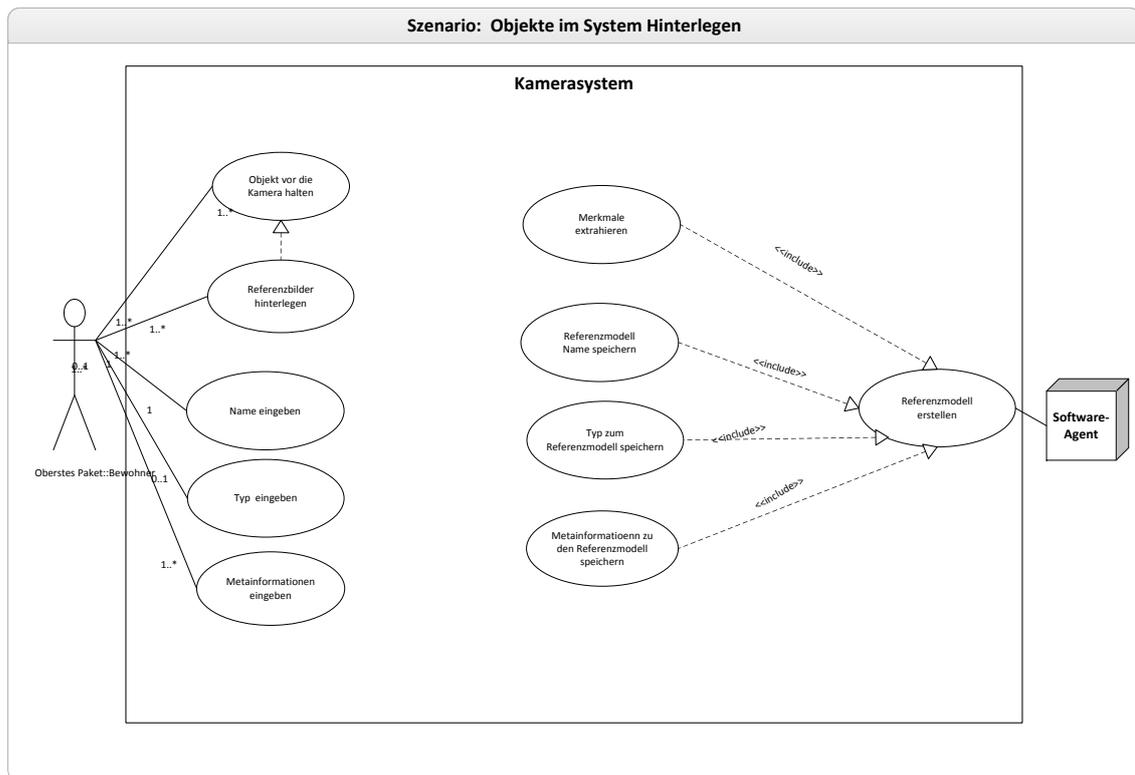


Abbildung 3.3.: Anwendungsdiagramm: Objekt im System hinterlegen

Von Sal werden die Fotos des Buches (in der Abb. 3.1 als Referenzbilder bezeichnet) und der Name des Buches eingegeben. Die Zusatzinformationen zu dem Buch werden als „Meta-Informationen“ zusammengefasst.

Im nächsten Schritt werden diese Eingaben vom System zu einem Referenzmodell verarbeitet und dem Gesamtsystem zur Verfügung gestellt, indem diese in einem Datencontainer (Datenbank) abgelegt werden. Dieses Referenzmodell dient weiteren Services als Eingabe.

### Anwendungsfälle: Objekt im System hinterlegen

Aus den modellierten Prozessen in der Abbildung 3.1 wurden für das Szenario die Basisanwendungsfälle „Referenzbilder hinterlegen“ und „Referenzmodell erstellen“ festgelegt.

Diese Anwendungsfälle sind Basisanwendungsfälle, die weitere Anwendungsfälle einbinden. Das Anwendungsfalldiagramm 3.3 stellt die Basisanwendungsfälle mit ihrer Beziehung zu weiteren Anwendungsfällen und Akteuren dar. Die Beschreibung des Basisanwendungsfalls „Referenzbilder hinterlegen“ ist in der Tabelle 3.2.2 und die von „Referenzmodell erstellen“ in der Tabelle 3.2.2 aufgeführt.

#### Anwendungsfall: Referenzbilder hinterlegen

<b>Anwendungsfälle</b>	„Objekt vor die Kamera halten“, „Referenzbilder hinterlegen“, „Namen eingeben“, „Typ eingeben“, „Metainformationen eingeben“
<b>Akteure</b>	Der aktive Akteur ist Sal als Bewohner des Living Place Hamburg . Der passiver Akteur ist ein Teilsystem als Software-Agent .
<b>Trigger</b>	Der Anwendungsfall wird von Sal gestartet. Es gibt keinen Anwendungsfall, der diesen triggert.
<b>Vorbedingung</b>	Es existiert eine Benutzerschnittstelle (Eingabemaske) für die Aufnahme der Referenzbilder.  Die Eingabemaske bietet die Möglichkeit, zu jedem Objekt den Namen, Typen und weitere Metainformationen zu dem Referenzmodell zu hinterlegen. Für die Bildaufnahmen ist eine Kamera mit dem Kamera-basierte Objektsuche System verbunden. Die Kameraeinstellung ist so gewählt, das nur das zu erfassendes Objekt im Fokus der Kamera ist. Ein vordefinierter Abstand zwischen der Kamera und dem Objekt wird von Sal eingehalten.
<b>Ergebnis</b>	Hinterlegte Referenzbilder zu einem Objekt mit den zugehörigen Namen, Typen und weiteren beschreibenden Metainformationen
<b>Standardablauf</b>	<ol style="list-style-type: none"> <li>1. Sal ruft die Eingabemaske auf.</li> <li>2. Sal gibt in der Eingabemaske den Namen, Typ und weite Metainformationen ein.</li> <li>3. Sal hält das Objekt vor die Kamera</li> </ol>

4. Das Objekt wird vom Software-Agent erfasst und grün umrandet, sobald die Erfassung korrekt erledigt ist.
5. Sal betätigt den „Bild speichern“-Button.
6. Der Software-Agent speichert das Bild unter dem angegebenen Namen und verknüpft die Daten mit dem Typen und Metainformationen

Schritte 3 bis 5 können beliebig oft wiederholt werden, um neue Referenzbilder zu hinterlegen.

Tabelle 3.6.: Anwendungsfall: Referenzbilder hinterlegen

### Anwendungsfall: Referenzmodell erstellen

<b>Anwendungsfälle</b>	„Merkmale extrahieren“, „Namen des Referenzmodells speichern“, „Typ des Referenzmodells speichern“, „Metainformationen zum Referenzmodell speichern“
<b>Akteure</b>	Der aktive Akteur ist Sal, als Bewohner des Living Place Hamburg . Der passiver Akteur ist ein Software-Agent , der ein Teilsystem des kamerabasierten Objektsuch-Systems darstellt.
<b>Trigger</b>	Dieser wird von Sal und den Anwendungsfall „Referenzbilder hinterlegen“ getriggert.
<b>Vorbedingungen</b>	Es existiert ein Button auf der Eingabemaske zum Erstellen des Referenzmodells. Sal hat einen Namen eingetragen. Der angegebene Name existiert nicht im System. Es wurde mindestens ein Referenzbild zu dem Objekt im System gespeichert durch den Anwendungsfall „Referenzbilder hinterlegen“.
<b>Ergebnis</b>	Ein aus den Merkmalen der Referenzbilder erstelltes Referenzmodell.
<b>Standardablauf</b>	1. Sal betätigt den „Referenzmodell erstellen“-Button auf der Eingabemaske.

- 
2. Der Software-Agent ruft die unter dem angegebenen Namen gespeicherten Referenzbilder.
  3. Extrahiert aus den hinterlegten Referenzbilder die Objektmerkmale.
  4. Erstellt das Referenzmodell aus den Merkmalen.
  5. Der Software-Agent speichert das Referenzmodell unter dem angegebenen Namen.
  6. Der Software-Agent ordnet das Referenzmodell unter dem angegebenen Typen ein.
  7. Der Software-Agent speichert die zu dem Referenzmodell hinterlegten Metainformationen.
- 

Tabelle 3.7.: Anwendungsfall: Referenzmodell erstellen

### 3.3. Anforderungen an die Benutzeroberfläche

Von dem System wird erwartet, dass es eine Benutzeroberfläche (User Interface) anbietet. Dem Benutzer soll damit die Möglichkeit geschaffen werden, über eine Eingabemaske Serviceanfragen an das System zu stellen.

Dabei soll das Interface einfach zu bedienen sein und bei der Benutzung Spaß machen. So sollte der Benutzer nicht mit der Bedienung überfordert sein, die Bedienung soll intuitiv und selbsterklärend sein. Daher ist der Aufbau der Bedienoberflächen einfach zu halten und dient so der Übersichtlichkeit. Die Bedienoberfläche ist auf die wesentlichen Funktionalitäten zu beschränken, um dem Benutzer eine überschaubare Darstellung anzubieten.

Im folgenden werden die Anforderungen für die einzelnen Szenarien detailliert.

#### Objekte im System hinterlegen

- Eine Eingabemaske für die Aufnahme der Bilder.
- Eine Kamera für die Bildaufnahmen, die mit dem Software-Agenten zusammenarbeitet.
- Ein vordefinierter Abstand zwischen der Kamera und dem Objekt wird vom Nutzer eingehalten.
- Das Objekt im dem Referenzbild auf der Eingabemaske wird mit einem grünen Rahmen umrandet.
- Ein Button zum Speichern von einzelnen Bildern.
- Ein Button zum Erstellen von Referenzmodellen.
- Das System muss aus den Referenzbildern und den Eingaben von Informationen das Objektmodell erstellen und speichern.
- Fehlermeldung, falls keine Namen eingetragen ist.
- Fehlermeldung, falls das Bild nicht von der Kamera erfasst wird.
- Name bei der Eintragung überprüfen, um Duplikate zu verhindern.

**Service-Frage starten**

- Eingabemaske für die Selektion von Objekten.
- Auswahl bereits hinterlegter Objekte.
- Anzeige der Bilder zu den Objekten in der Eingabemaske.
- Einen Button „Start/Stop“ zur Verfügung stellen.
- Suche starten in den Bildern der Kamera.

**Orten von Objekten in Bildern und Raum**

- Das System durchsucht die Wohnung.
- Das System berechnet den Bildpunkt des Objektes aus Bildern.
- Das System berechnet den Raumpunkt zu den Bildpunkten.
- Das System hat Zugriff auf ein 2D-Modell des Living Place Hamburg .
- Es hat zur Anzeige der Position Zugriff auf die Monitore des Living Place Hamburg .
- Es gibt eine Verbindung zu den externen Diensten.
- Das System kann aus einer Liste von Objekten die prominenten selektieren.
- Das System hat Zugriff auf einen Sprachausgabe-Service.
- Es formuliert einen Ausgabesatz.
- Es entscheidet, ob eine Sprachausgabe stattfinden soll oder nicht.

## 3.4. Anforderungen

Das Ziel dieser Masterarbeit ist ja, ein Konzept zu erstellen, das die beschriebenen Anwendungsfälle integriert. Bevor ein Konzept für die Umsetzung eines Prototypen zur Realisierung der Szenarien aufgestellt werden kann, müssen aber die Fragestellung und die Problemstellung erfasst werden.

In dieser Abschnitt wird nun ausgeführt, welche Vorüberlegungen für die Modellierung des Konzeptes getroffen wurden.

### 3.4.1. Technische Anforderungen

Analysiert man die Szenarien und Anwendungsfälle in Hinblick auf technische Funktionalitäten, so lässt sich aus den Beschreibungen folgendes ableiten.

Um dem Bewohner eine Anfrage zu ermöglichen, muss ihm eine Auswahl von Objekten zur Verfügung gestellt werden. Dies setzt voraus, dass diese Objekte im System hinterlegt sind. Daraus ergibt sich wiederum, dass das System einen Software-Agent en zur automatischen Erfassung von Objekten zur Verfügung stellen muss.

Ein weiterer Prozess, der sich aus dem Szenario ergibt, ist die Ortung von Objekten in Bildern und im Raum. Diese Prozesse bringen einige Herausforderungen mit sich und werden aufgeteilt in (i) die Suche bzw. Wiedererkennung von Objekten in den Kamerabildern und (ii) das Lokalisieren von Objekten im Raum, d. h. der realen Umgebung im Living Place Hamburg .

Die Analyse hat hier also ergeben, dass ein Software-Agent für die automatische Wiedererkennung von Objekten in Bildern der Kameras benötigt wird und ein weiterer Software-Agent für das Orten oder Lokalisieren von Objekten im Raum.

Für die sprachliche Beschreibung der Lage eines Objektes im Raum wird einen weiterer Software-Agent benötigt, der einen adäquaten Satz formulieren und diesen in einen Text überführen kann, um die Sprachausgabe zu ermöglichen.

**Externe Schnittstellen** Folgende Anforderung an das System sind die Grundvoraussetzung dafür, dass diese Masterarbeit die Technologien und Sensor-Netzwerke im Living Place Hamburg ergänzen kann:

Da diese Masterarbeit ein Teilgebiet des Living Place Hamburg behandelt, muss eine Anbindung an die zentrale Kommunikationsschnittstelle implementiert werden, um einen Zugriff auf die externen Aktoren und Sensoren im Living Place Hamburg zu haben.

Damit das Szenarien „Objekt Anzeigen“ realisiert werden kann, muss das System Zugriff auf die im Living Place Hamburg integrierten Monitore und Gerätschaften (z. B. Fernshen, Multitouch) haben. Außerdem muss es Zugriff auf die im Living Place Hamburg montierten Kameras haben, denn aus dem Anwendungsfall Objektsuche ergibt sich die Notwendigkeit, eine Schnittstelle zu den Kameras zu haben, die den Zugriff auf die Video-Streams der Kameras ermöglicht. Es muss zudem jederzeit möglich sein, auf die Kamera-Eigenschaften zuzugreifen und an die Kameras bestimmte Steuerbefehle zu versenden, um das Schwenken, Zoomen und Neigen zu kontrollieren. Die extrahierten Anforderungen werden im Abschnitt ?? aufgeführt.

### 3.4.2. Qualitative Anforderungen

Analysiert man die Anwendungsfälle in Hinsicht auf qualitative Anforderungen, so lässt sich folgendes sagen: Eine der wichtigste Anforderungen an das System, welche in den Szenarien auch sehr deutlich wird, ist die Schnelligkeit. Die Antwortzeit auf eine Service-Frage muss sehr gering sein.

Weitere qualitative Anforderungen wie z. B. Wartbarkeit, Zuverlässigkeit, Veränderbarkeit, Benutzerfreundlichkeit, Zuverlässigkeit und viele andere mehr werden nicht aufgeführt, da diese als Grundvoraussetzung einer Anwendung in einem „Smart Home“ angesehen werden können.

Im Folgenden wird deshalb überwiegend auf die funktionalen Anforderungen eingegangen und die Frage- und Problemstellung entsprechend diskutiert.

### 3.4.3. Funktionale Anforderungen

**Service-Anfrage verstehen** Das kamerabasierte Objektsuch-System muss dem Bewohner eine Benutzerschnittstelle zur Verfügung stellen, über die er Service-Anfragen stellen kann. Das kamerabasierte Objektsuch-System muss verstehen, wonach der Bewohner im Raum sucht.

**Orten von Objekten in Bildern** Hieraus ergibt sich die Anforderung, Objekte in den Bildern erfassen, erkennen und voneinander unterscheiden zu können. Das System muss ein Objekt in den Bildern der Kameras unabhängig von dessen Lage und von allen Seiten als dasselbe Objekt identifizieren können. Das System muss außerdem anhand der Video-Streams der Kameras die Position des Objektes ermitteln können.

**Orten von Objekten im Raum** Das System muss die Position und die Lage des Objekte im Raum lokalisieren, konkret also den Bildpunkt auf den Raumpunkt abbilden können.

**Prominente Objekte bestimmen** Die Beschreibung der Lage des Objektes im Raum muss in Bezug auf den auf den Bewohner im Living Place Hamburg und relativ zu weiteren Objekten in der Umgebung stattfinden. Dafür müssen die Objekte in der unmittelbaren Umgebung des geortetes Objekte im Raum erkannt und identifiziert werden. Aus der Liste der erkannten Nachbarobjekte müssen in einem weiteren Schritt diejenigen Objekte ausgewählt werden, die für die Bezugnahme geeignet sind (sog. prominente Objekte).

Diese prominenten Objekte müssen dann in Bezug zueinander gebracht werden. Für die Beschreibung der Lage des Objektes im Raum in Bezug auf den Bewohner muss herausgefunden werden, wo sich dieser befindet, d. h. es muss räumliche Informationen über den Aufenthalt der Bewohner im Living Place Hamburg geben.

**Sprachliches Beschreiben der Lage des Objektes im Raum** Die räumliche Beschreibung der Lage des Objekte im Raum muss in einem dem Menschen gebräuchlichen Satz zu formulieren sein.

Im Folgenden werden die obengenannte Punkte im Detail diskutiert.

### **Service-Anfrage verstehen**

Dieser Abschnitt listet die Anforderungen auf, die sich aus der Fragestellung ergeben: „Wie kann eine Anfrage der Bewohner verstanden werden?“. Dies nimmt Bezug auf den Anwendungsfall: „Service-Anfrage an das System stellen“ [3.2.1](#).

### **Anfrage eingeben**

Eine erste Einschränkung des kamerabasierten Objektsuch-Systems ist, dass keine Spracherkennung implementiert wird, so dass auch keine Spracheingabe möglich ist, dies würde der Rahmen dieser Arbeit bei weitem sprengen.

Folglich ist für die Anfrage eine Eingabeschnittstelle notwendig; die aber so gestaltet sein soll, dass fast alle Aspekte einer Spracheingabe berücksichtigt werden.

Daraus ergeben sich die folgenden Anforderungen: Die Objektauswahl soll nicht nur über den Namen, sondern auch über Eigenschaften, Merkmale, Kategorien und weitere Metainformationen eines Objektes möglich sein.

### Anfrage verstehen

Zunächst einmal ist zu klären, was unter einer Serviceanfrage verstanden wird und welchen Aufbau und Struktur sie haben muss, um vom System abgebildet werden zu können.

Eine Serviceanfrage ist definiert als eine Anfrage an das System, in der ein spezifisches Objekt aus der Wohnung genannt ist, nach dem gesucht werden soll, sie ist also ein Fragesatz, der mit Formulierungen wie „Wo ist “ oder „Wo befindet sich“ beginnt. Sie enthält zumindest den Namen eines Objekts.

In dem Szenario 3.2.1 lautet die Frage: „Wo ist das UML2-Buch“, sie ist sehr allgemein gehalten, von der Bezeichnung des gesuchten Gegenstandes einmal abgesehen.

Die Serviceanfrage enthält im allgemeinen Fall eine Beschreibung, welches Objekt gesucht werden soll, welche Eigenschaften dieses Objekt besitzt, in welche Beziehung das Objekt zu dem Bewohner steht und welche kognitive Eigenschaften das Objekt impliziert.

Eine Anforderung an das System ist, dass der Bewohner die Möglichkeit haben soll, die Suche auch anhand beschreibender Merkmale und Attribute eines Objekts in Auftrag zu geben.

Nun bietet die natürliche Sprache sehr viele Möglichkeiten, ein und dasselbe Objekt durch verschiedenen Satzkonstruktionen und mit unterschiedlichen Attributen zu beschreiben.

In Folgenden sind einige mögliche Sätze aufgeführt, die dasselbe Buch betreffen.

- „Wo ist das UML-Buch?“
- „Wo ist mein Lehrbuch?“
- „Wo ist das Buch von dem Autor . . . mit dem Titel . . . ?“
- „Wo ist das Lehrbuch mit den Titel ‚SE 2‘ ?“
- „Wo ist das Software-Buch mit den Titel ‚SE 2‘ ?“

Die oben aufgeführten Satzkonstruktionen einer Fragestellung machen deutlich, dass Überlegungen getroffen werden müssen, wie ein Satz konstruiert sein muss, um vom System verstanden zu werden.

Um die Komplexität der Serviceanfragen einzuschränken, muss die Beschreibung eines Objektes eingeschränkt werden. Es muss auch festgelegt werden, welche Serviceanfragen zugelassen sind.

Nun haben Analysen ergeben, dass das Hinzufügen von beschreibenden Attributen eines Objektes eine individuell anpassbare Suchanfrage ermöglicht. Die Anforderung an das System ist folglich, die Serviceanfragen so aufzubauen, dass mit wenigen Attributen ein bestimmtes Objekt, nach dem gesucht werden soll, eindeutig zu beschreiben ist.

Zusammengefasst ergeben sich folgende Fragen, die geklärt werden müssen, bevor die Serviceanfrage „Wo ist das XYZBuch?“ vom System verstanden werden kann:

- Was bedeutet ein „Buch“?
- Welches Buch ist gemeint?
- Wie sieht ein Buch aus?
- Wie soll das System wissen, was gesucht wird?
- Welche Worte können als Auslöser für die Suche nach einem Objekt im System agieren?
- Wie kann ein Objekt als Input definiert werden?
- Wie kann die Serviceanfrage einen Such-Prozess starten?

Aus den Fragestellungen ergibt sich, dass das System gleichsam lernen muss, was Objekte bedeuten, welche Eigenschaften ein Objekt kennzeichnen, worin sich Objekte unterscheiden und wie Objekte bezeichnet werden. Es muss dann eine eindeutige Zuordnung der Bezeichnungen und Eigenschaften zu den Objekten stattfinden.

Weiterhin muss untersucht werden, welche Lösungen die Computer Vision bietet, um Objekte in Bildern zu identifizieren und voneinander zu unterscheiden.

Heute gibt es bereits einige Algorithmen und Methoden zum maschinellen Erlernen von Objekten. Einige der bestbewerteten Verfahren für die Zuordnung von Objekten und Namen sind Verfahren mit neuronalen Netzen. Es gibt auch viele Projekte, in denen Maschinenlernen mit großem Erfolg eingesetzt wird. Die dabei verwendeten Algorithmen sind aber sehr speziell der Anwendungssituation angepasst und bieten keine allgemeine Lösung.

Die Verwendung solcher Algorithmen in dieser Masterarbeit hätte den zeitlichen Rahmen der Umsetzung ebenfalls weit überschritten und wird daher hier auch nicht weiter verfolgt.

Die Anforderung und Herausforderung liegt nun darin, eine Lösung zu entwickeln, die wenigstens die Eigenschaften bzw. den Charakter eines Lernverfahrens besitzt.

### **Anforderung an das Orten von Objekten in Bildern**

Dieser Abschnitt führt die Anforderungen auf, die sich aus dem Anwendungsfall „Objekt-Suche“ 3.2.1 ergeben.

Diese sind: das System soll Objekte unabhängig von ihrer Lage finden, außerdem auch bei Teilverdeckung, wenn es z. B. schräg oder auf dem Kopf auf dem Sofa liegt oder von einer Flasche oder einem anderen Buch verdeckt wird, und auch, wenn es seine Position verändert hat.

Ortung von Objekten in Bildern bedeutet bekanntlich einen Vergleich von Bildeigenschaften und Bildmerkmalen, um Korrespondenzen zueinander zu finden. Um also Objekte in Bildern zu verorten, muss zunächst analysiert werden, welche Voraussetzung das System erfüllen muss, um korrespondierende Eigenschaften in Bildern wiederzufinden und so ein Objekt in Bildern zu erkennen.

Näherhin muss analysiert werden, wie Objekte in Bildern wiedererkannt werden können, welche Eigenschaften Bilder haben müssen, damit man aus ihnen gute Objekt-Eigenschaften extrahieren kann und welche Eigenschaften dabei die Kamera besitzen muss.

Um Objekt-Eigenschaften wie z. B. Formen, Größe und Texturen eines Objektes aus den Bildern zu extrahieren, werden Bildverarbeitungs-Algorithmen und -Methoden verwendet. Es ist zu klären, welche davon für das System geeignet sind.

Grundsätzlich ist von den Bildverarbeitungs-Algorithmen und -Methodiken zu fordern, dass sie robust sind gegenüber

- Rotation,
- Skalierung und
- Translation, sowie der
- Änderung von Lichtverhältnissen im Raum;
- dass sie eine Wiedererkennung auch bei hoher Teilverdeckung ermöglichen
- und zudem schnell bei der Wiedererkennung von Objekten sind,

um so die Anforderungen zu erfüllen zu können.

Nun muss noch für den Anwendungsfall „Referenzmodell erstellen“ 3.2.2 analysiert werden, welche Eigenschaften erfüllt sein müssen, damit ein Referenzmodell aus den Referenzbildern erstellt werden kann.

In dem in Abschnitt 2 beschriebenen Verfahren der Merkmalsextraktion und Objekt-Wiederfindung wird das Referenzmodell aus den Merkmalen der Referenzbilder erstellt.

Die Verfahren der Merkmalsextraktion bringen bestimmte Anforderungen mit sich, die erfüllt sein müssen, um robuste und eindeutige Merkmale aus den Bildern zu extrahieren:

- Beim Hinterlegen der Referenzbilder sollte darauf geachtet werden, dass nur das Objekt in dem Bild zu sehen ist, für den das Referenzbild hinterlegt werden soll.
- Bei den Aufnahmen sollte ein neutrales Licht im Raum gegeben sein. Da bei der Merkmalsextraktion die Konturen des Objektes im Bild erkannt werden müssen, darf das Bild nicht zu dunkel werden, sonst würden die Kanten und Ecken des Objekte und somit auch die Punkte nicht erkannt. Daraus folgt die Anforderung, den Benutzer davon in Kenntnis zu setzen, wenn die Lichtverhältnisse nicht ausreichen, um robuste Merkmale zu extrahieren.

Aus dem Anwendungsfall „Referenzbilder hinterlegen“ [3.2.2](#) ergibt sich, dass

- viele Bilder hinterlegt werden können und
- die zu einem Objekt hinterlegten Bilder jederzeit geändert, gelöscht und erweitert werden zu können.

**Objekterfassung und Bildaufnahme** Für die Extraktion von Eigenschaften eines Objekts in einem Bild muss gewährleistet sein, dass von der Kamera nur ein Objekt erfasst wird, deshalb sollte nur ein Objekt im Fokus der Kamera sein.

Es muss geklärt werden, mit welcher Auflösung robuste und gute Merkmale eines Objektes extrahiert werden. Auch die Lichtverhältnisse bzw. die Belichtung des Objektes während der Bildaufnahme in einer Szene haben Einfluss auf die Extraktion von Objektmerkmalen. Es muss deshalb herausgefunden werden, unter welchen Lichtverhältnissen die Referenzbilder am besten erstellt werden, um robuste Merkmale extrahieren zu können.

Ein weiterer Aspekt der Bildverarbeitung, welcher einen Einfluss auf die Extraktion von Eigenschaften der Objekte aus den Bildern hat, ist die Kameraperspektive, der Blick der Kamera auf das Objekt in einer Szene. Man unterscheidet zwischen der Frosch-, der Normal- und der Vogelperspektive.

Auch die Entfernung des Objekts von dem Kameraobjektiv, die sogenannte Einstellgröße, hat einen Einfluss auf die Extraktion von Eigenschaften und Merkmalen eines Objektes aus den Bildern. Der Abstand des Objektes von der Kamera hat Auswirkung auf die darstellende Größe des Objektes. Diese wird durch die Einstellgröße bestimmt, die auch Einfluss auf den Informationsgehalt des Objektes in den Bildern hat. Ein weit entferntes Objekt sieht klein aus und liefert wenig Informationen z. B. über die Textur des Objektes und auch die Größe des Objektes weicht stark von der tatsächliche Größe des Objektes ab.

Auf weitere Veränderungen bei sich bewegenden Objekten oder Bewegungen der Kamera wird nicht eingegangen, da in dieser Arbeit davon ausgegangen wird, dass die Kameras fest an der Wand montiert sind und dass die Objekte sich während der Suchprozesse in Ruhe befinden.

**Objekt erkennen** Aus dem Anwendungsfall „Objekt suchen“ 3.2.1 ergibt sich die Anforderung, dass die Objekte von allen Seiten identifiziert werden müssen, d. h. ein Buch muss von allen Seiten betrachtet immer dasselbe Buch bleiben und beim Erfassen der Bilder darf ein Buch nicht als zwei verschiedene Bücher identifiziert werden.

Das kamerabasierte Objektsuch-System muss an Hand verschiedener Bilder einer Szene ein Objekt aus jeder Perspektive als dasselbe Objekt wiedererkennen.

Vorher muss wie im Anwendungsfall „Referenzbilder hinterlegen“ 3.2.2 beschrieben, dem kamerabasierten Objektsuch-System beigebracht worden sein, wie das betreffende Objekt aussieht.

Das kamerabasierte Objektsuch-System muss die Bilder den Objekten zuordnen.

Schließlich muss das System in der Lage sein, anhand der Namen ein Objekt zu identifizieren. Hierfür muss auch eine Zuordnung der Namen zu den Objekte stattfinden.

**Sichtbare Objekte** Für die Ortung des Objektes muss folgende Fragestellung betrachtet werden: Welche Objekte können verortet werden? Was passiert, wenn das zu suchende Objekt nicht sichtbar ist?

Diese Fragen implizieren folgende Fragestellungen, für die ein Lösungsweg gefunden werden muss:

- Was muss das kamerabasierte Objektsuch-System tun, wenn das Objekt von einer Kamera nicht wiedergefunden wird?
- Was muss das kamerabasierte Objektsuch-System tun, wenn das Objekt nicht im Fokus der Kamera ist?
- Was muss es tun, wenn alle Kameras das Objekt nicht wiederfinden? Und wann soll festgestellt werden, dass das Objekt nicht in der Wohnung zu finden ist?

Das kamerabasierte Objektsuch-System muss analysieren, in welchen Fällen das Objekt nicht im Raum zu finden ist, genauso wie es entscheiden muss, in welchen Fällen das Objekt als eindeutig identifiziert anzusehen ist. Ist das Objekt nicht im Raum oder wird es oder konnte es nicht von der Kamera erfasst werden, so muss eine Lösung gefunden werden.

Das kamerabasierte Objektsuch-System muss schließlich definieren, welche Objekte überhaupt gefunden werden können und welche nicht.

### **Einschränkungen beim Orten von Objekten**

Die erste Einschränkung, die sich ergeben hat, ist die, dass nur solche Objekte gesucht werden können, für die Referenzbilder hinterlegt sind.

Eine zweite Einschränkung besteht darin, dass die Suche auf bewegliche Gegenstände in der Wohnung eingeschränkt werden muss. Als beweglich gilt ein Objekt, wenn es eine bestimmte Größe nicht überschreitet. Grob gesagt fordert man, dass die zu suchenden Objekten tragbar sein sollen. Durch diese Einschränkung wird das System schlank gehalten, denn die wichtigste Anforderung an das System ist seine Schnelligkeit. Diese Einschränkung schließt solche Gegenstände aus, die quasi statisch in die Wohnung integriert sind. Sie haben die Eigenschaft, dass sie durch ihre Form, Größe und Eigenschaften einem Bereich in der Wohnung eine bestimmte Bedeutung bzw. Funktionalität zuordnen.

Eine weitere Einschränkung ist, dass das kamerabasierte Objektsuch-System nur solche Objekte im Raum finden können soll, die nicht zu stark abgedeckt sind.

### **Anforderung an das Orten von Objekten im Raum**

Ein Objekt im Bild zu erkennen ist noch nicht gleichbedeutend mit seiner Ortung in seinem realen Umfeld.

Die Anforderung aus dem Anwendungsfall „Pose ermitteln“ in Abschnitt 3.2.1 ist es, Lösungen zu wählen, die die Pose sowohl anhand einer Kamera als auch mit mehreren Kamera bestimmen können, denn der Living Place Hamburg ist nicht in allen Bereichen mit zwei Kameras ausgestattet.

### **Die Kamerasteuerung**

Die Integration der PTZ-Kamera im Living Place Hamburg [ vgl. A] stellt den Entwickler vor eine zusätzliche Herausforderung.

Denn bevor die Raumpunkte über mathematische Verfahren, welche in dem Abschnitt 2.3 vorgestellt wurden, ermittelt werden können, müssen die Kameras in jeder beliebigen Position positioniert werden können.

Es muss also eine Anwendung für die Steuerung der Kameras zu erstellen, die die Steuerungsfunktionen Zoomen, Schwenken und Lenken automatisch einstellt und steuert.

Laut Hersteller unterstützen die Kameras viele standardisierte Protokolle und standardisierte Schnittstellen für den Zugriff auf eine IP-Kamera. Es ist zu analysieren, wie über diese Schnittstellen die einzelnen Kameras angesprochen werden können.

Für die technische Umsetzung muss herausgefunden werden, welche Schnittstelle sich dazu eignet, den Status der Kameras jederzeit abzufragen und jede einzelne Kamera in variabler Geschwindigkeit in jede Position zu schwenken, zu zoomen und zu lenken.

Auf dieser Grundlage muss ein Planungs- und Koordinationsalgorithmus für das Durchsuchen der Wohnung erstellt werden. Welche Anforderungen und Aspekte dabei beachtet werden müssen, wird im folgenden Abschnitt dargestellt.

### **Planungs- und Koordinationseinheit**

Nachdem die Steuerung der Kameras realisiert wurde, muss für das Durchsuchen des Wohnbereichs noch ein Planungs- und Koordinationsalgorithmus erarbeitet und implementiert werden. Seine Aufgabe ist es, zu entscheiden, wann, welche und wie viele PTZ-Kameras im Living Place Hamburg bei einer Suchanfrage beauftragt werden. Dabei sollen möglichst wenige Video-Sequenzen und Kameras für das Durchsuchen der Wohnung herangezogen werden. Der Entscheidungsalgorithmus sollte zudem adaptiv sein, d. h. sich an unterschiedliche Bedingungen und Kontexte anpassen können. Er sollte vermeiden, dass ein Bereich der Wohnung von zwei Kameras gleichzeitig durchsucht wird, um so effizient und schnell ein Ergebnis zu erzielen.

Nach der Implementierung und Einbindung solch einer Planungs- und Koordinationseinheit in das Gesamtsystem sind weitere Analysen zu erarbeiten, um Lösungswege für folgende Fragestellungen zu finden:

- Inwieweit müssen die Kameraeinstellungen angepasst werden, um herauszufinden, ob sich das gesuchte Objekt im Suchbereich der jeweiliger Kamera befindet?
- Ab welchem Zeitpunkt wird die Suche einer Kameras als Misserfolg gewertet?
- Mit welchen Einstellungen erzielt die Kamera den besten Sucherfolg?
- Wie und Wann werden die Kameras benachrichtigt, wenn ein Objekt im Raum gefunden wurde?

### **Anforderungen an die sprachliche Beschreibung**

Eine weitere Herausforderung, die sich aus den Szenarien ergibt, ist die sprachliche Beschreibung der Pose des Objektes im Raum, welche in dem Anwendungsfall ?? dargestellt ist.

Für diese Beschreibung der Lage eines Objekts in einem sprachlich lokalisierenden Ausdruck muss die Kommunikation zwischen Mensch und Maschine betrachtet werden, insbesondere hinsichtlich der Fragestellung: Wie orientieren sich Menschen in Räumen und welche kognitive Prozesse laufen in den Köpfen der Menschen ab, wenn Sie die Lage des Objektes räumlich beschreiben?

Einige der der Forschungsfragen, die hier als erstes Betracht gezogen werden müssen, sind folgende:

- Wie wird die räumliche Beschreibung der Lage eines Objektes in einem dem Menschen verständlichen Satz formuliert?
- Wie können räumliche Informationen von einem System erlangt werden?
- Wie kann eine räumliche Beziehung von einem System modelliert werden?
- Wie können die räumlichen Informationen semantisch interpretiert werden?
- Wie kann entschieden werden, welche Referenzobjekte sich für die Beschreibung eignen. Wie können Objekte diesbezüglich klassifiziert werden?
- Wie kann ein lokalisierender Antwortsatz definiert werden, so dass der Bewohner die Position des gefundenen Objektes im Raum erkennt?

Das bildorientierte Objektsuch-System erfasst und liefert die Position des gesuchten Objektes (im weiteren „Locatum“) im Raum. Um die Lage des Locatums (in dem Szenario ist es das Buch) in einen lokalisierenden Ausdruck (z. B. „Das Buch liegt auf dem Tisch“, „das Buch ist links, neben dem Telefon“, „das Buch ist im Wohnzimmer“) zu überführen, ist eine Analyse der Szene bzw. Bestimmung des Kontextes in der Wohnung erforderlich. Einige Aspekte, die hierbei betrachtet werden müssen, sind:

- das Erkennen und Wahrnehmen der weiteren Objekte in der Umgebung des Locatums über ihre räumlichen Eigenschaften (Größe, Form, Gestalt). Diese sind notwendig für die Bestimmung eines „Relatums“ (eines zweiten Objekts, das als Bezugspunkt für das Locatum der Raumorientierung dient);
- die Bestimmung der relativen Lage des Locatums in Bezug zum Relatum.

Befindet sich das Objekt in unmittelbarer Umgebung des Betrachters, soll die Lage des Objekts mit Lokalisationsäußerungen wie „Es liegt vor/hinter dir“, „Es befindet sich links/rechts von dir“ beschrieben werden.

Dies erfordert

- die Identifizierung des Betrachters im Raum. Es muss festgestellt werden, welcher Bewohner in der Wohnung den Suchauftrag für das Kamerasystem gegeben hat und somit den Betrachter darstellt;

- das Erkennen und Wahrnehmen des Locatums in Abhängigkeit vom Betrachter. Hierzu muss der Standort des Betrachters in der Wohnung sowie seine Blickrichtung bestimmt werden.

Für die Umsetzung wurde die Festlegung getroffen, dass die Beschreibung der Pose unabhängig von dem Betrachter stattfindet.

Für das System ergibt sich die Anforderung, weitere geeignete Objekte (die sogenannten prominenten Objekte) in der Nähe des gesuchten Objektes zu erkennen. Es muss ein Bezugssystem bestimmen und für die Beschreibung eine Perspektive wählen.

Anhand eines Beispiels soll nur einige Aspekte einer sprachlichen Ausgabe verdeutlicht werden.

**Szenen-Beschreibung:** Ein Buch liegt auf dem Tisch. Links von dem Buch befindet sich eine Tasse. Die Position des Buches in dieser Szene soll dem Bewohner sprachlich beschrieben werden.

Auf die Frage „Wo ist das Buch?“ wären folgende Beschreibungen in Abhängigkeit von weiteren Nachbarobjekten möglich; dabei wären der Tisch und die Tasse die weiteren Objekte in der Nähe des Buches:

- Das Buch befindet sich neben der Tasse.
- Das Buch befindet sich links von der Tasse.
- Das Buch befindet sich auf dem Tisch.
- Das Buch befindet sich auf dem Tisch, neben der Tasse.

In Abhängigkeit vom Nachbarobjekt wurden die entsprechenden lokalisierenden Ausdrücke wie „neben“, „links“ oder „auf“ gewählt.

Hieraus wird deutlich, dass für ein computergestütztes Sprachausgabesystem viele Hürden zu überwinden sind, um einen für den Menschen verständlichen Satz zu formulieren

Für die sprachliche Lokalisierung eines Objektes im Raum muss der Zusammenhang zwischen den räumlichen und sprachlichen Strukturen betrachtet werden.

In der Forschungsgebiet Raumkognition beschäftigen sich die Wissenschaftler u. a. mit der Frage: *Welche kognitiven Prozesse finden statt, wenn wir uns in einem Raum bewegen, uns im Raum orientieren, jemandem einen Weg oder die Lage eines Gegenstandes sprachlich beschreiben?*

Im Abschnitt 3.5.2 wird erläutert, welche weiteren Anforderungen sich aus der Raumkognition für das kamerabasierte Objektsuch-System ergeben.

## 3.5. Related work / Aktuelle Forschungsprojekte

In dieser Abschnitt werden diejenigen Forschungsprojekte an der HAW-Hamburg vorgestellt, die sich ebenfalls mit der Problemstellung und den Anforderungen im Kontext dieser Arbeit beschäftigt haben.

Diese Arbeit profitiert dabei von den gewonnen Erkenntnissen und Lösungen der Forschungsprojekte und Forschungsbereiche und lässt sie in ihre Konzeption mit einfließen.

Das erste Themengebiet dieser Masterarbeit kommt aus dem Bereich der Computer Vision. Hiermit hat sich Carsten Fries im Zusammenhang mit der Modellierung von Objekten und ihrer Erkennung beschäftigt. Insbesondere ist die Objektmodellierung und -erkennung sowie Objektverfolgung sein Thema.

Es gibt auch zahlreiche andere Beispiele aus der Industrie und Forschung zum Thema Computer Vision und insbesondere die Gebiete der Objekterkennung und Objektverfolgung, doch beschränkt sich die Übersicht in dieser Arbeit auf die an der HAW-Hamburg entwickelten Systeme und Forschungsprojekte.

### 3.5.1. Masterarbeiten mit der SURF-Implementierung

Henrik Braue entwickelte in seiner Masterarbeit [Brauer (2010)] mit dem Titel „Entwicklung eines Augmented Reality Frameworks auf Basis von Kamera-basierten Trackingverfahren“ ein Framework, mit dessen Hilfe Augmented-Reality-Anwendungen erstellt werden können.

Für die Umsetzung entwickelte er einen Algorithmus, der eine Kombination aus Speeded Up Robust Features (SURF) und der Kanade–Lucas-Tracker (KLT)[Lucas (1981)] darstellt. Er ermöglichte es ihm, Augmented-Reality-Anwendungen in Echtzeit zu betreiben.

Diese Arbeit ist auf Grund der Verwendung eines markerlosen Verfahrens und einer SURF-Implementierung von Interesse für die Entwicklung eines Prototypen zur Suche nach Objekten.

Die Tests (vgl. Brauer, 2010, S. 71–75) haben gezeigt, dass die Kombination aus SURF und KLT eine gute Lösung darstellen und Merkmale zur Bestimmung der Position eines Objektes auch unter erschwerten Bedingungen wie z. B. Drehung um 180 Grad, Kippen und teilweiser Verdeckung festgelegt werden können.

Er zeigt auch, dass normale Veränderungen der Beleuchtung kein Problem darstellen, um ein Objekt zu erkennen und zu verfolgen, nur sollte die Beleuchtung nicht zu gering sein. Weiterhin zeigt er in seiner Arbeit, dass auf Basis eines 3D-Modells des Objektes und mit bildbasierten Algorithmen die Position im Raum bestimmt werden kann.

Hieraus wurden als Ergebnisse gewonnen das Konzept der Umsetzung für die 3D-Objektmodellierung, vor allem aber für die schnelle Wiedererkennung der Objekte mit Kameras.

Ein weiteres Projekt, das den SURF-Algorithmus verwendet hat, ist die Masterarbeit von Carsten Fries ([Fries, 2011](#)) mit dem Titel „Kamerabasierte Identifizierung und Lokalisierung von Gegenständen für flexible Roboter“.

Darin beschäftigte er sich mit dem Identifizieren, Lokalisieren und Greifen von Gegenständen für den Katana-Greifarm. Der Katana-Greifarm integriert eine Kamera, mit der er seine Umgebung erfasst. Karsten Fries entwickelte ein System, das es dem Katana-Greifarm ermöglicht, ein Objekt in seiner Umfeld zu erkennen und dessen Pose im Raum zu bestimmen. Es berechnet auf Basis der ermittelten Raumkoordinaten die Armbewegung und greift nach dem Gegenstand.

Für die Objekterkennung mit der Kamera hat er unter Verwendung des SURF-Algorithmus ein Modell des Objektes aus perspektivisch unterschiedlichen Aufnahmen erstellt und dieses dann in den Aufnahmen der Katana-Greifarm wiedererkannt. In nächsten Schritt wird aus der Bildposition die 3D-Raumposition bestimmt. Dann wird die Bewegung des Armes ermittelt und der Arm dahingehend gesteuert.

Die gestellten Anforderungen an das entwickelte System für den Katana-Greifarm decken sich bis zu einem bestimmten Punkt mit den Anforderungen an diese Masterarbeit. Er verwendet für die Wiedererkennung des Objekte die Kamera des Katana-Greifarms, in meiner Umsetzung werden die PTZ-Kameras im Living Place Hamburg verwendet.

Die gewonnenen Erkenntnisse aus diese Arbeiten betrafen die Modellierung des Objektes. Seine Ergebnisse haben gezeigt, dass die Erstellung eines Modell aus Bildern unterschiedlicher Perspektiven die Suche beschleunigt.

Zwischen beiden Projekten wurden Erkenntnisse und Erfahrungen darüber ausgetauscht, wie ein 3D-Modell für die Umsetzung dieser Masterarbeit erstellt werden kann und wie die Position im Raum ermitteln kann.

### **3.5.2. Beziehung zwischen Raum und Sprache**

In diesem Abschnitt geht es um die Frage: Wie sage ich es dem Bewohner? Dafür muss der Zusammenhang zwischen den räumlichen und sprachlichen Strukturen betrachtet werden. Dementsprechend stellt sich die weitere Frage: Wer hat sich in der Forschung mit der Beziehung zwischen Raum und Sprache beschäftigt?

Das Thema „Sprache und Raum“ ist auch Gegenstand der gegenwärtig stattfindenden interdisziplinären Diskussionen des Sonderforschungsprojekts SFB/TR 8 Spatial Cognition:<sup>1</sup> Reasoning, Action, Interaction (vgl. Abschnitt 3.5.2).

In dem Forschungsgebiet Raumkognition beschäftigen sich die Wissenschaftler u. a. mit den Fragen:

Welche kognitiven Prozesse finden statt, wenn wir uns in einem Raum bewegen, uns im Raum orientieren, jemandem einen Weg oder die Lage eines Gegenstandes sprachlich beschreiben?

**Kognition** In den unterschiedlichen Disziplinen der Wissenschaft wird der Begriff Kognition unter verschiedenen Blickwinkeln betrachtet und somit unterschiedlich definiert.

Es würde den Rahmen dieser Arbeit sprengen, jede Definition zu nennen und zu erläutern, daher sei hier nur die Definition eines Lexikons genannt: Kognition wird als Sammelbegriff für alle Prozesse und Strukturen, die mit Wahrnehmen und dem Erkennen zusammenhängen, definiert (z. B. Denken, Erinnern, Vorstellen, Gedächtnis, Lernen, Planen u. a.) [Brockhaus Lexikon]

**Kognitive Wissenschaft** Die kognitive Wissenschaft ist mit dem Anspruch angetreten, den menschlichen Geist in seiner vollen Komplexität zu erforschen.

Die Kognitionswissenschaft besteht aus verschiedenen Teildisziplinen. Eine der Teildisziplinen ist die Psycholinguistik, aus dieser entwickelte sich die Teildisziplin Raumkognition.

Eine weitere Teildisziplin der Kognitionswissenschaft ist die Psycholinguistik, die eine Brücke zwischen der Linguistik und der Psychologie darstellt. Sie wurde von Gert Rickheit und Hans Strohner in dem Buch „Grundlagen der kognitiven Sprachverarbeitung“ (vgl. Rickheit und Strohner, 1993) begründet mit dem Ziel, die menschliche Sprachverarbeitung mit ihren kognitiven und kommunikativen Aspekten in Experimenten zu erforschen.

**Raumkognition** In dem Bereich Raumkognition erforschen die Psycholinguisten in Experimenten mit unterschiedlichen Ansätzen, wie Menschen unterschiedliche Beschreibungen räumlicher Strukturen verstehen, die tagtäglich in Texten, Anleitungen, Wegbeschreibungen und ähnlichen verbalen Äußerungen enthalten sind (Vorweg, 2001).

Die Kognitionswissenschaftler versuchen mit Hilfe von Computermodellen die Prozesse der Kognition zu ergründen, um im Anschluss daran menschliche Sprache durch Maschinen generieren zu lassen.

---

<sup>1</sup>[Quelle: <http://www.sfbtr8.uni-bremen.de/index.html>]

In den Kognitionswissenschaften werden unter anderem Prozesse der visuellen und auditiven Wahrnehmung, der Problemlösung sowie die Mechanismen der Sprachbeherrschung in Hinblick auf Menschen und Maschinen thematisiert (Habel, 2001).

Es ist zu beobachten, dass die Untersuchungen zur Erforschung der kognitiven Fähigkeiten des Menschen in den letzten Jahren stark zugenommen haben.

### Sonderforschungsbereich TR 8 – Spatial Cognition

Ein aktuelles Forschungsprojekt der Raumkognition ist das Sonderforschungsprojekt „SFB/TR 8 Spatial Cognition - Reasoning, Action, Interaction“.<sup>2</sup> Dies ist ein aktuelles interdisziplinäres Forschungsprogramm der Universitäten Bremen und Freiburg, bei dem die Forscher unter anderem der Frage nachgehen, wie Menschen und Roboter Wissen über ihre räumliche Umgebung erwerben und verarbeiten, sich in ihrer Umgebung zurechtfinden und Informationen über ihre Umgebung austauschen können.

Aber auch Fragen wie die, was in unserem Gehirn passiert, wenn wir über den Raum nachdenken, sind Gegenstand der Forschung.

Finanziell gefördert wird der SFB/TR 8 Spatial Cognition seit 2003 von der Deutschen Forschungsgemeinschaft (DFG),<sup>3</sup> die nationale und internationale Kooperation wird unterstützt durch den Deutschen Akademischen Austauschdienst (DAAD).

Auf der Suche nach neuen Erkenntnissen und bei der Entwicklung neuer Formalisierungen für den Computer im Bereich Raumkognition findet ein regelmäßiger Austausch der neuesten Erkenntnisse und Forschungsergebnisse zwischen nationalen und internationalen Wissenschaftlern verschiedener Disziplinen (wie Biologie, Psychologie, Informatik, Linguistik usw.) statt.

Hier beispielhaft aufgeführt sind die letzten Themen und Fragestellungen, mit denen sich die Wissenschaftler in den Workshops beschäftigten.

- Um kognitive Fähigkeiten der Menschen besser zu verstehen und daran beteiligten Prozessabläufe besser zu verstehen, auf den Computer zu übertragen und diese zu optimieren, gehen die Wissenschaftler der Frage auf den Grund, was räumliche Fähigkeiten ausmacht und welche Rolle Navigatoren und Visualisierung dabei spielen.

Das **Ziel** der Forschung ist es, ein Verfahren zu entwickeln, das die kognitiven Fähigkeiten auf verschiedenen Stufen erfasst. Hierdurch sollen individuelle Unterschiede der kognitiven Fähigkeiten erforscht werden.

---

<sup>2</sup><http://www.sfbtr8.spatial-cognition.de/>

<sup>3</sup><http://www.dfg.de/index.jsp>

- Auf den internationalen und interdisziplinären Workshops und Konferenzen beschäftigten sich die Wissenschaftler mit dem Thema „Räumliches Verhalten und sprachliche Repräsentation (spatial behaviour and linguistic representation)“.

Es findet ein Austausch der neuesten Erkenntnisse und Methoden im Bereich menschliche Orientierung im Raum statt. Die Forscher analysierten, wie sich Menschen in unbekanntem Umgebungen verhalten. Sie untersuchten, wie Menschen räumliches Wissen sich aneignen und wie Menschen Räume verstehen und wahrnehmen; welche Strategien sie nutzen, um sich in Räumen zu orientieren und dort zu navigieren.

Das **Ziel** der Forscher ist es, eine Repräsentation und Modellierung von räumlichem Verhalten zu erstellen und die Beziehung zwischen den räumlichen Konzepten und sprachlichen Strukturen zu analysieren, um daraus Modelle und Formalismen zu definieren und diese auf den Computer übertragen.

Mit den Ergebnissen der Forscher werden bereits Navigationsgeräte entwickelt, die automatische Routenbeschreibungen generieren, die an die Denkweise des Menschen angepasst und dafür optimiert sind.

Die Forschungsergebnisse werden in den Sammelbänden der Reihe „Spatial Cognition“<sup>4</sup> veröffentlicht,<sup>5</sup> die jährlich alle wissenschaftlichen Untersuchungen und die Ergebnisse der Forschung zusammenfasst.

### Raumkognition

Die Raumkognition ist ein Teilgebiet der Kognitionswissenschaft. Sie beschäftigt sich mit der Fähigkeit des räumlichen Denkens. Hierzu gehören Raumwahrnehmung, Raumrepräsentation und Raumbeschreibung sowie der intelligente Umgang mit ihnen.

Die interdisziplinären Forschungen zur Raumkognition verfolgen unter anderem das Ziel, menschliche kognitive Fähigkeiten und Leistungen im Umgang mit dem Raum sowie der Orientierung in ihm genauer zu analysieren.

Die Beziehung zwischen den räumlichen Konzepten und sprachliche Strukturen wurde bereits mit der kognitiven Wende in den siebziger Jahren thematisiert (Rickheit und Strohner, 1993). In experimentellen Untersuchungen wurde gezeigt, dass sprachliches Verstehen ohne die Berücksichtigung des Weltwissens nicht möglich ist (Rickheit und Strohner, 1993).

Zahlreiche und intensive Untersuchungen zu Raumvorstellung und sprachlichen Strukturen haben gezeigt, dass Sprechen über gegenständliche Welten komplexe Übertragungsprozesse zwischen sprachlichen und nichtsprachlichen Repräsentationen voraussetzt.

---

<sup>4</sup><http://www.spatial-cognition.de/>

<sup>5</sup>Freksa u. a. (1998); Vorweg (2001); Freksa u. a. (2003, 2004); Barkowsky u. a. (2008); Freksa u. a. (2008)

Untersuchungen zur Erfassung des kognitiven Raums haben in den letzten Jahren stark zugenommen. In Forschungsprojekten und Forschungsverbänden werden Prozesse der Raumwahrnehmung und der räumlichen Repräsentationen bei Menschen experimentell untersucht.

### **Räumliche Informationen**

Die Forscher haben unter anderem die Erkenntnis geliefert, dass räumliche Wahrnehmung auf der Gewinnung von räumlichen Informationen basiert (Rickheit, 1999; Habel und von Sutterheim). Die Verarbeitung räumlicher Informationen hat innerhalb der menschlichen Kognition eine zentrale Stelle, denn räumliche Informationen bildet die Grundlage für das Erkennen und Kategorisieren von Objekten und Ereignissen durch die visuelle und haptische Wahrnehmung.

Interne räumliche Informationen werden auf verschiedene Weisen aufgebaut, einerseits durch haptische Sinneswahrnehmung, andererseits durch Kommunikation. Zur haptischen Sinneswahrnehmung gehören der Tastsinn, die Reaktion auf Temperatur, Vibration und Schmerz.

Miller und Johnson-Laid entwickelten die Theorie, dass reale Sachverhalte, die sprachlich beschrieben werden, im Arbeitsgedächtnis mental modelliert werden. Diese Theorie wurde durch die Untersuchungen und Ergebnisse weiterer Kognitionswissenschaftler untermauert (Rickheit, 1999).

Die Wissenschaftler haben herausgefunden, dass Menschen, bevor sie über Objekte und ihre räumlichen Eigenschaften bzw. ihre räumliche Beziehung zu anderen Objekten sprechen können, diese erst visuell, auditiv, taktil erfahren müssen. Hierdurch wird eine räumliche Repräsentation aufgebaut, die dann als Grundlage weiterer kognitiver Prozesse fungiert.

Ein Beispiel eines solchen kognitiven Prozesses ist das Bilden sprachlicher Ausdrücke (Rickheit, 1999; Habel und von Sutterheim).

Um über etwas zu sprechen, werden mehrere Repräsentationsebenen gebildet, die aufeinander aufbauen. Die Frage, wie viele und welche spezifischen mentalen Prozesse in unserem Gehirn stattfinden, ist Gegenstand der heutigen Forschung.

Im Folgenden werden die kognitiven Prozesse betrachtet, die sich für die sprachliche Lokalisierung eines Objekts im Raum ergeben haben. Diese wurde unter anderen von Vorweg & Rickheit (Rickheit, 1999; Habel und von Sutterheim) in der Kognitionswissenschaft untersucht. Ihre wesentlichen Züge sollen im Folgenden skizziert werden.

Im ersten Schritt der Visualisierung muss der Sprecher aus einer mentalen Repräsentation ein Objekt auswählen, das er lokalisieren möchte (Locatum) und ein weiteres Objekt als Bezugspunkt (siehe: Räumliche Bezugssysteme 3.5.2) bestimmen. Für die Konzeptualisierung

der räumlichen Relation muss eine Perspektivenwahl getroffen werden. (siehe 3.5.2). Dieser Zusammenhang zwischen der Blickpunktabhängigkeit und dem Bezugssystem wird in der Sprache reflektiert. Für die Orientierung im Raum ist weiterhin räumliches Wissen (siehe 3.5.2) erforderlich, welches weitere Anforderungen mit sich bringt.

### **Räumliche Bezugssysteme**

Die Wahrnehmung der Position eines Objekts findet relativ zu anderen Objekten statt. Die Position eines Objekts wird dabei durch die Distanz und die Richtung physikalisch und perspektivisch vollständig beschrieben und erfordert entsprechende Bezugssysteme.

Ein Bezugssystem ist definiert als eine Menge von Werten, zu denen ein zu beurteilender, wahrgenommener, erinnerter oder vorgestellter Wert in Beziehung gesetzt werden kann. Sie bieten Vergleichswerte, eine Art Maßstab für die Zuordnung einer Erscheinung, eines Dimensionswertes oder eines Ortes im Raum.

Die Orientierung in einer Umgebung an der eigenen Position (Bezugspunkt) wird als das *egozentrische* Bezugssystem bezeichnet. Für die Navigation oder die Orientierung in geographischen Räumen ist es allerdings unzureichend.

Nützlich ist hingegen die Orientierung an gut und leicht wiedererkennbaren sowie stabil vorhandenen Objekten.

Diese räumliche Bezugnahme auf die Umgebung wird oft als *exo-* oder *allozentrisches* Bezugssystem oder auch als *environment-centric* bezeichnet.

Beide Bezugssysteme nehmen ihrerseits Bezug auf das geozentrische Bezugssystem, das auf der Gravitationsrichtung basiert. Nord, Süd, Ost, West sind allozentrische Relationen in Bezug auf die Erdpole.

### **Blickpunktabhängigkeiten der Bezugssysteme**

Die Verbalisierung der Lage eines Objektes im Raum erfordert neben dem Verstehen der Lage auch die Berücksichtigung des Blickpunkts, der Perspektive, denn es stehen dem Sprecher und Hörer mehrere Möglichkeiten offen, ein Bezugssystem einzubeziehen. Es wird unterschieden zwischen der *Sprecher-*, *Hörer-* und der *kanonischen* Perspektive.

So kann ein Objekt, das vom Sprecher aus gesehen (Sprecherperspektive) vor einem Stuhl liegt, sich gleichzeitig vom Hörer aus gesehen hinter ihm befinden. Vom Stuhl aus betrachtet, könnte es links liegen und von der Eingangstür aus z. B. rechts.

Während eine Aussage wie „Sie steht am Fenster“ vom Blickpunkt des Betrachters unabhängig ist, hängt es vom Betrachter ab, ob die Angabe „vor dem Fenster“ oder „hinter dem Fenster“ zutreffend ist (kanonische Perspektive).

Bei der sprachlichen Lokalisierung eines Objekts werden wahrgenommene Richtungsrelationen und Distanzrelationen den lokalisierenden Ausdrücken zugeordnet. Dafür werden sowohl Richtungs- („vor“, „dahinter“) als auch Distanzangaben (z. B. „bei“, „nahe“) verwendet. Sind die Richtungsausdrücke nur unter Bezugnahme auf eine Perspektive interpretierbar, so sind Distanzrelationen blickpunktunabhängig.

Die Selektion von Richtungs- und Distanzausdrücken im Sprachproduktionsprozess und deren Kategorisierung wurden von Vorweg und Rickheit empirisch untersucht und die Ergebnisse in [Habel und von Sutterheim](#) veröffentlicht.

Mit sprachlichen Lokalisationsäußerungen wie „vor“, „hinter“, „links“ und „rechts“ beschäftigt sich der Beitrag von Klaus Kesser in [Rickheit \(1999\)](#).

### **Räumliches Wissen**

Das Wissen über die räumliche Anordnung und über Navigationsmöglichkeiten wird durch die Bewegung im Raum aufgebaut. Diese mentalen Repräsentationen werden räumliches Wissen und seit Tolmann (1948) auch „kognitive Karten“ genannt.

Es wird unterschieden zwischen *strip-maps* und den *comprehensive-maps*, sogenanntem *Routenwissen* und *Überblickswissen*. Das Routenwissen besteht aus Landmarken und wird durch die Navigation in der Umgebung aufgebaut; Landmarken sind auffällige Objekte in der Umgebung, die als Wegweiser fungieren. Routenwissen ist sequentiell, d. h. zeitlich organisiert, Überblickswissen dagegen räumlich.

In der Folge gelangte man zu der Auffassung, dass Routenwissen bereits am Anfang eine gewisse Eigenschaft von Überblickswissen enthält, d. h. jede Wegmarke weist auch eine räumliche Beziehung auf. Dieses Wissen wird durch zusätzliche Kenntnis der Umgebung weiterentwickelt und ausgebaut.

### **Fazit der Forschungsergebnisse**

Um die Pose eines Objekts sprachlich beschreiben zu können, wird räumliches Wissen benötigt. Räumliche Wahrnehmung basiert auf der Gewinnung von räumlichen Informationen. Räumliche Informationen wiederum werden durch Bewegungen und Wahrnehmungen gewonnen. Genauer gesagt werden interne räumliche Informationen auf verschiedene Weisen

aufgebaut, einerseits durch haptische Sinneswahrnehmung, andererseits durch Kommunikation.

Die Wissenschaftler haben herausgefunden, dass Menschen, bevor sie über Objekte und ihre räumlichen Eigenschaften bzw. ihre räumliche Beziehung zu anderen Objekten sprechen können, diese erst visuell, auditiv, taktil haben erfahren müssen. Hierdurch wird eine räumliche Repräsentation aufgebaut, die dann als Grundlage weiterer kognitiver Prozesse fungiert.

In einem ersten Schritt der Visualisierung muss der Sprecher aus einer mentalen Repräsentation ein Objekt auswählen, das er lokalisieren möchte (Locatum) und ein weiteres Objekt als Bezugspunkt (siehe: Räumliche Bezugssysteme 3.5.2) bestimmen. Für die Konzeptualisierung der räumlichen Relation muss eine Perspektivenwahl getroffen werden. (siehe: 3.5.2). Dieser Zusammenhang zwischen der Blickpunktabhängigkeit und dem Bezugssystem wird in der Sprache reflektiert. Für die Orientierung im Raum ist weiterhin räumliches Wissen (siehe: 3.5.2) erforderlich, welches weitere Anforderungen mit sich bringt.

Aus den wissenschaftlichen Ergebnissen kann gefolgert werden, dass für die sprachliche Beschreibung im Living Place Hamburg Landmarken definiert werden müssen. Durch die Verbindung der Landmarken miteinander kann Routenwissen aufgebaut werden.

Die Forscher haben unter anderem die Erkenntnis geliefert, dass räumliche Wahrnehmung auf der Gewinnung von räumlichen Informationen basiert [Rickheit \(1999\)](#) [Habel und von Sutterheim](#).

Die Forschungsarbeit an der HAW-Hamburg, bei der Landmarken verwendet wurden, um eine Route zu ermitteln, ist die Masterarbeit „Ein einfach benutzbares mobiles Navigationssystem für Fußgänger“ von Alewtina Schuman [Schuman \(2008\)](#). Sie entwickelte ein Navigationssystem für Fußgänger auf dem Flughafen, bei dem zur Orientierung in den Räumen Landmarken analysiert und extrahiert wurden. Mit Hilfe dieser Landmarken konnten sich die Menschen besser orientieren und den Weg besser finden.

Die Erkenntnisse aus der Forschung an der HAW-Hamburg und die dort gewonnenen Erfahrungen und Erkenntnisse über die praktische Umsetzung der Verfahren zur Entwicklung von räumlichen Informationen sind Hilfestellungen, die im Rahmen dieser Masterarbeit eine Lösung für die sprachliche Beschreibung der Lage des Objektes ermöglichen.

### 3.6. Fazit

In dem Abschnitt Anwendungsszenarien wurde gezeigt, wie der Bewohner Sal durch die durch Services im Living Place Hamburg im Alltag unterstützt wird. Die Abbildung 3.4 veran-

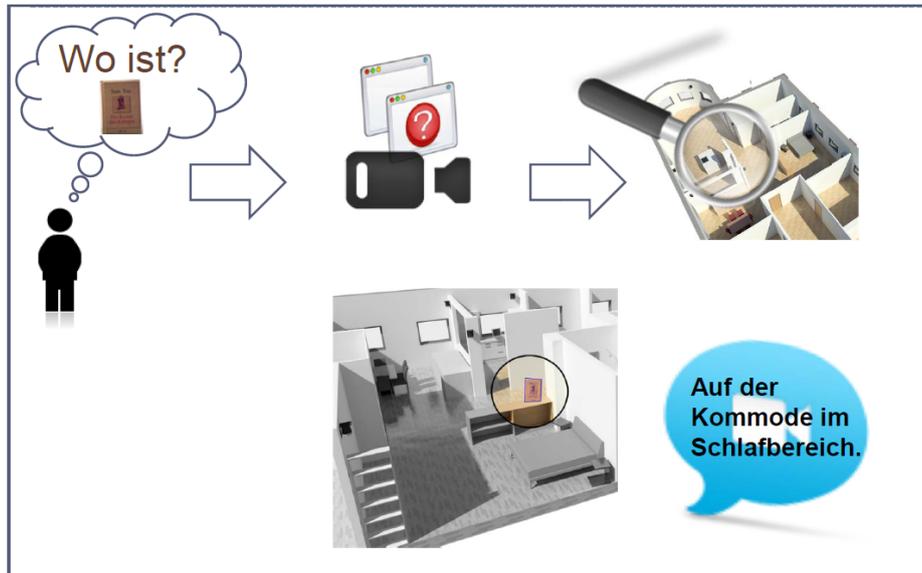


Abbildung 3.4.: Kameragestützte Suche des Buches

schaulicht noch einmal das Szenario, in dem der Bewohner des Living Place Hamburg das kamerabasierte Objektsuchsystem nutzt, um eine Serviceanfrage – hier im Beispiel „Wo ist mein Buch?“ – an die Wohnung zu stellen (vgl. Abschnitt 3.1.2).

Über ein Benutzer-Interface selektiert der Bewohner das zu suchende Referenzmodell und aktiviert damit den Suchprozess des kamerabasierten Objektsuch-Systems. In dem oben dargestellten Szenario wurde das gesuchte Referenzmodell „Buch“ von der Kamera im Schlafbereich auf der Kommode wiedergefunden. Sobald das gesuchte Objekt in dem Video-Stream lokalisiert ist, wird die Lage des Objektes auf einem der Monitore in der Wohnung dargestellt. Weiterhin wird dem Bewohner die Position des gefundenen Objektes und seine relative Lage zu den anderen Gegenständen in der Wohnung verbal in einem dem menschlichen Sprachgebrauch entsprechenden Satz formuliert und über den Lautsprecher angesagt. Es kann z. B. ein solcher Satz wie folgt lauten: „Das Buch ist im Schlafbereich auf der Kommode.“

Die Szenarien verdeutlichen das Zusammenspiel der Services in dem Gesamtkonzept des Living Place Hamburg. Ihre Analyse im Abschnitt 3.1 zeigte, welche der Services in den Szenarien durch diese Masterarbeit bzw. das kamerabasierte Objektsuch-System im Living Place Hamburg integriert werden sollten.

In dem Abschnitt technische Anforderungen 3.4.1 wurden die Service-Anforderungen vorgestellt, die sich aus den Szenarien für das kamerabasierte Objektsuch-System ergeben haben.

Daraus wurden für das kamerabasierte Objektsuch-System folgende Software-Agenten abgeleitet, die benötigt werden und für die ein Konzept und Prototyp entwickelt werden soll:

- Ein Software-Agent zur automatischen Erstellung von Referenzmodellen
- Ein Software-Agent für die automatische Wiedererkennung der Referenzmodelle in Bildern
- Ein Software-Agent für das Lokalisieren der Objekte im Raum
- Ein Software-Agent für die sprachliche Beschreibung der Lage des Objektes im Raum

Die dazu beschriebenen Anwendungsfälle wurden in dem Abschnitt 3.2 vorgestellt.

In dem Abschnitt 3.4.1 werden die funktionalen Anforderungen aufgeführt, die erfüllt sein müssen, um eine Anwendung zu realisieren, die den Bewohnern eine Unterstützung im Alltag bietet.

In dem Abschnitt „Related Work“ 3.5 wurden einige aktuelle Projekte und Forschungsprojekte aufgeführt. Die aktuellen Projekten lieferten Erfahrungsberichte und Erfahrungswerte aus dem Bereich der Objekterkennung und -verfolgung in Videobildern.

Die Forscher des Sonderforschungsprojekts „SFB/TR 8 Spatial Cognition – Reasoning, Action, Interaction“<sup>6</sup> lieferten einige Lösungsansätze für die sprachliche Beschreibung (vgl. Abschnitt 3.5.2); unter anderem die Erkenntnis, dass räumliche Wahrnehmung auf der Gewinnung von räumlichen Informationen basiert. Um die Pose eines Objektes sprachlich beschreiben zu können, wird räumliches Wissen benötigt. Dieses Wissen über die räumliche Anordnung und über Navigationsmöglichkeiten wird also durch die Bewegung im Raum aufgebaut. Diese mentalen Repräsentationen werden räumliches Wissen und seit Tolmann (1948) „kognitive Karten“ genannt

Mit diesen Erkenntnissen wird ein System entwickelt, welche den Ansprüchen eines „Smart Homes“ genügen soll.

Im folgenden Kapitel werden dann die Konzepte aufgeführt, die für die Anwendungsfälle und die Anforderungen im Rahmen der Masterarbeit entwickelt wurden.

---

<sup>6</sup><http://www.sfbtr8.spatial-cognition.de/>

## 4. Konzeption

In diesem Kapitel soll nun ein Konzept bzw. eine Systemarchitektur für ein System entworfen werden, das mit optischen Sensoren dem Bewohner einen Service zur Verfügung stellt, um beliebige Objekte im Living Place Hamburg zu suchen.

Die Grundlage dieser Zielsetzung bilden die sechs montierten PTZ-Domekameras<sup>1</sup>, die die optischen Sensoren im Living Place Hamburg darstellen. Deren Eigenschaften und Funktionalitäten sind im Anhang A beschrieben. Als Basis der Konzeption dienen die folgenden in den technischen Anforderungen 3.4.1 spezifizierten Software-Agenten:

- ein Software-Agent zur automatischen Erstellung von Referenzmodellen. Das Konzept wird durch die *Training GUI*- und *Location*-Komponente modelliert und der Zusammenhang ist in dem Abschnitt ?? beschrieben.
- ein Software-Agent für die automatische Suche der Referenzmodellen in Bildern. Das Konzept wird von der *Search GUI*- und *Search-Komponente* modelliert, der Zusammenhang ist in Abschnitt 4.2 beschrieben.
- ein Software-Agent für das Lokalisieren des Objekten im Raum. und eine weitere Software-Agent für die sprachliche Beschreibung der Lage des Objektes im Raum. Das Konzept wird von der *Speech*- und *Location*-Komponente modelliert und der Zusammenhang ist im Abschnitt 4.3 beschrieben.

Jeder Software-Agent wird wiederum als ein Modul modelliert, der aus mehreren Komponenten besteht, um so die Komplexität der einzelnen Software-Agenten zu reduzieren.

---

<sup>1</sup><http://www.dallmeier-electronic.com/de/cctv-ip-videoueberwachung/kameras/modelle/ptz-domekameras/ddz4010-yyhshd-domerar.html>

## 4.1. Systemarchitektur

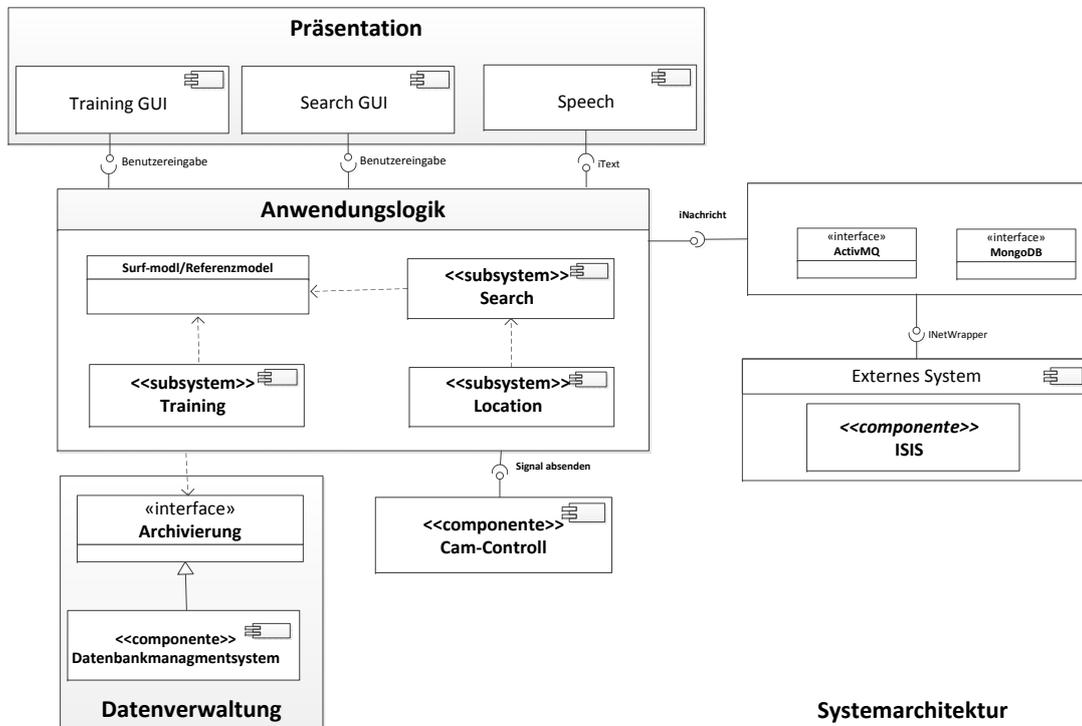


Abbildung 4.1.: Systemarchitektur in Gesamtüberblick

Das Komponenten-Diagramm in der Abbildung 4.1 stellt die Systemarchitektur dar. Sie ist unterteilt in eine Präsentations-, Anwendungs- und Datenverwaltungs-Schicht. Die einzelnen Schichten sind voneinander entkoppelt und agieren nur über jeweils eine Schnittstelle miteinander. Durch diese Drei-Schichten-Architektur wird vor allem die Benutzerschicht (der sogenannte User Layer) von der Anwendung (dem Application Layer) getrennt.

Jede dieser Komponente der Systemarchitektur modelliert eine bestimmte Funktionalität, die einen Teil der in Kapitel 3 beschriebenen Anforderungen erfüllt. Um die Wartbarkeit, Flexibilität und Weiterentwicklungsmöglichkeiten des kamerabasierten Objektsuch-Systems zu gewährleisten, sind alle (Teilsystem-)Module voneinander entkoppelt modelliert, so kann jedes Teilsystem unabhängig von anderen genutzt und erweitert werden.

### 4.1.1. Präsentationsschicht

Die Training GUI-, Search GUI- und Speech-Komponente modellieren die Präsentationsschicht. Diese bilden die Benutzeranwendungen (Schnittstelle) zwischen dem Anwender und der Anwendungslogik. Die Präsentationsschicht ist verantwortlich für die Verwaltung der Benutzereingaben bzw. Servicanfragen.

Die Anwendungslogik stellt die Ergebnisse der Anfragen den entsprechenden Komponenten der Präsentationsschicht zur Verfügung. Diese ist dann dafür verantwortlich, dass die Ergebnisse situationsadäquat dem Benutzer zur Verfügung gestellt werden.

Die Training-GUI-Komponente konzipiert die Benutzeranwendung für den Anwendungsfall „Referenzmodell erstellen“ für das Szenario 3.2.2.

Die Search-GUI-Komponente konzipiert die Benutzeranwendung für den Anwendungsfall „Service Anfrage an das System stellen“ 3.2.1 für das Szenario 3.2.1. Die Search GUI integriert aber auch die visuelle Anzeige der Lage des Objektes in dem Bildern der Kameras, welche durch den Anwendungsfall „Objekt anzeigen“ 3.2.1 beschrieben wird.

Die Speech-Komponente repräsentiert die Sprachausgabe, aus dem Anwendungsfall „Sprachliche Beschreibung der Pose“ 3.2.1. Es enthält von der Anwendungslogik einen Text, der die Beschreibung der Lage des Objektes im Rum enthält. Diese Komponenten ist dafür zuständig, den formulierten Text sprachlich wiederzugeben. Somit konzipiert es die Implementierung einer „Text to Speech“-Anwendung.

### 4.1.2. Kamera-Komponente

Eine weitere wichtige Komponente im ganzem System ist die Kamera-Komponente, die „Cam-controll“ in . Abb. 4.1. Sie stellt eine Schnittstelle für den Zugriff auf die einzelnen PTZ-Domekamas im Living Place Hamburg dar. Über dieser Schnittstelle greift die Anwendungsschicht auf die einzelne Kamera zu und stellt die Video-Bilder dem Gesamtsystem zur Verfügung.

Die Benutzeranwendungen aus der Präsentationsschicht können nicht direkt auf die Kamera zugreifen, vielmehr ist die Anwendungsschicht dafür verantwortlich, dass der zuständigen Benutzeroberfläche das richtige Kamerabild zur Verfügung gestellt wird. Die Anwendungsschicht integriert die Logik, wann und auf welche Kamera zugegriffen werden soll.

Die Kamera-Komponente selber implementiert die entsprechende Schnittstelle für den Zugriff auf die einzelne Kamera. Sie bekommt von der Anwendungsschicht die Nachricht, welche Kamera wie gesteuert werden soll, und sendet die Nachricht an die entsprechende Kamera weiter und stellt umgekehrt dann die Bilder der Anwendung zur Verfügung.

### 4.1.3. Datenverwaltung

Die Datenverwaltung in dem Systemarchitektur (vgl. Abb.4.1) integriert die Schnittstelle „Archivierung“. Diese abstrahiert den Zugriff auf die Datenbank und stellt die benötigte Technik zum Datenaustausch zwischen der Anwendungslogik und der Datenbank zur Verfügung; sie implementiert die Funktionen Lesen, Schreiben und Speichern von Daten in der Datenbank.

**Datenbank Entwurf** Für die Anforderung das Referenzmodell im System zu hinterlegen, aus dem Szenario „Objekt im System hinterlegen“ 3.2.2, wird durch das modellieren eines Datenbankmodell realisiert.

Der Zugriff auf das Referenzmodell von jedem Modul wird über die entsprechende Schnittstelle, in dem Systemarchitektur als „Archivierung“ dargestellt, gewährleistet.

Die Anforderung aus dem Anwendungsfällen „Referenzmodell erstellen“ 3.2.2 und 3.2.2 an das Konzept des Datenbankmodell sind:

Das Referenzmodell enthält einen Namen. Es wird einen Typen zugeordnet. Weiterhin soll die beschreibenden Merkmale, die sogenannte Metainformationen jedem Objekt zugeordnet werden und neben dem Namen als Suchkriterium dienen.

Weitherin ist ein Referenzmodell definiert, beschrieben im Abschnitt „Merkmalbasiertes Objekterfassungs- und modellierungsverfahren“ 2.2 in dem Kapitel 2 und in dem Anwendungsfall „Referenzmodell erstellen“; es repräsentiert ein Objekt durch durch die extrahierten Merkmale, die sogenannten Deskriptoren (engl. descriptors), einen Bildpunkt (keypoint) und eine Orientierung.

Im Folgenden wird anhand eines Datenbank-Entwurfs (vgl. Abb. 4.2) erläutert, welche minimale Tabellen eine Datenbank besitzen muss, um die gestellten Anforderungen erfüllen zu können.

Die Datenbank enthält mindestens die folgenden Tabellen:

**Das Referenzmodell**, bestehend aus mindestens einem Bild, sowie einem oder mehreren

**Key points**, die ihrerseits mindestens einen oder mehrere

**Deskriptoren** enthalten. Die

**Tag-Tabelle** speichert alle Benutzereingaben, durch die er Meta-Informationen über die Benutzeroberfläche ins System einträgt.

**Bilder** speichert die Bilder zu einem Referenzmodell.

**Kategorien** speichert den Typen, die einem Benutzer in einer Liste zur Auswahl geeigneter Gegenstände angezeigt werden sollen.

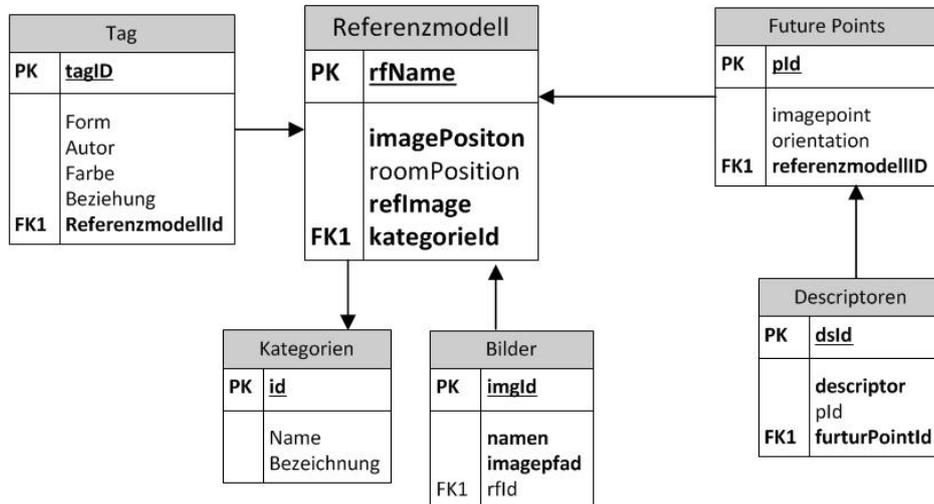


Abbildung 4.2.: Datenbank-Entwurf

#### 4.1.4. Anbindung an Living Place Hamburg

Die zentrale Kommunikationsschnittstelle zum Austausch von Nachrichten und Daten zwischen den einzelnen Applikationen im Living Place Hamburg bildet der ActiveMQ Message Broker. Damit wurde eine Blackboard-Architektur als zentrale Kommunikationsplattform im Living Place Hamburg realisiert. Die Umsetzung berücksichtigt die Anforderungen für eine intelligente Umgebung (vgl. ?).

Im Rahmen des Living Place Hamburg wurde in dem Projekten bereits ActiveMQ eingesetzt, ein Message-Broker, der einen vollständigen Java Messaging Service implementiert. Auch wenn er in Java implementiert wurde, ist der Zugriff von verschiedenen Programmiersprachen aus möglich. Zudem unterstützt er verschiedene Services und Transportprotokolle.

Als zentrale Datenbank wurde MongoDB in die Blackboard-Architektur integriert und an den ActiveMQ angebunden. Dies bringt den Vorteil mit sich, dass Sensor-Daten und Informationen, wie z. B. Zustandsinformationen eines Gerätes, von jeder Anwendung und zu jeder Zeit erfragt werden können.

In der Systemarchitektur (vgl. Abb.4.1) stellt die *LP-Kommunikation*-Komponente die Schnittstelle zu ActivMQ dar. Darüber werden die Anfragen des Kamerabasierte Objektsuche System an einen bestimmten Service des kamerabasierten Objektsuch-Systems abgewickelt.

(Außerdem ermöglicht es diese Kommunikationsplattform, eigene Dienste für andere Systeme im Living Place Hamburg zur Verfügung zu stellen.)

#### 4.1.5. Anwendungsschicht

Die Anwendungsschicht, in der Abb. 4.1 als Anwendungslogik bezeichnet, modelliert die Logik des kamerabasierten Objektsuch-Systems und stellt somit dessen Hauptkomponente dar. Diese bindet weitere Komponenten ein oder bietet diesen eigene Services an.

Die Binnenansicht der Anwendungsschicht findet sich in der Abbildung 4.3. Sie besteht aus der SURF-Klasse und den Komponenten *Training*, *Search* und *Location*.

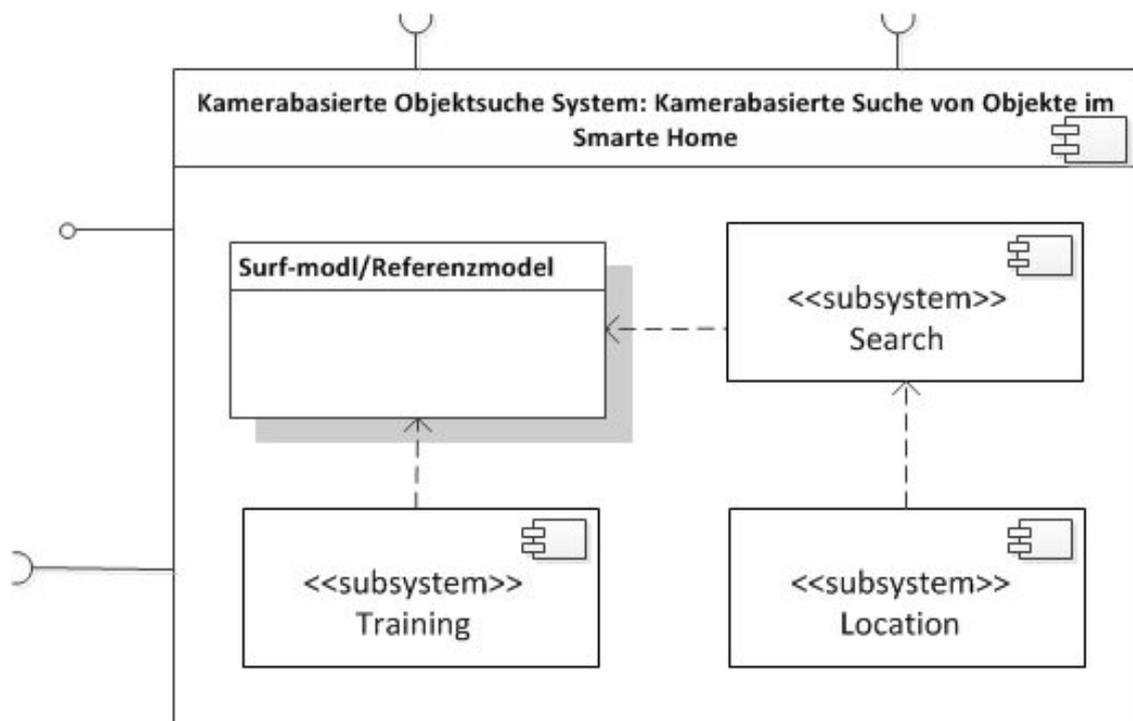


Abbildung 4.3.: Anwendungs-komponente Binnenansicht

Die Komponenten haben folgenden logischen Aufbau und Abhängigkeiten untereinander:

- Die Klasse SURF-Klasse integriert die Implementierung der Computer-Vision-Funktionalitäten zum Erstellen eines Referenzmodells.
- Die Training- und Search-Komponenten stellen jeweils ein eigenes Subsystem in der Anwendungsschicht dar.
- Die Training- und Search-Komponenten verwenden die SURF-Klasse.

Im Folgenden wird das Zusammenspiel der einzelnen Komponenten detaillierter erläutert. Es werden die Konzepte für die in der Analyse aufgeführten Teilbereiche: „Serviceanfrage verstehen“ (??), „Orten von Objekten in Bildern“ (4.2) und „Orten von Objekte im Raum“ (4.3) aufgeführt, um schließlich die Lage des Objektes im Raum sprachlich wiedergeben zu können.

Dieser Abschnitt beschreibt das Konzept für das Szenario: „Objekt im System hinterlegen“, 3.2.2.

In der Analyse wurden die funktionalen Anforderungen bereits in Abschnitt 3.4.3 aufgestellt; im Folgenden werden die zuständigen Komponenten und deren Verantwortlichkeiten in ihrem Zusammenspiel dargestellt; in der Abbildung ?? sind sie farblich markiert.

Die Aufgabe dieser Komponente ist es, ein Objekt im System zu hinterlegen; hierfür dient die Training-GUI in der Präsentationsschicht als die Schnittstelle zwischen dem Benutzer und der Anwendung. Das Konzept für die Realisierung dieses Szenarios ist in der Abbildung ?? dargestellt.

#### 4.1.6. Training-Komponente

Die Training-Komponente implementiert die in dem Anwendungsfall „Objekt im System hinterlegen“ 3.3 (mit den konkreten Anwendungsfällen „Referenzbilder hinterlegen“ und „Referenzmodell erstellen“) beschriebenen Funktionen.

Hierfür bindet sie die Surf-Klasse/Surf-Modellein, um, wie in dem Abschnitt 2 dargestellt, Referenzmodelle aus den Referenzbildern zu erstellen und im System zu hinterlegen.

**Das Konzept** Die Training-GUI erfasst die Eingaben der Benutzer. Diese werden von der Präsentationsschicht an die Training-Komponente über die Anwendungslogik weiter delegiert. Näherhin gibt der User die Referenzbilder über die Training-GUI Benutzeroberfläche ein. Bei Betätigen des Buttons „Referenzbilder speichern“ werden sie dann an die Training-Komponente übergeben und von ihr in der Datenbank abgelegt. Bei Betätigen des Buttons „Referenzmodell erstellen“ auf der Training-GUI (der Anwendungsfall „Referenzmodell erstellen“) extrahiert die Training-Komponente aus den Referenzbildern die Merkmale und legt sie in der Datenbank unter dem angegebenen Namen ab. Dafür macht sie Gebrauch von den Computer-Vision-Funktionalitäten der Surf-Klasse/Surf-Modell.

Das Ergebnis der Training-Komponente ist somit das aus den Referenzbildern und Eingabedaten des Benutzers erstellte Referenzmodell.

Um es im Gesamtsystem zu hinterlegen, stellt die Training-Komponente über die Verwaltungskomponente eine Verbindung mit der Datenbank her und speichert es in dieser ab,

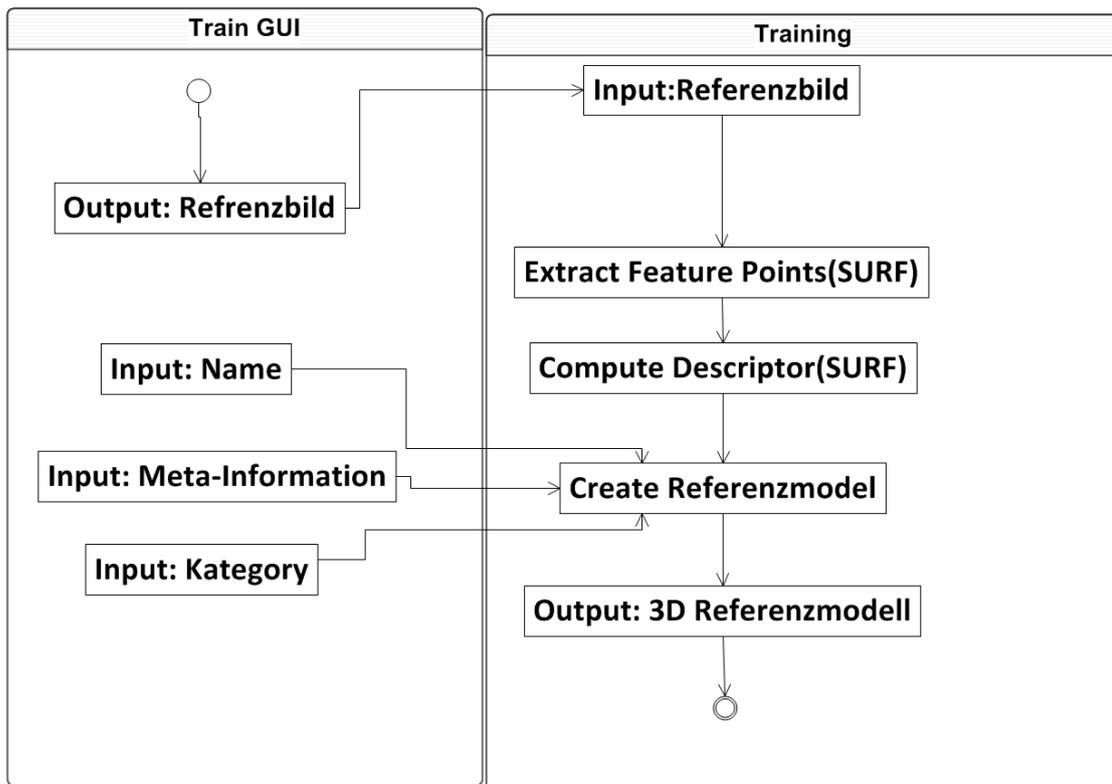


Abbildung 4.4.: Dynamischer Ablauf: Serviceanfrage verstehen

dafür bindet sie das Interface „Archivierung“ ein, um deren Funktionalitäten für den Zugriff auf die Datenbank zu benutzen. Das Konzept eines ersten Datenbankaufbaus wurde bereits in dem Abschnitt [4.1.3](#) dargestellt.

#### 4.1.7. Benutzeroberfläche: Training-GUI

Im Folgenden wird das Konzept aus Benutzersicht beschrieben. Dabei wurde darauf geachtet, dass die Anwendung einfach zu bedienen ist, die Bedienoberfläche ist deshalb auf die wesentlichen Funktionen beschränkt. Die beschriebenen Funktionalitäten stellen nur das Minimum dar, das noch je nach Bedarf erweitert werden kann. Erst in der Praxis wird herausgefunden werden, was an Funktionen zu implementieren ist; dabei soll der Usability-Aspekt im Vordergrund stehen, um den Benutzer mit der Bedienung nicht zu überfordern, dennoch soll er durch Zusatz-Funktionalitäten einen höheren Komfort angeboten bekommen können. Die Abbildung [4.5](#) stellt einen Mock-Up des Training-GUI dar. Die Training-GUI konzipiert die Benutzeroberflächen für die Anwendungsfälle „Referenzmodell hinterlegen“ [3.2.2](#) und „Referenzmodell erstellen“ [3.2.2](#). Darüber hinaus ermöglicht sie es, das Referenzmodell zu editieren.

In der Detailansicht in Abbildung [4.5](#) hat der Benutzer die Optionen:

- den Namen für das zu erstellte Referenzmodell zu hinterlegen.
- Bilder durch die integrierte Kamera zu erfassen und durch Betätigen des Buttons „Bild speichern“ im System zu speichern.
- In dem unteren Bereich „Meta-Daten“ können zu dem Objekt weitere Metainformationen eingetragen werden.
- Das Button „Referenzmodell erstellen“ triggert den gleichnamigen Anwendungsfall.

Die in der Abbildung [4.5](#) dargestellten Buttons und das Namensfeld bilden die Hauptbestandteile, mit denen der Benutzer die verschiedenen Funktionalitäten erreichen kann.

**Textfeld: Objekt-Namen** Um die Zuordnung des Objektes zu dem Referenzmodell eindeutig zu gestalten, wurde das System so konzipiert, dass das Referenzmodell anhand des Namens im System referenziert und identifiziert wird. Damit keine zwei Objekte mit dem gleichen Namen im System hinterlegt werden, findet bei der Angabe des Namens ein Abgleich mit den bereits gespeicherten Namen statt. Wenn der Name bereits im System vorhanden ist, wird dem Benutzer eine entsprechende Meldung angezeigt. Der Name ist ein Pflichtfeld, denn er ist zur Auswahl des Referenzmodells notwendig.

Training GUI

Objekt Name \*:

*Bitte halten sie das Objekt vor die Kamera, bis das Bild hier zu sehen ist*

Meta Informationen

Bitte eine Kategorie Wählen ▼ Edit New

Buch  
Kleidung  
Geschirr

Meta Daten:

Abbildung 4.5.: Konzept Training-GUI: Referenzmodell erstellen

Möchte der Benutzer ein bereits hinterlegtes Referenzmodell editieren, löschen oder ergänzen, kann der Benutzer den Namen in das Feld eingeben und mit Enter bestätigen. Bei der Angabe des Namens findet ein Abgleich mit den bereits gespeicherten Namen statt und das unter dem Namen hinterlegte Referenzbild wird dem Benutzer angezeigt. Der Benutzer kann jetzt die Funktionalitäten des Training-GUI nutzen, um das Referenzmodell zu ändern.

Mit der Auswahl eines Namens aus der vorgeschlagenen Liste signalisiert das User-Interface Training-GUI dem System, dass ein bereits unter diesem Namen erstelltes Referenzmodell geändert werden soll. Das System stellt dann die gespeicherten Informationen dem User-Interface Training-GUI zur Verfügung und diese werden dem Benutzer angezeigt. Der Benutzer hat nun die Möglichkeit die Angaben des Datensatzes zu korrigieren oder zu ergänzen.

**Action-Button: Bild speichern** Mit dem Action-Button „Bild speichern“ wird der Anwendungsfall „Referenzbilder hinterlegen“ [3.2.2](#) ausgelöst.

Sobald das Objekt vor die Kamera gehalten wird, errechnet das System, ob das Objekt der Anforderung „Referenzbilder hinterlegen“ im Abschnitt [3.4.3](#) entspricht. Falls dies der Fall ist und die Erfassung von Bildeigenschaften angefordert wurde, wird das Bild auf dem Interface im Bildbereich angezeigt und der Button „Bild speichern“ aktiviert. Solange die Aufnahme des Bildes nicht den entsprechenden Anforderungen genügt, bleibt der Button „Bild speichern“ hingegen deaktiviert.

Der Benutzer kann selbst entscheiden, wie viele Bilder er für die Erstellung des Referenzmodells im System hinterlegen möchte. Daher löst der Button „Bild speichern“ nicht gleich den Prozess „Referenzmodell hinterlegen“ aus, sondern dieser wird erst durch den Button „Referenzmodell erstellen/hinterlegen“ ausgelöst.

Dadurch die oben genannte Konzept wird die in der Anforderung „Referenzbilder hinterlegen“, „Objekterfassung und Bildaufnahme“, im Abschnitt [3.4.3](#) beschriebene Eigenschaft realisiert.

Um das Durchsuchen des System nach den hinterlegten Referenzmodellen smarter und performanter zu gestalten, wurde der Prozess „Referenzmodell erstellen“ so konzipiert, dass jedes Referenzmodell einer vordefinierten Kategorie zugeordnet ist und dass zu jedem Referenzmodell optional zusätzliche Metainformationen hinterlegt werden können. Hierdurch werden die Anforderungen an „Serviceanfrage verstehen“ in Abschnitt [3.4.3](#) (genauer die „Anfrage eingeben“ und „Anfrage verstehen“ aus [3.4.3](#)) implementiert.

**Kategorien** Die Angaben einer Kategorie bieten dem Benutzer einen zusätzlichen Komfort bei der Auswahl der Referenzmodelle aus der Datenbank. Kategorien erleichtern zusammen mit den Metainformationen die Suche nach einem Referenzmodell. Der Benutzer muss zu jedem Referenzmodell eine Typen vergeben (vgl. Anwendungsfall 3.2.2).

Zur Auswahl einer Kategorien stehen bereits einige vordefinierte Kategorien in einer Auswahlbox zur Verfügung. Die Training-GUI-Eingabemaske bietet neben der Auswahlbox mit den Kategorien einen „Edit“-Button, um sie zu ändern, und den „New“-Button, um eine zusätzliche Kategorie hinzuzufügen.

**Metainformationen** Diese Metainformationen dienen ausschließlich zur Unterstützung der Auswahl des Objektes im Such-Prozess, also dazu, dass das System die Service-Anfrage des Benutzers besser versteht.

Um die Anforderung: „Das System soll smart sein“ zu realisieren, d. h. in diesem Fall konkret, um dem Benutzer die Auswahl des Referenzmodells auch über die Eigenschaften des Objektes ermöglichen, modelliert/konzipiert dieser Bereich Metainformationen, die als beschreibende Informationen zu dem Referenzmodell hinterlegt werden können. Im Ergebnis sollte z. B. folgende Service-Anfrage verstanden werden können: „Wo ist mein Buch mit dem Titel ‚Du schaffst es!!‘ oder ‚UML‘, vom Autor [ . . . ], geschenkt von meinem Freund Felix?“

Im Feld Name genügt es, den Titel des Buches, etwa „Du schaffst es“ oder „UML“, einzutragen. In der Abbildung 4.5 wurde das Textfeld „Meta-Daten“ für die Eintragung der zusätzlichen Informationen zur Verfügung gestellt; wobei dieses Text-Feld nur einen eher symbolischen Charakter hat. Die Darstellungsform für die Eingabe der Metainformationen ist prinzipiell abhängig von Konzeption der Datenbank und den darin enthaltenen Datenstrukturen für die Metainformationen.

Der Aufbau der Metainformationen und die Bestimmung von Typen wird in dem folgenden Abschnitt „Baumstruktur“ 4.1.7 erläutert.

### **Aufbau einer Baumstruktur für mobile Objekte**

Die erste Frage, die hier behandelt wird ist: Wie können Objekten voneinander unterschieden werden? Dazu bedient man sich der Form, Größe und weiterer Eigenschaften.

Jedes Objekt besitzt einen Namen und Attribute, welche die Eigenschaften eines Objekte widerspiegeln. Um diese Attribute und Eigenschaften eines Objekte in Beziehung zu setzen und im System zu hinterlegen, bietet sich der Aufbau einer Baumstruktur an.

Diese Baumstruktur wird aus den beweglichen Objekten im Living Place Hamburg aufgebaut. Den Wurzelknoten des Baumes bilden die „beweglichen Objekte“. Er hat keine konkrete Ausprägung, sondern fasst nur alle Daten zusammen.

Die obersten Kinds-Knoten stellen die Kategorien dar, so werden die Objekte in verschiedenen Kategorien aufgeteilt. Um die Objekte mit weiteren Attributen zu versehen, kann jede Kategorie ihre eigene Spezialisierungen enthalten. Dieser Baum kann beliebig in der Breite und in der Tiefe wachsen. Die Breite des Baumes repräsentiert alle mögliche Kategorien von Objekten und die Tiefe des Baumes repräsentiert deren Spezialisierung.

Jeder Zweig des Baumes repräsentiert auch eine mögliche Serviceanfrage an das System. Je länger ein Zweig, desto mehr Attribute sind einem Objekt zugeordnet worden und desto präziser ist dessen Beschreibung; desto mehr Möglichkeiten bestehen auch, eine Objektauswahl anhand der Attribute durchzuführen.

Je breiter die Abzweigung, desto mehr Möglichkeiten stehen zur Auswahl.

Bei der Serviceanfrage „Wo ist mein XYZ-Buch“ werden hingegen nur die Blätter des Baumes durchsucht und das Objekt nur anhand des Namens ausgewählt.

**Action-Button: Referenzmodell erstellen** Mit dem Button „Referenzmodell erstellen“ wird der Prozess angestoßen, um ein Referenzmodell zu erstellen. Das dazu benutzte Verfahren ist in dem Kapitel 2 in Abschnitt 2.2 dargestellt.

Um den Prozess „Referenzmodell erstellen“ anstoßen zu können, müssen folgende Daten vorhanden sein:

- Mindestens ein Referenzbild wurde gespeichert hinterlegt.
- Es wurde ein Name eingetragen, der noch nicht im System existiert.
- Es wurde eine Kategorie ausgewählt.

## 4.2. Subsystem Search: Orten von Objekt in Bildern

Angelehnt an Szenario und Anwendungsfall „Objektsuche in Bildern“ 3.2.1 modelliert die Abbildung ?? das Konzept dazu. Die zuständigen Komponenten in der Systemarchitektur sind darin farblich markiert und im Folgenden wird deren Zusammenspiel beschrieben.

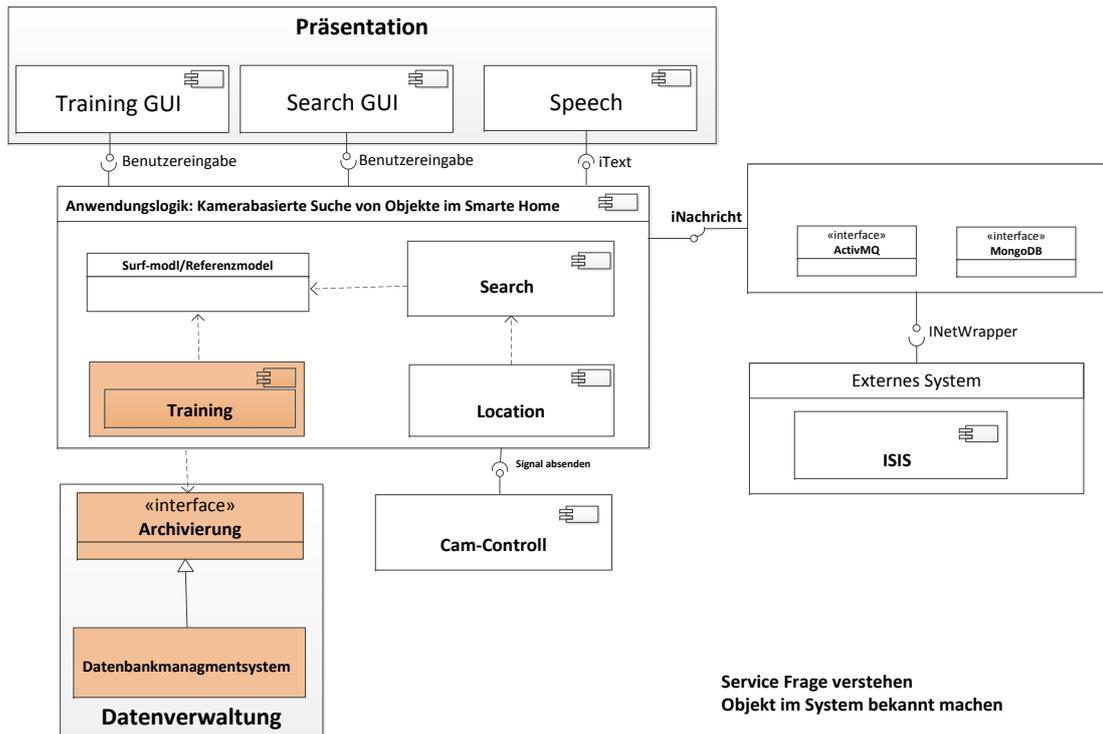


Abbildung 4.6.: Orten von Objekten in Bilder

Die Suche nach einem Objekt in Bildern bedeutet das Vergleichen eines Referenzmodells mit den Kamerabildern des Living Place Hamburg . Hierfür benötigt die Search-Komponente zum einen die Surf-Klasse/Surf-modell, zum anderem die Live-Bilder der Kameras aus dem Living Place Hamburg . In der Präsentationsschicht modelliert die „Search-GUI“-Komponente die Benutzerschnittstelle für die Suche nach Objekten im Raum. Dazu stellt der Benutzer am Anfang eine Anfrage an das System über die Search-GUI-Komponente.

### 4.2.1. Search-Komponente

Für die Suche des Objektes in Bilder, die durch den Anwendungsfall „Objekt-Suche“ in Abschnitt 3.4 beschrieben wird, benötigt die Search-Komponenten eine Kameraschnittstelle für den Zugriff auf die Kameras im Living Place Hamburg ; dafür bindet sie die Kamera-Komponente “cam-controll“ ein, an die alle Anfragen der Search-Komponente an die einzelnen Kameras weitergeleitet werden.

Die Kamera-Komponente (Cam-Control) ist lediglich für den Zugriff auf die entsprechende Kamera über das Netzwerk verantwortlich. Sie leitet ihrerseits die Steuersignale bzw. alle Anfragen von der Search-Komponente an die entsprechende Kamera weiter.

Die Search-Komponente ist weiterhin verantwortlich für das Modellieren einer Steuerungs-Logik der Kameras im Living Place Hamburg , die die Anforderungen aus dem Abschnitt 3.4.3 realisiert. Dafür integriert die Search-Komponente ein internes (Kamerasteuerung-)Modul, das zu entscheiden hat, welche Kamera welchen Bereich der Wohnung durchsuchen soll. Es hat Informationen darüber, welcher Bereich der Wohnung von welcher Kamera abgedeckt wird, und auch, wie weit das Sichtfeld einer Kamera reicht und weiß damit, von welcher Raumkoordinaten bis zu welcher Raumkoordinaten welche Kamera den Wohnbereich durchsuchen kann.

Auf Grundlage dieser Informationen konzipiert die Search-Komponente ein weiteres Modul, das Such-Planungs-Modul, das die Lösung der Anforderung „Planungs- und Koordinations-einheit“ im Abschnitt 3.4.3 integriert. Es entscheidet darüber,

- wann ein Objekt in der Wohnung zu finden ist,
- welche Kamera, welchen Bereich, mit welcher Einstellung durchsuchen soll und
- wann die Suche einen Erfolg oder einen Misserfolg liefert.

Neben diesen beiden Modulen nutzt die Search-Komponente noch die SURF-Klasse, um die Objekte in den Bilder der Kamera wiederzufinden. Die SURF-Klasse benötigt für die Suche zu einem auf die Kamerabilder zum anderem ein Referenzmodell.

In dem Abschnitt 2.2 wurde das SURF-Verfahren vorgestellt und der Ablauf eines Suchprozesses beschrieben, insbesondere, wie Objekte in den Bildern der Kamera wiedererkannt werden. Damit wird die Anforderung : „Orten von Objekten in Bildern“ aus dem Abschnitt 3.4.3 erfüllt.

Im folgenden wird erläutert, wie das Zusammenspiel der Komponenten untereinander aussieht. Der Ablauf des Konzept ist in der Abbildung 4.7 visuell dargestellt. **Das Konzept:** Der Benutzer wählt den Namen des zu suchenden Objekts über die Benutzeroberfläche Search-GUI aus. Die Präsentationsschicht gibt den Namen des Objektes an die Search-Komponente weiter. Die Search-Komponente stellt eine Verbindung über die Verwaltungsschicht zu der

Datenbank her und stellt eine Anfrage an die Datenbank. Diese liefert das Referenzmodell zu dem gewählten Namen zurück.

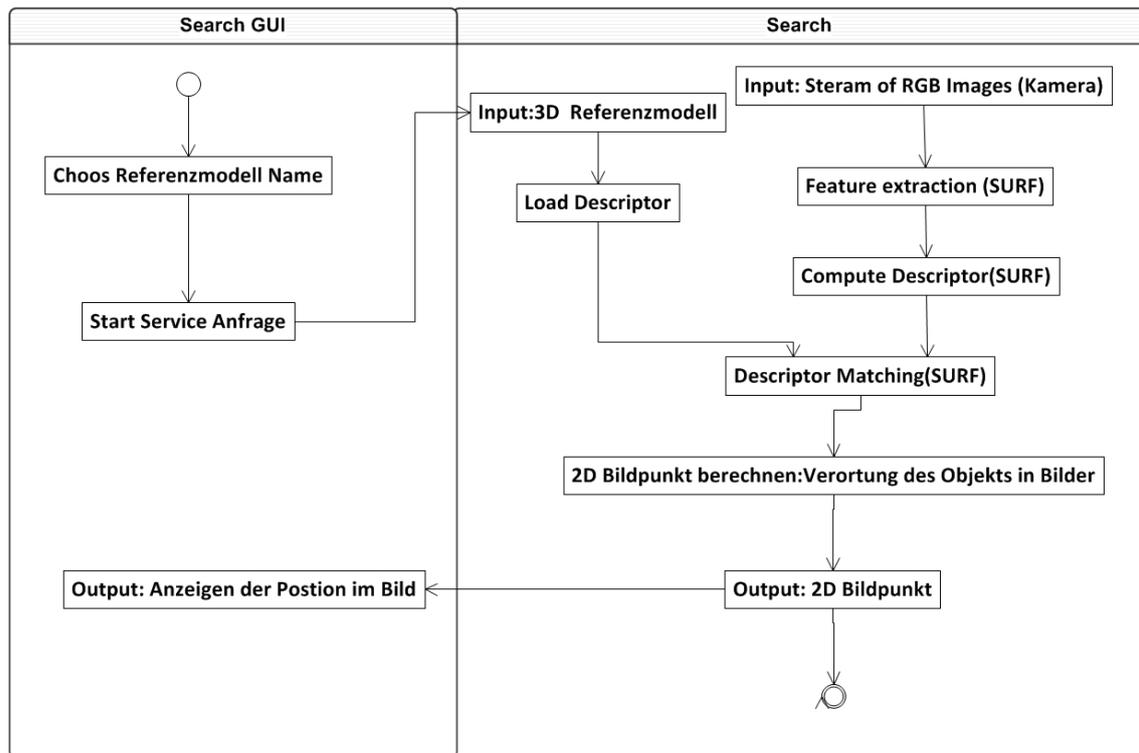


Abbildung 4.7.: Dynamischer Ablauf : Orten in Bildern

Dann wählt die Komponente die Kamera(s) für die Suche aus und steuert sie entsprechend. Der von der Kamera gelieferte Videostream wird auf die Merkmale des Suchobjekts hin überprüft. Sobald eine der Kameras das Objekt im Raum wiedergefunden hat, sendet die Search-Komponente ein Signal an die Kamera-Komponente, um die Suche wieder zu stoppen.

Sie ruft das Bild der Kamera, in dem das Objekt wiedergefunden wurde, auf und leitet dieses an die Search-GUI-Komponente weiter zur Anzeige auf der Benutzeroberfläche. Darüber hinaus sendet die Search-Komponente die errechneten 2D-Koordinaten des Objekts im Bild, den sogenannten Bildpunkt, an die Search-GUI weiter. Um den Bildpunkt im System bekannt zu machen, wird die Position in der Datenbank hinterlegt. Hierfür bindet die Search-GUI-Komponente die Datenbank ein und delegiert das Speichern der Daten an die Verwaltungsschicht.

Im folgenden Abschnitt wird die Search-GUI-Komponente aus der Sicht des Benutzers im Detail beschrieben.

### 4.2.2. Benutzeroberfläche: Search-GUI

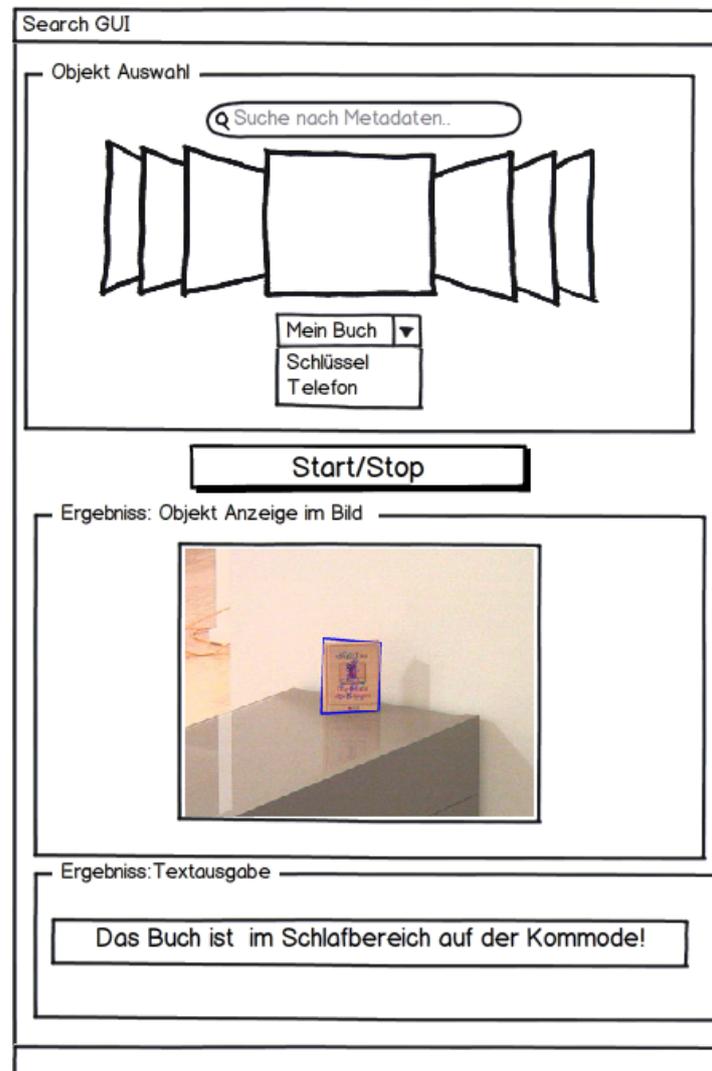


Abbildung 4.8.: Interface Search-GUI

Dieser Abschnitt erläutert die konzipierte Anwendung aus der Sicht der Benutzer. Die Search-GUI-Komponente modelliert die Benutzeranwendung für die Serviceanfragen an das System. Dies sind die Anwendungsfälle „Suchanfrage starten“ [3.3](#) und „Position anzeigen“ [3.2.1](#) mit der anschließenden visuellen Ausgabe der Lage des Objektes.

Die Benutzeroberfläche ist in drei Bereiche unterteilt: Objektauswahl, visuelle Anzeige des Suchergebnisses und die sprachliche Beschreibung der Lage des Objektes im Raum.

Die Anwendungsfälle „Serviceanfrage starten“ und „Serviceanfrage stoppen“ werden durch

das Betätigen des entsprechenden Buttons ausgelöst. Einmaliges Betätigen des Buttons „Start/Stop“ aktiviert den Suchprozess, durch das erneute Betätigen des Buttons wird der Suchprozess wieder gestoppt.

Der obere Bereich mit der Beschriftung „Objektauswahl“ bietet dem Benutzer die Möglichkeit, ein bereits hinterlegtes Referenzmodell im System auszuwählen. Dafür bietet das Interface in dem Bereich Objektauswahl einmal die Möglichkeit an, das hinterlegte Referenzmodell über seinen Namen auszuwählen, zum anderen gibt es für den Fall, dass der Benutzer den genauen Namen des Objektes nicht kennt, die Möglichkeit, die Namensliste nach bestimmten Eigenschaften zu filtern.

Die Identifikation des hinterlegten Referenzmodells findet über den Namen des Objektes statt. Die Benutzeroberfläche stellt die Namen der hinterlegten Referenzmodelle in einer Dropdown-Liste zur Verfügung.

Durch die Auswahl eines Names wird eine Anfrage an die Anwendungslogik gestartet. Als Ergebnis wird von der Anwendungslogik der unter diesem Namen hinterlegte Datensatz der Training-GUI zu Verfügung gestellt. Der Datensatz besteht aus dem Namen und einem Bild des Referenzmodells.

Diese Ergebnisse der Suchanfragen werden von dem Search-GUI dem Benutzer im Mittelbereich angezeigt.

Die in der Abbildung 4.8 dargestellte Image-Box modelliert die Anzeige aller gespeicherten Bilder der Referenzmodelle. Sie ist mit einer Auswahl-Aktion hinterlegt; sobald ein Bild ausgewählt wurde, wird der Button „Start/Stop“ zum Auslösen der Serviceanfrage aktiviert.

Die Auswahl über den Namen modelliert die Serviceanfrage: „Wo ist mein Buch?“ Die Auswahl über das Suchfeld modelliert die Serviceanfragen: „Wo ist das rote Buch, geschenkt von meiner Schwester im Januar ?“. „Wo sind die Bücher von dem Autor XYZ“.

Für den Fall, dass der Benutzer den genauen Namen des Objektes nicht kennt, bietet die Benutzeroberfläche also die Möglichkeit, das Objekt nach einem bestimmten Tag (Meta-Informationen) oder einer Eigenschaft wie etwa einer Kategorie zu selektieren; z. B. erfolgt eine Einschränkung der Namens-Liste durch Eintragung eines Begriffes in das Suchfeld.

**Auswahl über den Namen:** Wie in der Abbildung modelliert, werden dem Benutzer die Namen der gespeicherten Referenzmodelle in der Dropdown-Liste zur Verfügung gestellt. Durch die Auswahl eines Namens wird eine Anfrage an die Search-Komponente angestoßen. Die Search-Komponente liefert genau jenes Referenzmodell, das unter diesem Namen hinterlegt ist.

**Einschränkung der Auswahl über die Kategorie** Jedes Objekt im System ist unter einer bestimmten Kategorie ablegt. Diese Kategorien werden dem Benutzer in einer Drop-box zu Verfügung gestellt. Der Benutzer hat so die Möglichkeit, die Auswahl der Namen über die Kategorie einzuschränken.

**Einschränkung der Auswahl über einen Tag** In der Abbildung wird durch das Suchfeld die Aktion „Auswahl über einen Tag“ modelliert.

Dem Benutzer wird die Möglichkeit geboten, diejenigen hinterlegten Referenzmodelle, die bestimmte Eigenschaften besitzen, über das Suchfeld vom System aufzurufen. In dem Suchfeld befindet sich ein Freifeld „Text“, das einen alleinstehenden Begriff oder auch einen Satz verarbeiten kann.

Mit der Enter-Aktion des Benutzers wird eine Anfrage mit dem eingegebenen Text an die Kern-Anwendung gesendet. Diese vergleicht dann den Suchbegriff mit den hinterlegten Meta-Informationen aller Referenzmodelle.

Die Kern-Anwendung liefert als Ergebnisse alle Referenzmodelle, für die dieser Suchbegriff als Meta-Information im System hinterlegt ist. Das Suchergebnis besteht aus keinem oder mindestens einem Datensatz.

Egal welche Wahl der Benutzer trifft, um die Auswahl der Namen einzuschränken, am Ende muss er das Objekt über den Namen wählen. Sobald ein Auswahl getroffen wurde, wird die Suchanfrage gestartet. Als mögliches Ergebnis liefert es die Position des Objektes im Bild zurück. Diese wird visuell in dem Bereich „Anzeige“ des GUI angezeigt, und zwar wird das Objekt dabei markiert.

Der unterste Bereich der Benutzereingabe enthält noch eine textliche Beschreibung der Lage des Objekts im Raum. Dies implementiert den Anwendungsfall „Verortung des Objektes im Raum“. Hier wird der formulierte Text ausgegeben, der die räumliche Lage des Objektes beschreibt.

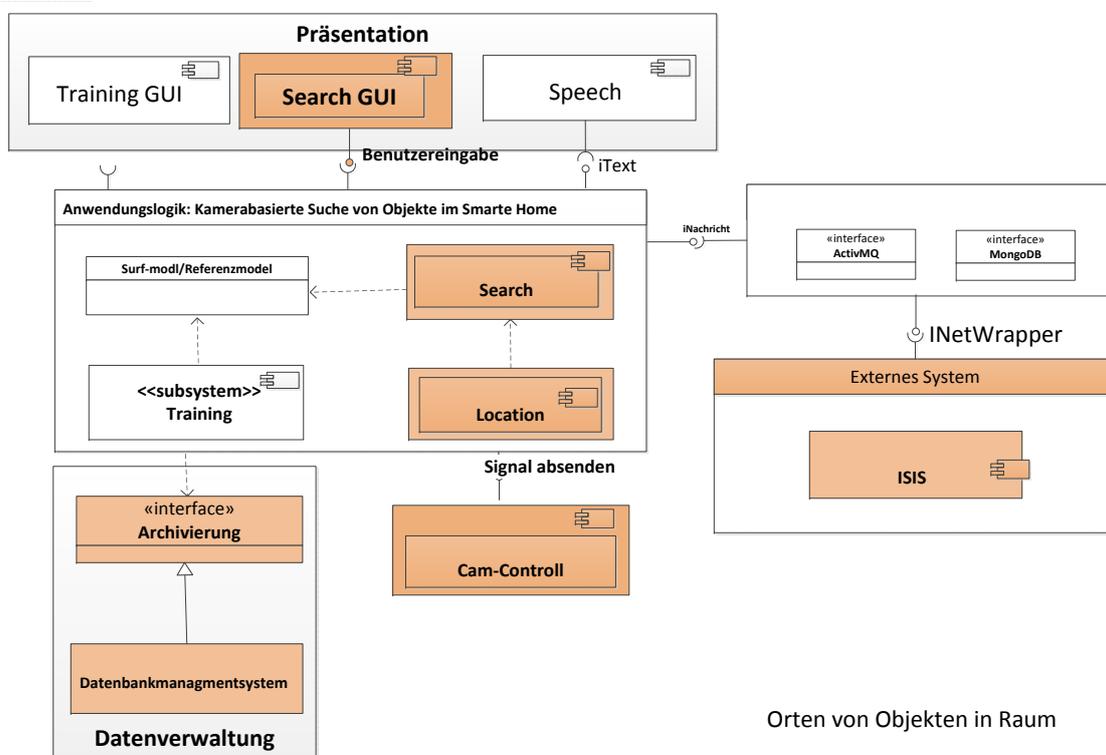
### 4.3. Subsystem Location: Orten von Objekten im Raum

Die Anwendungslogik enthält als letzte Subkomponente die Location-Komponente.

Sie beschreibt das Konzept für den in der Analyse aufgeführten Teilbereich: „Orten von Objekten im Raum“ 3.4.3.

In der Abbildung 4.9 sind die zuständigen Komponenten in der Systemarchitektur farblich markiert und im Folgenden werden sie in ihrem Zusammenspiel beschrieben.

Die Location-Komponente ist in zwei Teilbereiche unterteilt. Der erste Teilbereich hat die Aufgabe, zu einem Bildpunkt in 2D-Koordinaten den zugehörigen Raumpunkt in 3D-Koordinaten zu bestimmen; dazu bedarf es der Kameraschnittstelle. Der zweite Teil der Location-Komponente



Orten von Objekten in Raum

Abbildung 4.9.: Orten im Raum

übernimmt die sprachliche Beschreibung der Pose im Raum und benötigt Informationen über die relative Lage von Objekten im Raum, die als Nachbarobjekte zu ermitteln sind.

Hierfür bindet die Location-Komponente die LP-Kommunikations-Komponente ein, mit der externe Dienste und weitere Services im dem Living Place Hamburg über ActiveMQ zugänglich sind. Sie ermöglicht es, die benötigten externen Dienste einzubinden.

Die Speech-Komponente in der Präsentationsschicht gibt den formulierten Satz akustisch wieder.

Im Folgenden wird das Konzept und die Verantwortlichkeit der Lokation-Komponente genauer erläutert.

### 4.3.1. Location-Komponente

Zur Realisierung der funktionalen Anforderungen integriert die Lokation-Komponente folgende Module:

**Objekt anzeigen** Diese Klasse (Modul) ist zuständig für die Anzeige des Objekts im Bild, wie im Anwendungsfall „Objekte anzeigen“ beschrieben [3.2.1](#).

Hierfür wird die Position des Bewohners von einem externen Service des Living Place Hamburg über die Kommunikationsschnittstelle erfragt. Befindet sich der Bewohner in der Wohnung, ermittelt es den ihm nächstgelegene Monitor und stellt eine Verbindung dazu her. Dann wird das Bild der Kamera mit dem dort befindlichen Suchobjekt angezeigt. Zusätzlich integriert es ein 2D-Raummodell des Living Place Hamburg, in dem die Position des Objekts markiert und hervorgehoben wird. Dieser Modul kann erweitert werden durch Schnittstellen zu weiteren Geräten wie z. B. Smartphones, Moutitouch oder auch dem Fernseher im Living Place Hamburg für die Anzeige des Objektes in den Kamerabilder.

**3D-Raumkoordinaten bestimmen** Dieses Modul ist für die Bestimmung des Raumpunktes des Objektes zuständig. Sie integriert dazu die mathematische Formeln und Verfahren, um zu einem 2D-Bildpunkt den Raumpunkt zu bestimmen. Welche der mögliche Methoden zur Berechnung des 3D-Raumpunktes in dieser Masterarbeit zum Einsatz kommen, ist in dem Abschnitt „Positionsbestimmung im Raum“ [2.3](#) in Kapitel [2](#) erläutert und aufgeführt.

Welche Verfahren und Methoden mit welcher Kameras eingesetzt werden sollenn, bestimmt das Modul „Verfahren ermitteln“.

**Verfahren ermitteln** Dieses Modul ist für die Kommunikation mit den Kameras verantwortlich. Sie baut die Verbindung mit ihnen über die Kameraschnittstelle auf. Sie sendet Steuerkommandos an die Kameras über die Cam-Control-Komponente und fragt deren aktuelle Position ab. Sie erfragt, wie weit eine Kamera gezoomt und geschwenkt ist und welche Neigung sie hat. Die Klasse behandelt auch die Sichtfelder der Kameras in

der Wohnung. Sie erhält die Information, in welchem Raum welche Kamera montiert ist, bestimmt für jede Kamera das Sichtfeld und speichert diese Informationen. So legt sie fest, wieweit die Zuständigkeit einer Kamera reicht. Auf Basis dieser Informationen analysiert sie, wie viele Kameras für die Berechnung zu Verfügung steht.

Sie entscheidet, welche Methoden und Verfahren in welchen Bereich eingesetzt werden können. Dafür beantwortet sie folgende Fragen: Welche Kamera hat das Objekt gefunden? und: Steht eine zweite Kamera im gleichen Bereich zu Verfügung?

Ist das Verfahren zur Berechnung der 3D-Raumpunktes festgelegt, so ist Sie zuständig für die Einstellung der Kameras für die Berechnung des 3D-Raumpunktes. Hierfür integriert sie eine Kamerassteuerung, die Schwenkung, Neigung und evtl. Zoom der Kamera errechnet und die Steuerbefehle an die entsprechende Kameras über das „Cam-control“ versendet.

Das Konzept der Kamerasteuerung ähnelt der Funktionalität der Steuerlogik in der Search-Komponenten, sie hat aber einen ganz anderen Schwerpunkt. Hier liegt der Schwerpunkt darauf zu entscheiden, welche Methodik für das Ermitteln des Raumpunktes eingesetzt werden können. In der Praxis wird sich erweisen, ob diese überlappenden Eigenschaften der Module in eine separate Komponente extrahiert werden sollten.

Auf Grundlage dieser Klassen löst die Location-Komponenten das Problem der Berechnung des 3D-Raumpunktes.

**Das Konzept:** An Hand der Abbildung 4.10 wird das Konzept der Location-Komponenten erläutert. Die Location-Komponente wird von der Search-Komponente aufgerufen. Sobald die Position des Objektes berechnet ist, wird der Bildpunkt an die Location-Komponente weitergeleitet.

Weiterhin liefert die Search-Komponente die Information, in welchem Bereich des Living Place Hamburg , bzw. von welcher Kamera das Objekt wiedergefunden wurde und wie die Kameraeinstellung dabei war, d. h. sie liefert die Informationen, wie weit die Kamera gezoomt und geschwenkt ist und welche Neigung sie hat.

Damit entscheidet die Location-Komponente, welche Methoden und Verfahren für die Berechnung des 3D-Raumpunktes eingesetzt werden sollen. Es stellt die Kamera für die Berechnung des 3D-Raumpunktes ein und verwendet die entsprechenden mathematische Verfahren zur Berechnung des 3D-Raumpunktes an Hand der Kameradaten.

**Ein Beispiel der Kameraeinstellung** für das Verfahren Stereo Vision wäre:

Auf Grundlage der von der Search-Komponente gelieferten Informationen berechnet das Module „Verfahren ermitteln“, welche weitere Kamera (KAMERA2) das Objekt eventuell im Sichtfeld hat.

Es wird errechnet, wie weit die KAMERE2 geschwenkt, geneigt und evtl.gezoomt werden soll, um das Objekt erfassen zu können. Die errechnete Werter werden über die „cam-control“ an die entsprechende Kameras übermittelt.

Für die Suche des Objektes im Bilder der KAMERA2 wird die Suchfunktion der SURF-Klasse verwendet und der Bildpunkt des Objektes in den Bildern der KAMERA2 ermittelt. So wird das Objekt von zwei Kameras erfasst. Nun ist das Modul dafür verantwortlich, die beiden Kameras für das Stero-Vision Verfahren entsprechend einzustellen. Beide Kameras werden so eingestellt, dass das Objekte im Focus der Kameras liegt, d. h. das Objekt ist im Mittelpunkt der Bilder. Dieser Einstellung wird an das Modul übermittelt, das auf Basis dieser Informationen den Raumpunkt mit de gewählten Methode berechnet.

Eine Problematik der integrierten Kameras in dem Living Place Hamburg ist, dass nicht in jedem Bereich der Wohnung zwei Kameras montiert sind. Somit ist ein Verfahren zur Berechnung von Raumpunkten, das zwei Kameras benötigt, nicht in allen Bereichen der Wohnung anwendbar. Deshalb ist dass Konzept so gewählt, dass das Modul „3D-Raumpunkt ermitteln“ alle benötigten Verfahren Verfahren zur Berechnung des Raumpunktes integriert. Das Modul „Verfahren ermitteln“ trifft hierbei die Entscheidung.

### 4.3.2. Sprachliche Beschreibung der Pose im Raum

Eine weitere Herausforderung, die sich aus den Szenarien ergibt, ist die sprachliche Beschreibung der Pose des Objektes im Raum. Zur Erinnerung: Als Pose oder räumliche Lage wird im technischen Zusammenhang die Kombination von Position und Orientierung eines Objektes bezeichnet.

Für die sprachliche Lokalisierung eines Objektes im Raum muss nun der Zusammenhang zwischen den räumlichen und sprachlichen Strukturen betrachtet werden. Dies ist im Zusammenhang mit dem Thema Raumkognition und dem Sonderforschungsprojekt SFB/TR 8 Spatial Cognition<sup>2</sup> bereits in dem Abschnitt 3.5 behandelt worden.

Eines der Forschungsergebnisse ist, dass räumliches Wissen nötig ist, um die Lage eines Objektes räumlich beschreiben zu können. Daraus wurden die Anforderungen für die Beschreibung der Pose gefolgert, nämlich dass geeignete Objekte in der Nähe des gesuchten Objektes zu ermitteln sind und dass ein Bezugssystem bestimmt und eine Perspektive gewählt werden muss (vgl. Abschnitt 3.4.3).

Die Lösung des Ganzen bietet das dreidimensionale auf Basis der IFC-Klassen generierte Modell des Living Place Hamburg (in der Folge 3D-Modell) an. Entwickelt wurde es von Bastian Karstett im Rahmen seiner Masterarbeit mit dem Titel „Das 3D-Modell modelliert alle

---

<sup>2</sup><http://www.sfbtr8.uni-bremen.de/index.html>

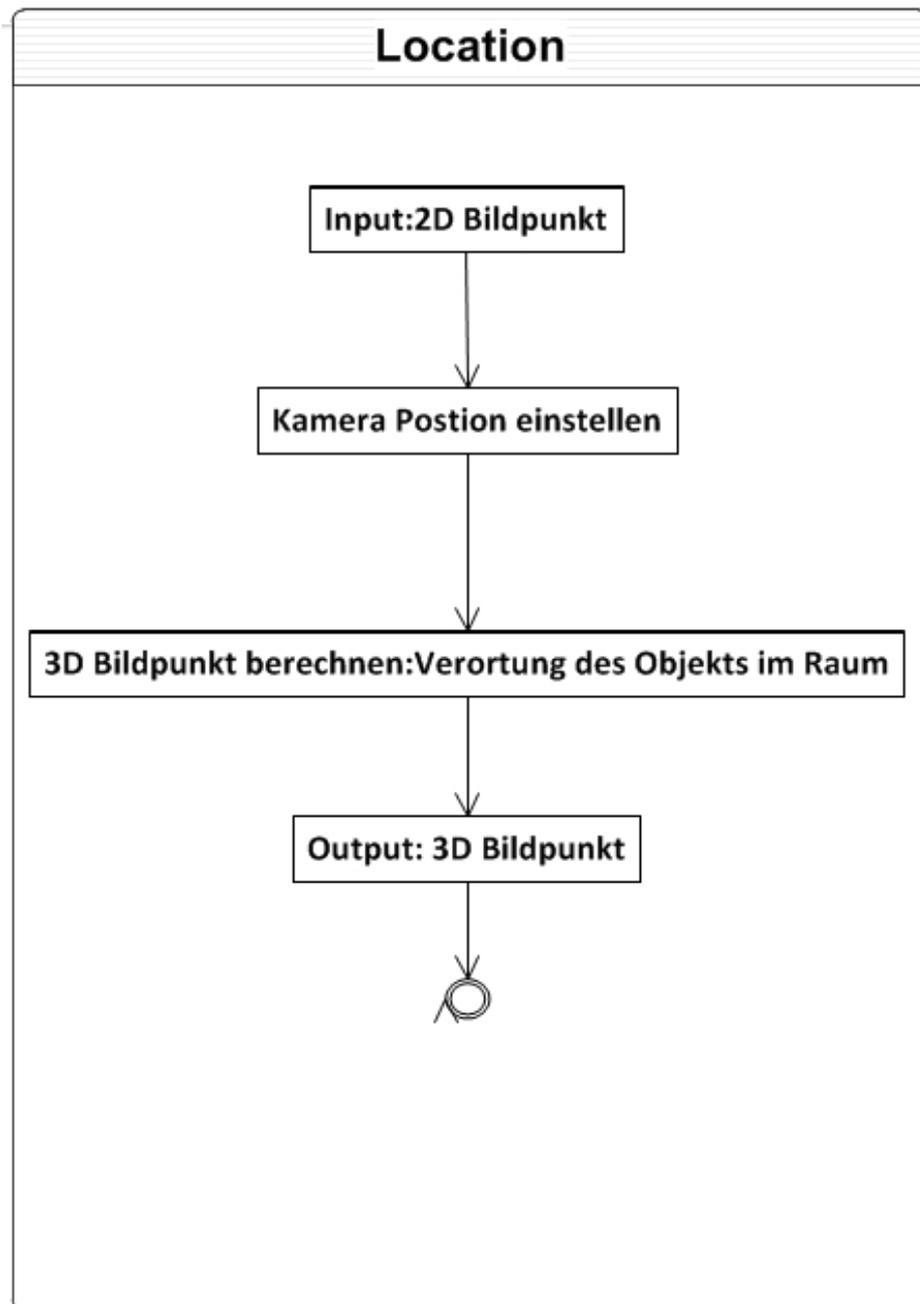


Abbildung 4.10.: Ablauf Orten im Raum

im Living Place Hamburg vorhandenen Möbel und erfasst deren Positionen“ (Vgl. [Karstaedt, 2012](#)).

Das 3D-Modell des Living Place Hamburg integriert in sich das räumliche Wissen; das kann bei der Formulierung einer sprachlichen Lokalisierung helfen.

Mit dem 3D-Modell ist der „Indoor Spatial Information Service“ (ISIS) implementiert. ISIS ist ein Serviceanbieter für Anfragen über räumliche Informationen und die Relationen von Möbeln bzw. Objekten im Living Place Hamburg.

Um dem Bewohner auf die Serviceanfrage „Wo ist das Buch?“ mit umgangssprachlichen Lokalisationsäußerungen antworten zu können, wurde das Szenario in [Abbildung 4.11](#) entwickelt, es zeigt die Interaktion zwischen dem kamerabasierten Objektsuch-System und dem 3D-Modell im Living Place Hamburg.

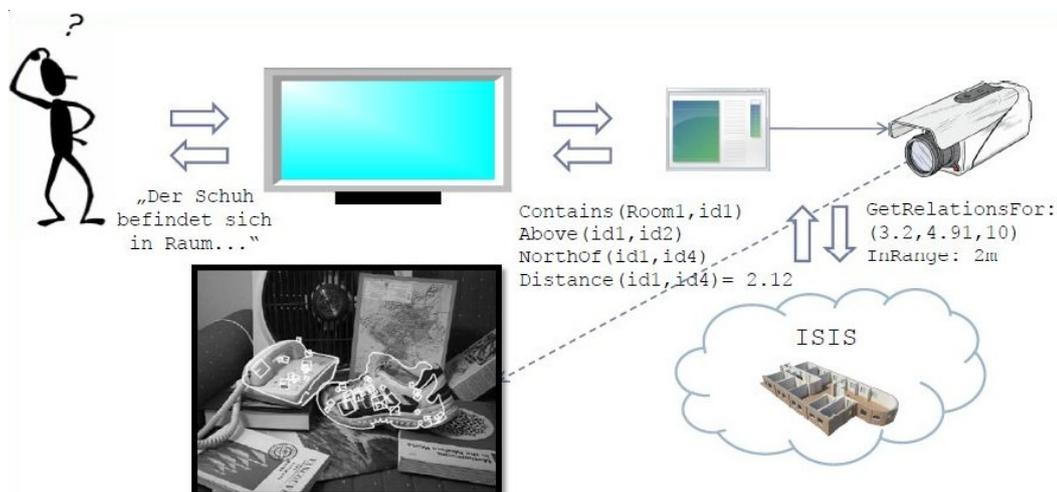


Abbildung 4.11.: Interaktion zwischen Anwender, Kamerasystem und ISIS (erstellt von. [Karstaedt, 2011a](#))

Nachdem das gesuchte Objekt gefunden wurde, wird mit den ermittelten 3D-Koordinaten des Objektes die Serviceanfrage „Welche Objekte befinden sich in unmittelbarer Umgebung des gefundenen Objektes?“ an den ISIS gestellt (vgl. [Abb. 4.12](#)).

Der ISIS liefert dann Informationen darüber, in welchem Raum sich das gesuchte Objekt bzw. seine Koordinaten befinden und welche Nachbarobjekte in seiner unmittelbaren Umgebung im 3D-Modell vorhanden sind. Darüber hinaus liefert er Informationen über räumliche

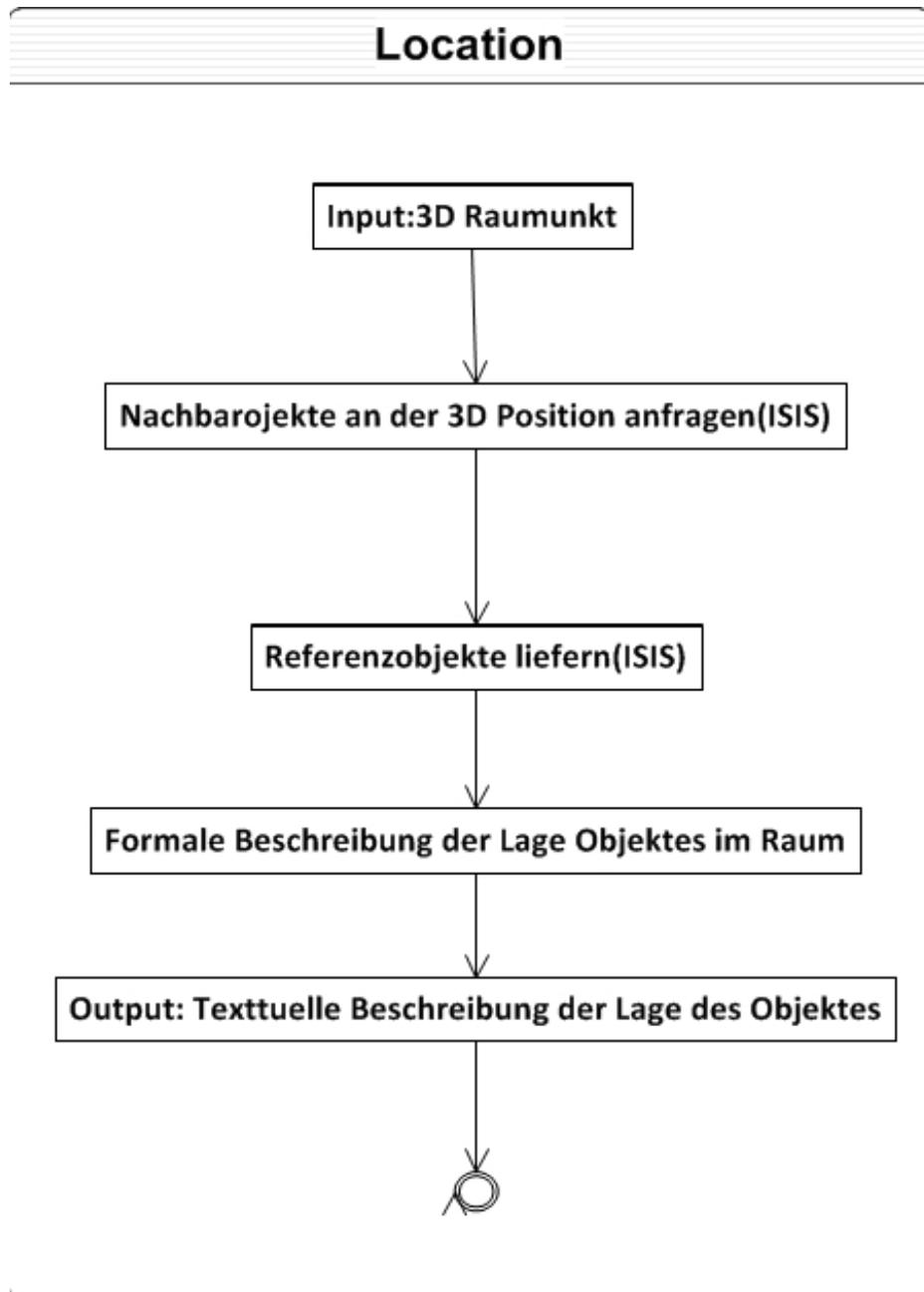


Abbildung 4.12.: Ablauf sprachliche Beschreibung

Relationen der Nachbarobjekte untereinander und in Bezug auf das gesuchte Objekt bzw. dessen Raumkoordinate.

Für die räumlichen Relationen wurden der im Projekt „Indoor Spatial Information Service“ Prädikate wie etwa „contains“, „above“, „below“, „northOf“ usw. definiert. Um diesen Prädikaten weitere semantische Informationen zuordnen zu können, ordnete Bastian [Karstaedt \(2011a\)](#) in Rahmen seiner Masterarbeit diese in Kategorien wie topologische, direktionale und metrische Relationen ein.

Damit der Bewohner solche Äußerungen verstehen kann, werden die gewonnenen Informationen von der Lokation-Komponente analysiert und interpretiert, um diesen semantischen Informationen Bedeutungen zuweisen zu können. Weiterhin entscheidet die Lokation-Komponente auf Basis der gewonnenen Informationen darüber, welches der von ISIS gelieferten Objekte ein gutes Referenzobjekt darstellt, um eine aussagekräftige Lokalisierung formulieren zu können. Die hierfür nötigen Konventionen und Vereinbarungen werden im Abschnitt [5.4.1](#) erläutert.

In einem weiteren Schritte werden die entsprechenden Objekte als Referenzobjekte gewählt und der entsprechende Ausgabesatz formuliert.

Die Kooperation mit dem Projekt ISIS ermöglicht es also dem System, dem Bewohner die Position des gesuchten Objektes wie folgt zu formulieren: „Das Buch ist im Schlafbereich auf der Kommode“.

Dieser Satz wird schließlich an die Speech-Komponente weitergegeben, die ihn dann dem Bewohner über den Lautsprecher mitteilt.

# 5. Umsetzung

In diesem Kapitel wird ausgeführt, wie die beschriebenen Konzepte im Rahmen dieser Masterarbeit realisiert wurden. Ausgangspunkt der Masterarbeit war die Aufgabe, die sechs im Living Place Hamburg integrierten PTZ-Domekameras dafür zu benutzen, Objekte im Raum zu suchen. Dazu sollte ein Prototyp entwickelt und getestet werden.

Nun soll also geklärt werden, wie die einzelnen Komponenten implementiert wurden. Darüber hinaus werden in den einzelnen Abschnitten die Ergebnisse einiger Experimente und Versuche aufgeführt.

Implementiert werden sollten:

- eine Benutzeroberfläche,
- ein für die Objekterfassung,
- einer für die Objektsuche und
- einer für die sprachliche Ausgabe des Suchergebnisses, sowie
- Anbindungen an eine Datenbank und bereits existierende Dienste.

In folgenden wird aufgeführt, welche Teile des Konzepts in welchen Modulen umgesetzt wurden. Zunächst aber wird beschrieben, welche Tools hierfür verwendet wurden und warum die Umsetzung so stattgefunden hat.

Dann wird im Abschnitt [5.3](#) erläutert, wie Objekte im System modelliert werden und mit welchen Modulen die Suche stattfindet. Im folgenden Abschnitt [5.4](#) wird erläutert, welche Konventionen und Regelungen getroffen wurden, um eine sprachliche Beschreibung mit Hilfe von ISIS zu gewährleisten.

## 5.1. Verwendete Tools und Bibliotheken

Im Rahmen der Masterarbeit wurden Verfahren

- zur Modellierung von Objekten und

- zur Suche und zum Wiederfinden von Objekten

analysiert

Nun werden folgende Fragestellungen thematisiert:

- Welche Programmiersprachen soll benutzt werden?
- Welche Bildverarbeitungs-Bibliothek bietet die beste Lösung für die Umsetzung des Konzeptes?

### 5.1.1. Programmiersprache

Das kamerabasierte Objektsuch-System wurde implementiert in der Programmiersprache C# unter Verwendung des Microsoft .NET4-Frameworks. Als Entwicklungsumgebung diente Microsoft Visual Studio 2010. Die ersten Tests und das Verfahren zur Modellierung und Suche des Objektes in Bildern wurden hingegen in C++ implementiert.

Eine Anforderung an die Implementierung war es, eine Lösung zu finden, die sich dynamisch anpassen lässt und in die bereits integrierte Technik des Living Place Hamburg einfügt. Im Laufe der Masterarbeit und der Weiterentwicklung des Living Place Hamburg hat sich die Programmiersprache C# dafür als günstiger erwiesen.

Dazu trug bei, dass das 3D-Raummodell des Living Place Hamburg ebenfalls in C# implementiert ist. Für die Kommunikationsschnittstelle wurden ein C#- und ein Java-Wrapper entwickelt. Schließlich sind die Bildverarbeitungs-Prozesse in C# bzw. C++ viel performanter als die in der Programmiersprache Java.

### 5.1.2. OpenCV

Die *Open Source Computer Vision Library* (OpenCV)<sup>1</sup> ist die am häufigsten verwendete und auch erfolgsversprechendste Bildbe- und -verarbeitungsbibliothek. Es ist eine sehr große und performante Programmbibliothek, die Algorithmen und Methoden nach den neuesten Forschungsergebnissen enthält, unter anderem in den Bereichen Bildverarbeitung, maschinelles Sehen und maschinelles Lernen. Als freie Software steht sie unter der BSD-Lizenz zur Verfügung.

OpenCV bietet ein plattformunabhängiges API an, das ein breites Anwendungsgebiet hat und somit Module mit Lösungen, Methoden, und Algorithmen für viele Anwendungsfelder zur Verfügung stellt.<sup>2</sup>

---

<sup>1</sup>Homepage:<http://opencv.org/>

<sup>2</sup>Homepage der OpenCV-Module: <http://docs.opencv.org/modules/refman.html>

Um aus Messbildern eines Objektes seine räumliche Lage oder dreidimensionale Form zu bestimmen sind viele Verfahren der Photogrammetrie und des stereoskopischen Sehens (Stereopsis) integriert. Darüber hinaus enthält die Bibliothek unter anderem Algorithmen für die 3D-Funktionalität, verschiedene sehr schnelle Filter (z. B. Gauß) und Funktionen für die Kamerakalibrierung.

In dieser Masterarbeit kamen folgende Module zum Einsatz:

- 2D- und 3D-Merkmale (z. B. SURF, Interest-Operator oder Deskriptoren),
- Nächste-Nachbarn-Klassifikation.

### 5.1.3. SURF-Implementierung

In dieser Arbeit wird für die Modellierung und Erkennung von Objekten in Bildern der SURF-Algorithmus verwendet. Wie im Kapitel 2.2.2 bereits dargestellt, ist dies ein in der Bildverarbeitung bestbewertetes rotations- und skalierungsinvariantem featurebasierten Verfahren zur Objekterkennung und Objektverfolgung in Videosequenzen. Die theoretischen Grundlagen des Verfahrens finden sich im Kapitel „Verfahren der Objekterkennung“ 2.

Als Implementierung des SURF-Algorithmus wurde die C#-Bibliothek *OpenSURF* von Chris Evans verwendet.<sup>3</sup>

**Warum SURF-Verfahren?** Es musste ein Verfahren verwendet werden, das robust gegenüber Skalierung, Rotation und Verdeckung ist.

Warum muss es robust sein gegenüber Skalierung? Weil die verwendeten Kameras eine Schwenk-Neige-Zoom-Funktion besitzen. Damit die Kamerabilder mit unterschiedlichen Brennweiten, also unterschiedlichen Abbildungsmaßstäben benutzt werden können, ist also Skalierungstoleranz eine Grundvoraussetzung.

Genauso wichtig sind auch die Toleranz gegenüber Rotation und Verdeckung, denn in der Realität liegen die meisten Objekten in einem Wohnraum teilweise verdeckt unter anderen Gegenständen und wurden natürlich nicht an der Kamera ausgerichtet. Ohne eine solche Robustheit hätte die kamerabasierte Objektsuche nur eine sehr eingeschränkte Anwendbarkeit.

Ein weiterer Grund, warum das SURF-Verfahren verwendet wurde, ist die große praktische Erfahrung mit dem Verfahren; so wurde es an der HAW Hamburg in vielen Forschungsprojekten eingesetzt und implementiert. In Abschnitt „Aktuelle Forschungsprojekte“ 3.5 sind

---

<sup>3</sup>Siehe dazu: <http://www.chrisevansdev.com/computer-vision-opensurf.html>

zwei dieser Forschungsprojekte aufgeführt, um Objekte im Raum zu lokalisieren. Von deren Erfahrungen konnte ich profitieren und sie für meine Umsetzung verwenden.

Ziel der Implementierung ist, praktisch in Erfahrung zu bringen, welche Eigenschaften die SURF-Implementierung besitzt und wie diese für mein Softwaresystem optimiert werden können. Dazu wird im Abschnitt 5.3.2 anhand von Experimenten gezeigt werden, mit welchen Methoden und Einstellungen ein Referenzmodell erstellt und ein Objekt im Living Place Hamburg gefunden werden konnte.

#### 5.1.4. Microsoft Speech API

Unter Sprachsynthese versteht man die künstliche Erzeugung der menschlichen Sprechstimme. Ein Text-to-Speech-System (TTS) oder Vorleseautomat wandelt Fließtext in eine akustische Sprachausgabe um.

Das *Speech Application Programming Interface* (SAPI) ist eine Schnittstelle zur Anbindung von Bibliotheken zur Sprachsynthese und Spracherkennung unter dem Betriebssystem Microsoft Windows.<sup>4</sup>

Die SAPI-5.3-kompatiblen Stimmen (Engines) sind in das Windows Betriebssystem XP integriert und benötigt mindestens das .NET-Framework in Version 3.0.

Die *ISpRecoContext* Text-to-Speech-API bietet die Möglichkeit, das Geschlecht, Alter und den Namen des Sprechenden zu wählen. Die Sprache kann in Betonung, Geschwindigkeit und Lautstärke variiert werden.

Die Aussprache der Wörter kann sprachspezifisch gewählt werden. Es wird eine Auswahl von Stimmen in den Sprachen Englisch, Französisch, Deutsch, Italienisch und Spanisch angeboten. Es können auch weitere Stimmen aus dem Internet heruntergeladen werden.

Die in Windows integrierten Microsoft-Stimmen haben eine mittlere Qualität, besonders die englische MS-Stimme ist ausgereifter als die deutsche. Die deutsche Sprachmelodie ist schwer verständlich, daher wird die englische Aussprache verwendet.

## 5.2. Klassenaufbau

In der Konzeption ist das Gesamtkonzept dargestellt worden. Im Folgenden soll die Beziehung der einzelnen Komponenten zu den implementierten Modulen dargestellt werden.

---

<sup>4</sup>[http://msdn.microsoft.com/en-us/library/ms720151\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms720151(v=vs.85).aspx)

Dem in der Abbildung 5.1 dargestellten Surfmodell entspricht die Implementierung des Konzepts SURF-Klasse (vgl. Abschnitt 4.1.5).

Dem in der Abbildung 5.1 dargestellten Modul TeachObjekt entspricht die Training-Komponente und SearchObjekt implementiert die Search-Komponente.

Das Projekt ist nach dem Model-View-Controller-(MVC)-Muster aufgebaut. Die Klasse Surfmodel stellt dabei das Modell dar und enthält alle benötigten Basismethoden zum Erstellen des Referenzmodells. Sie bindet die OpenCV-Bibliotheken für das SURF-Verfahren ein. TeachingObjekt und SearchingObjekt stellen eine Art Controller dar.

Die Views des Projektes sind die in der Konzeption bereits vorgestellten Benutzeroberflächen Training-GUI und Search-GUI. Die Benutzereingaben werden vom Controller verarbeitet. Er leitet die Anfragen von den Views an das Modell (Surfmodel) weiter. Dieses verarbeitet die Eingaben und liefert die angefragten Daten zurück an den Controller.

Das Modell (Surfmodel) ist verantwortlich für die Verarbeitung und das Manipulieren (Löschen, Hinzufügen, Verändern) der Daten. Es erstellt aus den Eingaben der Controller den neuen Datensatz.

Es stellt die Ergebnisse dann dem Controller zu Verfügung. Dieser leitet die Anfragen an die entsprechende Views weiter, die diese dann dem Benutzer präsentieren.

Durch das Verwenden des MVC-Patterns wird die Präsentationsschicht von der Datenverwaltungsschicht getrennt.



Abbildung 5.1.: Ausschnitt der Klassendiagramme für Surfmodel, Teaching-, Search Objekt

### 5.3. Modellierung und Suche von Objekten

Die Implementierung der Schnittstellen des kamerabasierten Objektsuche-Systems waren der erste Ansatz, um das Konzept umzusetzen.

Für den ersten Prototypen wurde eine Benutzerschnittstelle implementiert, die sowohl das Konzept der Referenzmodell-Erstellung wie auch die Objektsuche in Bildern integriert. Die Implementierung wird im folgenden Abschnitt beschrieben.

**Benutzerschnittstellen** In der Konzeption wurden zwei Benutzeroberflächen als Schnittstelle zwischen dem Benutzer und dem System vorgestellt.

Der erste Schritt der Umsetzung war die Erstellung einer in C# implementierten Benutzerschnittstelle für die automatische Erstellung und Speicherung von Referenzmodellen. Denn die Erstellung des Referenzmodells ist Voraussetzung für die Suche nach dem Objekt im Raum.

In der Konzeption wurde das Konzept der Teaching-GUI-Benutzeranwendung erläutert. Hierfür wurde der erste Prototyp erstellt, dieser wird in der Abbildung 5.2 dargestellt.

In dem Bereich Teaching-Referenzen kann der Benutzer jederzeit Referenzbilder im Gesamtsystem hinterlegen. Hierfür betätigt er den Button „Start Teaching“. Das von der Kamera erfasste Bild wird in dem mittleren Bereich der Benutzeroberfläche angezeigt. In dem Feld „Name“ wird der Name des Referenzmodells eingegeben und mit dem Button „Save Image“ wird das angezeigte Bild gespeichert und daraus das Referenzmodell erstellt. Die Action-Buttons implementieren auch die in der Konzeption aufgeführten Einschränkungen und Sicherheitskonzepte. Ein Referenzbild darf also nicht gespeichert werden, solange kein Name eingetragen ist. Aus allen Bildern, die unter einem Namen gespeichert wurden, wird ein Referenzmodell erstellt. Es bleibt dem Benutzer überlassen, darauf zu achten, dass die Zuordnung von Bildern und Namen korrekt ist.

In der in Abbildung 5.2 dargestellten Benutzeroberfläche ist die Funktionalität der Objektsuche im Raum mit integriert. Der Benutzer kann mit dem Button „Change References“ ein Referenzmodell aussuchen. Es öffnet sich ein Auswahlfenster, in dem die Bilder zu allen gespeicherten Referenzbildern angezeigt werden.

Für die erste Implementierung wurde aus Zeitgründen auf die Einbindung einer Datenbank verzichtet. Stattdessen werden die erstellten Referenzbilder unter den angegebenen Namen in einer Ordner-Struktur abgelegt und die dazugehörige Meta-Informationen in einer Textdatei abgelegt. Diese Einschränkung führt dazu, dass die Selektion der Referenzmodelle nur anhand des Namens stattfinden kann.

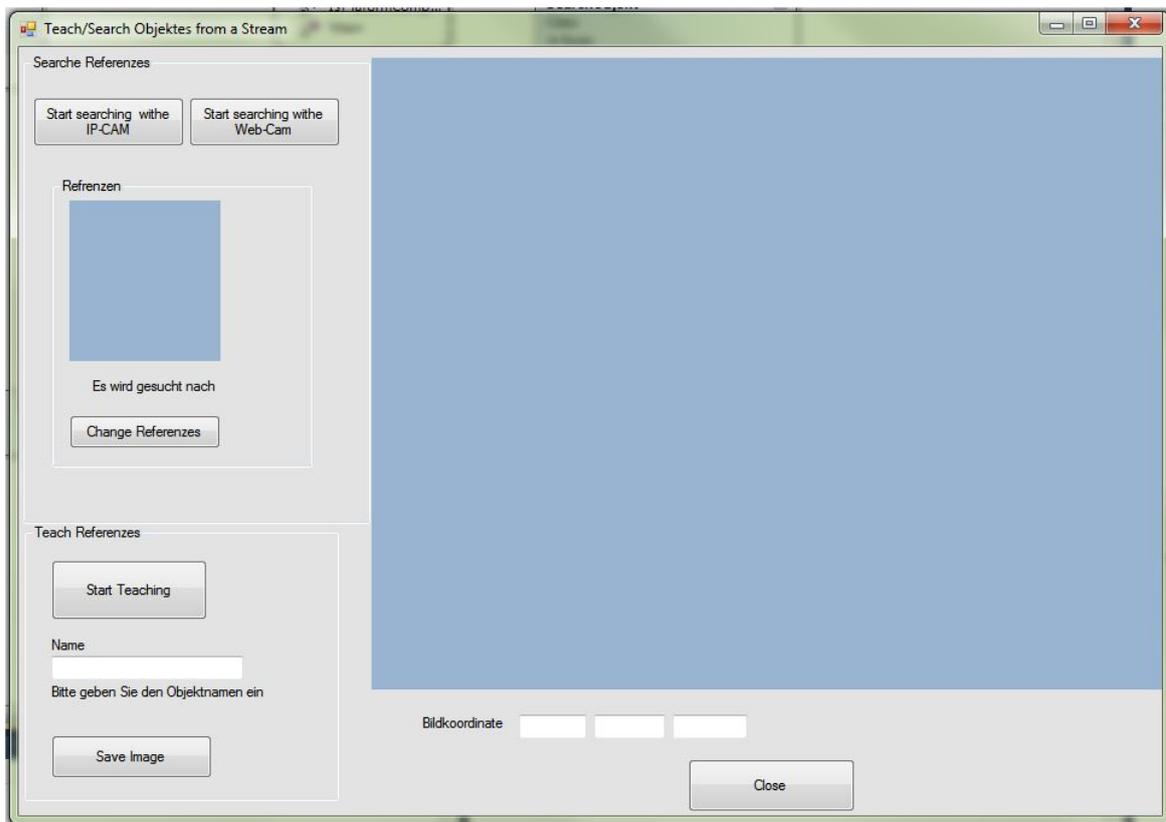


Abbildung 5.2.: Benutzerschnittstelle

Die in der Abbildung 5.2 dargestellte Benutzeroberfläche integriert auch die Umsetzung des beschriebenen Search-GUI Konzepts. Durch den Button „Start Searching with IP-Cam“, wird die Suche nach dem gewählten Referenzmodell in den Bilder einer Kamera im Living Place Hamburg gestartet.

Für die Auswahl des Referenzmodells anhand des Namens bietet die Benutzeroberfläche den Bereich Referenzen an. Dort wählt der Benutzer mit dem Button „Change References“ ein Referenzmodell aus; das Referenzbild erscheint daraufhin in der Image-Box des Referenzbereichs.

Die Live-Bilder der Kameras im Living Place Hamburg werden in der großen Image-Box im mittleren Bereich angezeigt. Hier kann der Benutzer sehen, wie die Kamera das Objekt sucht. Ist die Suche erfolgreich, dann wird dem Benutzer das Objekt markiert angezeigt. So wird der Anwendungsfall „Objekt anzeigen“ realisiert.

### 5.3.1. Training/Teaching Objekt

Der in der Konzeption dargestellte Komponente „Training“ umfasst die automatische Erstellung und Speicherung von Referenzmodellen aus eingegebenen Referenzbildern.

Die Arbeitsweise des SURF-Algorithmus wurde in dem Abschnitt 2.2 bereits erläutert. Sie besteht zusammengefasst aus den folgenden Schritten:

- Der Ermittlung von potentiellen markanten Punkten (Keypoints ) aus den Referenzbildern.
- Anschließend werden diese auf Stabilität geprüft, um daraus die potentiellen Merkmale zu bestimmen.
- Im dritten Schritt wird für jeden markanten Punkt die Hauptorientierung berechnet.
- Im letzten Schritt wird zu jedem markanten Punkt der Deskriptor ermittelt.

In dem Abschnitt 2.2 des Kapitels 2 ist bereits ausgeführt, wie die Merkmale aus den Referenzbildern extrahiert werden. Ein Merkmal ist dabei definiert als ein Tupel aus einer Bildposition  $(x, y)$ , einer Hauptorientierung und einem Deskriptor.

Die SURF-Methode aus der OpenCV-Bibliothek stellt hierfür alle benötigten Methoden zur Verfügung, die durch den Modul Surfmodell aufgerufen werden.

### 5.3.2. Test des SURF-Algorithmus

Das Ziel der Implementierungsphase war es ja, praktisch in Erfahrung zu bringen, welche Eigenschaften die SURF-Implementierung besitzt, und wie sie eventuell für das kamera-basierte Objektsuch-System optimiert werden könnte, so ging es u. a. darum, mit welchen Parametereinstellungen die SURF-Methode die am besten markante Punkte (Keypoints) aus den Referenzbilder erstellen kann.

Im Folgenden werden die Tests und deren Ergebnisse aufgeführt.

**Parametereinstellung** Die Erstellung bzw. Extraktion von Referenzpunkten aus einem Bild oder einem Frame erfolgt über die SURFDETDDES-Methode. Die Methode bekommt als Eingabe ein Bild und liefert die extrahierten Bildpunkte mit ihren 64-dimensionalen Deskriptoren zurück.

Im ersten Schritt ging es darum herauszufinden, mit welchen Parametereinstellungen eine optimale Erstellung von Deskriptoren aus den Bildern gelingt. Hierfür wurden einige Referenzmodelle mit unterschiedliche Parametereinstellungen erstellt. Um die Qualität der Merkmale des Referenzmodell zu messen, wurde versucht, mit der Matching-Methode Korrespondenzen in den den Videobildern zu finden.

Die Vergleiche haben praktisch gezeigt, dass die Bildverarbeitung mit der folgenden Einstellung für SURFDETDDES:

**Octaves:** 4  
**Intervals:** 4  
**Samples:** 2  
**Rotationinvariance:** true  
**Threshold:** 0.0001f

optimale, den Anforderungen genügende Ergebnisse für die Extraktion von Merkmalen aus einem Referenzbild liefert.

Bei den Versuchen hat sich bestätigt, dass der Kontrast ein ausschlaggebender Faktor für die Eliminierung von nicht aussagekräftige Keypoints in den Bildern darstellt. Die Bilder müssen m. a. W. einen sehr hohen Kontrast aufweisen, damit man robuste und markante Bildpunkte aus den Bildern extrahieren kann.

Das SURF-Verfahren braucht einen vordefinierten Schwellenwert (THRESHOLD), anhand dessen entschieden wird, welcher Keypoint verworfen wird, welcher erhalten bleibt. Hierfür wurde durch Experimentieren herausgefunden, welcher Schwellenwert die besten Ergebnisse erzielt.

Beim Vergleich von Bildern mit gleichem Kontrast ist der Schwellenwert THRESHOLD = 0.0002f ein guter Wert für das Auffinden von Referenzmodellen.

Die Kamerabilder im Living Place Hamburg weisen allerdings oftmals unterschiedliche Kontraste auf, daher hat sich beim Vergleich von Bildern der Wert THRESHOLD = 0.0001f besser bewährt und wurde im Folgenden für die Erstellung von Referenzmodellen verwendet.

Neben der Fragestellung, mit welchen Parametereinstellungen für die SURF-Methoden die besten Merkmale aus dem Referenzbilder extrahieren werden, war ebenfalls noch zu ermitteln, welche Auflösung und Komprimierung die Referenzbildern besitzen sollten, um ein gutes Referenzmodell zu erstellen.

**Auflösungseinstellung** Um zu testen, welche Auflösung des Referenzbilds für die Extraktion von guten Merkmalen geeignet ist, wurden zwei Messungen durchgeführt; daneben gab es noch einen weiteren Aspekt dieser Messungen, nämlich herauszufinden, wie nach kleinen Brennweitenänderungen die Referenzmodelle wieder erkannt werden.

Wie im dem Anwendungsfall „Objektsuche in Bildern“ 3.2.1 beschrieben, werden nach einem Misserfolg die Einstellungen (z. B. Brennweite oder Neigung) der Kamera häufig verändert und die Suche erneut gestartet. Dieser Schritt kann beliebig wiederholt werden. Das Ziel ist nun, dass die Suche nur wenige Frames und insbesondere nur selten Änderungen der Kameraeinstellung benötigt. Denn das Suchen des Referenzmodells (das Descriptor-Matching) kann dadurch beschleunigt werden, wenn so wenig wie möglich die Kameraeinstellung während des Descriptor-Matching geändert werden muss (z. B. wenig bis gar kein Zooming).

Der optimale Suchprozess wäre: Ein Frame der Kamera wird erfragt und das Referenzmodell wird sofort dort wiedergefunden. Somit sollte bei diesem Versuch die Frage geklärt werden, welche Eigenschaften bzw. wie viel Merkmale (Deskriptoren) ein Referenzmodell haben muss, um schon durch wenige Frames der Kameras im Living Place Hamburg wiedererkannt zu werden. Bei diesem Versuch sollte auch herausgefunden werden, mit welcher Auflösung die Referenzbilder aufgenommen werden müssen, und außerdem, ob die Objekte zur Erfassung (Erstellung des Referenzmodells) unbedingt mit den Kameras aus dem Living Place Hamburg aufgenommen werden müssen, um sie dann wiederfinden zu können.

Bei der ersten Messung wurden vom gesuchten Objekt durch Veränderung der Zoomeinstellungen der Kameras Aufnahmen in vier unterschiedlichen Auflösungen im Living Place Hamburg gemacht.

Aus diesen Bildern wurden die Merkmale mit der SURFDETDDES-Methode erstellt und dabei die benötigte Zeit gemessen. Die Ergebnisse sind in Tabelle 5.1 zusammenfassend dargestellt.

Bildpixel beim realem Zoom	Anzahl der gefundenen Merkmale	Berechnungszeit in sec.
671949	834	11,52
302428	241	3,13
134188	29	0,45
7498	8	0,13

Tabelle 5.1.: Zeitmessung bei der Extraktion der Referenzpunkte der Kamerabilder mit unterschiedlicher Einstellung des analogen Zooms [Quelle:Hosnia Najem]

Bei der zweiten Methode wurde vom gesuchten Objekt ein Kamerabild mit einer Auflösung von 680950 Pixeln aufgenommen. Diese Auflösung wurde dann viermal nacheinander manuell soweit herunter gerechnet, dass sie gleich stark skaliert wurden wie die realen Zoombilder in obiger Tabelle 5.1.

Für jede manuell berechnete Auflösung wurden die Merkmale ebenfalls mit der Methode SURFDETDDES extrahiert und dabei die Zeit gemessen. Die Ergebnisse sind in Tabelle 5.2 zusammenfassend dargestellt.



Abbildung 5.3.: Referenzbilder mit den extrahierten Merkmalen

Bildpixel beim simulierten Zoom	Anzahl der gefundenen Merkmale	Berechnungszeit in sec.
671949	834	12,41
302427	293	3,54
133188	55	0,67
84119	19	0,23

Tabelle 5.2.: Zeitmessung bei der Extraktion der Merkmale durch manuelle Berechnung der Auflösung eines Kamerabildes [Quelle:Hosnia Najem]

Die Ergebnisse zeigen, dass beim ersten Verfahren das Bild immer verschwommener wurde, da bei der Verringerung der Auflösung Bildinformationen verloren gehen.

Bei den Aufnahmen von Bildern für Referenzmerkmale eines Objektes, müssen diese Bilder sehr hohen Kontrast aufweisen. Dieser Kontrast geht aber verloren, wenn die Bilder des Objektes mit der PTZ-Kamera mit unterschiedlichen Zoomeinstellung aufgenommen werden.

Die Ergebnisse zeigen ebenfalls, dass sich für die Extraktion von Merkmalen ein Referenzbild eignet, das mit einer ganz normalen Webkamera aufgenommen wurde, die PTZ-Kameras mit ihrer hohen Auflösung sind nicht nötig.

Die nächste Anforderung an das System ist, das Objekt von allen Seiten zu erkennen. Dafür muss ein 3D-Modell des Objektes hinterlegt werden, wofür wiederum Referenzbilder von allen Seiten des Objekts notwendig sind.

Zusammenfassend werden also Bilder aus unterschiedlichen Perspektiven und mit unterschiedlichen Auflösungen zur Erstellung des Referenzmodells gebraucht.

Die Abbildung 5.4 zeigt die Erstellung der Referenzmerkmale eines Buches von allen Seiten und in verschiedenen Auflösungen. Hierfür wurden die Referenzpunkte mit deren Merkmalen in der Book.key Datei gespeichert.

### 5.3.3. Search/ Search Objekt

Nun geht es darum, wie die Objektsuche in Bildern realisiert wurde. Die im Kapitel Konzeption vorgestellte „Search“-Komponente integriert zum einen die Funktionalitäten zur Steuerung der im Living Place Hamburg eingebauten Kameras (Cam-Control in der Abb. 4.1) und zum anderen das sogenannte Matching-Verfahren, bei dem Korrespondenzen zwischen dem gewählten Referenzmodell und den extrahierten Merkmalen der vorkommenden Objekte in den Bildsequenzen der Kamera gesucht werden. Seine Grundlagen sind im Abschnitt [2.2.3] aufgeführt.



Abbildung 5.4.: Erstellung des Referenzmodells „Buch“ aus vier Perspektiven mit jeweils drei unterschiedlichen Auflösungen

Diese Funktionalitäten werden, wie in Abbildung 5.1 dargestellt, durch das Modul „Search-Objekt“ implementiert. Das Ergebnis der Suche wird auf der Benutzeroberfläche (vgl. 5.2) dargestellt, wie durch den Anwendungsfall „Objekt anzeigen“ „3.2.1“ beschrieben .

Für die erste Umsetzung des Anwendungsfalls „Objektsuche in Bildern“ 3.2.1 wurde zunächst nur mit einer Kamera im Living Place Hamburg gearbeitet, die manuell über die von den Kameras zur Verfügung gestellte Web-Oberfläche gesteuert wurde.

In der Praxis hat sich gezeigt, dass die Suche mit den Kameras beschleunigt wird, wenn das Referenzmodell Informationen über die Objekte in verschiedenen Zoomstufen enthält. Je mehr Informationen das Modell enthält, desto weniger muss die Kamera gesteuert werden. Das Matching-Verfahren wird auf diese Weise beschleunigt. Ein weiterer Performance-Vorteil liegt darin, dass die Bildsequenzen der Kamera beim Matching-Verfahren keine HD-Auflösungen aufweisen müssen.

Die Ergebnisse eines solchen „Searching“ sind in der Abbildung 5.5 dargestellt. Diese zeigt die Wiedererkennung des Objektes „Buch“ anhand des zuvor erstellten Referenzmodells (BOOK.KEY) mit den PTZ-Domekameras in zwei Szenen im Living Place Hamburg.

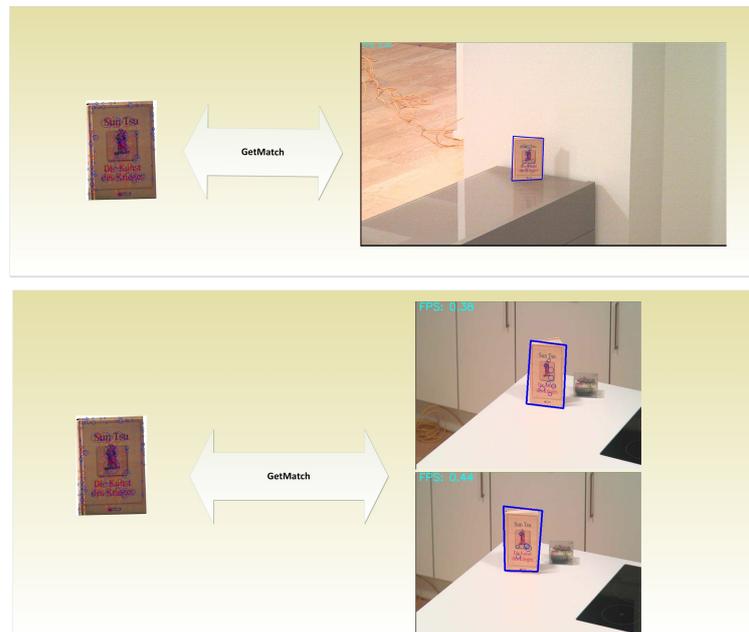


Abbildung 5.5.: Wiederfinden des Buches anhand des Referenzmodells im Living Place Hamburg

## 5.4. Sprachliche Beschreibung der Pose

Die Komponente „Lokation“ aus der Abbildung 4.1 stellt die Schnittstelle zum 3D-Modell des Living Place Hamburg dar, das von Bastian Karstaedt in Rahmen seines Masterprojektes entwickelt wurde (Vgl. [Karstaedt, 2011a,b](#)). Aufbauend auf dem 3D-Modell ist der Indoor Spatial Information Service (ISIS) implementiert, der Anfragen über räumliche Relationen von Objekten erlaubt.

Das dreidimensionale auf Basis der IFC-Klassen generierte Modell des Living Place Hamburg erfasst alle im Living Place Hamburg vorhandenen Möbel und deren Positionen (vgl. [Karstaedt, 2011a](#)).

Die Antwortnachricht von ISIS wird mit Hilfe von vordefinierten globalen Bezeichnungen, den sogenannten Objekt-IDs, für die erfassten Möbel im 3D-Modell des Living Place Hamburg generiert. Dadurch können Teilnehmer, wenn sie eine Serviceanfrage an das 3D-Modell des Living Place Hamburg stellen, das Möbelstück eindeutig benennen. Die bislang definierten IDs im Living Place Hamburg sind in der Abbildung 5.7 dargestellt.

Für Serviceanfragen an ISIS wurde der Zugriff auf den ActiveMQ als eine Client-Server-Applikation in C# implementiert, welche alle Anfragen (Requests) und Antworten (Responses) an und von ISIS über einen ActiveMQ-Push-Pull-Client abgewickelt. Die Interaktion zwi-



Abbildung 5.6.: Möbel im Living Place Hamburg [Quelle: Karstaedt (2012)]

Living Place Objekt IDs

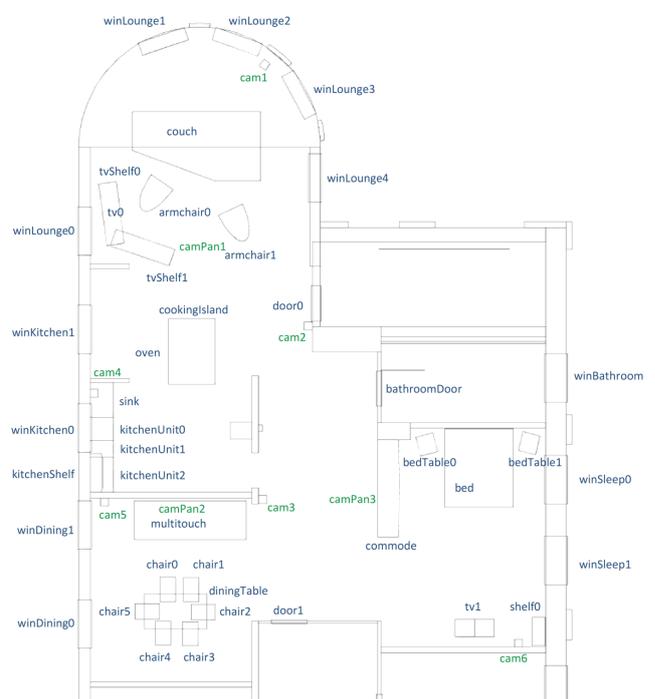


Abbildung 5.7.: Living Place Hamburg Objekt IDs [Quelle: Karstaedt (2012)]

schen dem System bzw. der Lokation-Komponente und der ISIS wird in der Abbildung 5.8 dargestellt und im folgenden genauer erläutert.

ISIS bietet für die Kommunikation mit dem Kamerabasierte Objektsuche System zwei Schnittstellen an: eine ActiveMQ-Queue namens *LP3D.REQUEST.SPATIALRELATIONSHIPS*, um Anfragen entgegenzunehmen, und ein ActiveMQ-Topic namens *LP3D.RESPONSE.SPATIALRELATIONSHIPS*, um die angeforderten Informationen zu einer Raumkoordinate zu publizieren.

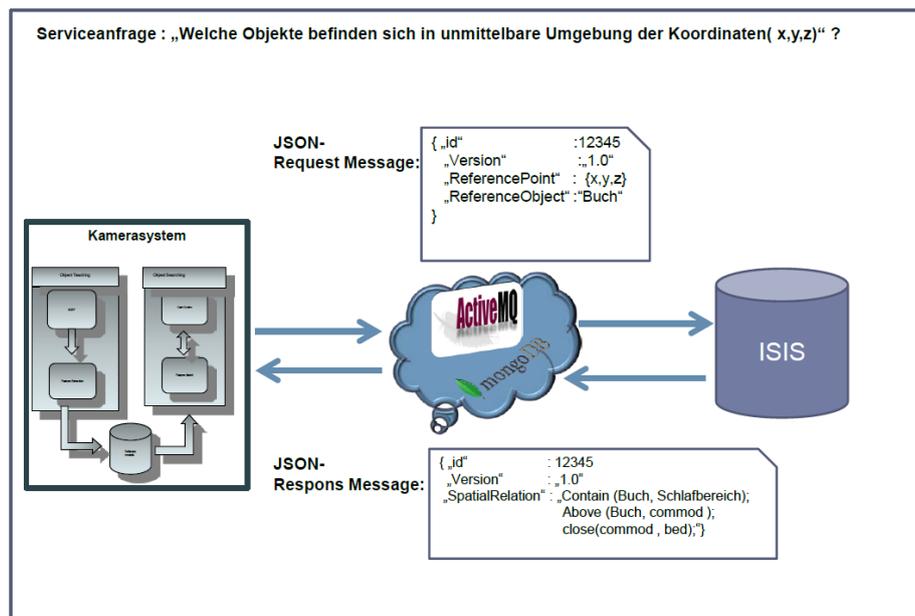


Abbildung 5.8.: Kommunikation der Kamerasystem mit ISIS

Eine ActiveMQ-Message im Living Place Hamburg ist definiert als eine Textnachricht im JSON-Format (Vgl. [Crockford, 2006](#)), die mindestens eine *Versionsnummer* und eine eindeutige *ID* der Applikation enthält. Für die Umsetzung des Szenarios in der Abbildung 4.11 wurde die ActiveMQ-Message um ein weiteres Feld namens „*ReferencPoint*“ erweitert, das die 3D-Koordinaten des gesuchten Objektes enthält.

Die angeforderten räumlichen Informationen und die Relationen werden durch ISIS auf dem ActiveMQ-Topic *LP3D.RESPONSE.SPATIALRELATIONSHIPS* publiziert und von hier an die Location-Komponente weitergeleitet, welches mit dem ActiveMQ-Topic verbunden ist.

Als Beispiel ist in dem nachfolgenden Listing die JSON-Nachricht dargestellt, die nach der Service-Anfrage "Welche Objekte befinden sich in unmittelbarer Umgebung der Koordina-ten(x,y,z)?" an den ISIS gesendet wird:

```
{ "id":1235,
  "Version":"1.0",
  "ReferencPoint":{"x,y,z"}}
```

Die Antwortnachricht ist eine JASON-Massage mit der ObjektID und Versionsnummer der Anfrage. Die angeforderten semantischen Informationen zu den angegebenen 3D-Koordinaten werden im Feld *SpatialRelations* gespeichert.

Die Informationen über die Relationen der gefundenen Objekte werden auf Basis definierter Prädikate, die beispielhaft in dem folgenden Listing dargestellt sind, aufgebaut.

```
{ "id":11235,
  "Version":"1.0",
  "SpatialRelations":
  {Contains(Room1;ref _Point),
   Above(ref _Point;commode),
   Distance(commode;bed) = "CLOSE"}
}
```

### 5.4.1. Vereinbarte Konventionen

Die räumlichen Relationen sind topologische, relationale und direktionale Informationen und werden auf Basis der im Projekt „Indoor Spatial Information Service“ definierten Prädikate, z. B. „above“, „below“ usw. angegeben.

Um dem Bewohner die Pose eines Objektes relativ zu den anderen Gegenständen in der Wohnung umgangssprachlich mitzuteilen, werden diese Prädikate von den Lokation-Module in lokalisierende Ausdrücke wie „im“, „auf“, „unter“ usw. übertragen. Es wurden Konventionen vereinbart, wie und wann die von ISIS gelieferten Prädikate in lokalisierende Ausdrücke übersetzt werden. Dies wird im Folgenden dargestellt.

#### Topologische Relationen

Das Prädikat „*Contains*“ spezifiziert die topologischen Informationen. Es liefert die Information, in welchem Raum sich die angefragten Koordinaten  $((x, y, z))$  (hier REF\_POINT) befinden. „CONTAINS“ wird durch Präpositionen wie „IM“ oder das Prädikat „ENTHÄLT“ interpretiert. Folglich lautet die Aussage zu folgender Information CONTAINS (ROOMID; REF\_POINT): „Der Raum enthält die Koordinate  $(x, y, z)$ .“ Die Aussage hinter folgender Information CONTAINS( REF\_POINT; ROOMID) lautet: „Die Koordinate  $(x, y, z)$  ist im Raum“.

Der Satz „Das Buch liegt auf der Kommode“ würde es dem unter Umständen Bewohner schwer machen, die räumliche Konstellation intern zu rekonstruieren. Denn er enthält keine Information darüber, in welchem Raum sich die Kommode befindet. Um eine Lösung für dieses Problem zu finden bzw. um es gar nicht erst entstehen zu lassen, wurden im Rahmen des Projektes folgende Regelungen für die Formulierung bzw. den Satzbau getroffen:

Das Kamerasystem bildet aus den gelieferten topologischen Relationen einen Satzanfang, der unabhängig vom Standpunkt des Betrachters und der Kameraperspektive ist. Die gewählte Formulierung des Satzes geht immer vom Prädikat „*Contains*“ aus, das als Lokalisationsäußerung „IM“ interpretiert wird.

Somit wird jeder Antwortsatz wie folgt anfangen, „Das gesuchte Objekt BEFINDET SICH IM ...“ bzw. „DAS ‚BUCH‘ BEFINDET SICH IM Schlafbereich.“ Darauf aufbauend werden mit Hilfe der gelieferten direktionalen Relationen weitere Objekte als Bezugspunkte genommen, welche dann die weitere Struktur des Satzes bestimmen.

### Direktionale Relationen

Das Prädikat „*Above*“ wird von dem Kamerasystem übersetzt in das Prädikat „AUF“. Die Aussage der gelieferten Information ABOVE (COMMODE;REF\_POINT) lautet also: „Die Koordinate (x,y,z) befindet sich (ist) ‚auf‘der Kommode“.

Das Kamerasystem interpretiert anhand der Objekt-ID COMMODE als die Kommode und REF\_POINT entspricht den 3D-Koordinaten des gesuchten Objektes, im Beispiel ist es das „Buch“.

Der von dem Kamerasystem formulierte Satz würde mit Hilfe der Informationen über die direktionalen Relationen wie folgt anfangen: „AUF“ bzw. hier im Beispiel „AUF DER KOMMODE BEFINDET SICH DAS BUCH.“

Das Prädikat BELOW wird von dem Kamerasystem übersetzt in das Prädikat „unter“.

Die Aussage hinter der Information BELOW (COMMODE;REF\_POINT) lautet: „Die Koordinate (x,y,z) befindet sich (ist) ‚unter‘“ der COMMODE. Der vom Kamerasystem formulierte Satz würde lauten: „UNTER DER KOMMODE.“

Im Rahmen dieses Projektes werden für den Aufbau einer Satzstruktur zunächst nur diese beiden Prädikate „*above*“ und „*below*“ betrachten.

Aufbauend auf dem ersten definierten Satz DAS GESUCHTE „BUCH“ BEFINDET SICH IM SCHLAFBEREICH würden mit den gelieferten direktionalen Relationen folgende Sätze formuliert werden können:

Bei geliefertem direktionalen Relationen ABOVE(COMMODE;REF\_POINT) würde der Satz mit dem Prädikat „AUF“ ergänzt werden, z. B. DAS GESUCHTE OBJEKT, „BUCH“ BEFINDET SICH IM SCHLAFBEREICH „AUF DER KOMMODE.“

Das Kamerasystem ordnet jedem Referenzobjekt anhand von dessen Form und Größe nur bestimmte Prädikate zu. Dadurch ist es möglich, einen Satz zu formulieren, der den Bewohner einen Mehrwert an Informationen liefert und sinnvoll ist.

Bei direktionalen Relationen BELOW(COMMODE;REF\_POINT) würde von dem Kamerasystem keine Übersetzung stattfinden, da für das Referenzobjekt „Kommode“ keine Zuordnung mit den Prädikat „UNTER“ besteht. Denn die Aussage DAS GESUCHTE OBJEKT, „BUCH“ BEFINDET SICH IM SCHLAFBEREICH UNTER DER KOMMODE wärd für den Bewohner sinnlos.

### Metrische Relationen

Die metrischen Informationen werden mit dem Prädikat „*Distance*“ angegeben, das Angaben über den Abstand eines Objektes zu anderen Objekten liefert. Als Bezugspunkt der metrischen Informationen wird entweder ein weiteres Objekt oder die angegebene 3D-Koordinate gesehen.

Diese Angaben werden nicht quantitativ, sondern qualitativ mit den Prädikaten „VERY\_CLOSE, CLOSE, FAR, VERY\_FAR“ ausgedrückt.

Die Aussage hinter der gelieferten Information DISTANCE(COMMODE;BED) = CLOSE lautet: Die COMMODE ist in der Nähe vom BED. Sie wird dann vom Kamerasystem interpretiert als: „Die Kommode befindet sich in der Nähe vom Bett“.

Der oben formulierten Satz würde mit der gelieferten metrischen Information DISTANCE(COMMODE;BED) = CLOSE wie Folgt ergänzt werden: DAS GESUCHTE OBJEKT, „BUCH“ BEFINDET SICH IM SCHLAFBEREICH – „AUF DER KOMMODE.“ – „NEBEN DEM BETT.“

Weitere im Rahmen der Masterarbeit von Bastian definierte Prädikate, z. B „northOf“, „southOf“,oder „VERY\_CLOSE“, „FAR“, „VERY\_FAR“ wurden in diese Arbeit nicht weiter in Betracht gezogen.

Wie groß die Abstände der Objekte für die einzelnen Prädikate definiert ist, ist in der Masterarbeit [Karstaedt \(2012\)](#) beschrieben.

### **Annahmen und Präzision bei der Sprachausgabe**

Bislang wurde davon ausgegangen, dass jede von ISIS gelieferte räumliche Relation ein sinnvolles Ergebnis liefert. Was passiert, wenn ISIS eine räumliche Information mit folgenden Aussagen an das kameragestützte System übermittelt?

*Die angegebene Koordinate liegt zwischen einem Stuhl und dem Bett.*

*Die angefragte Koordinate liegt unter der Kommode.*

*Die angegebene Koordinate befindet sich einen halben Meter über der Kommode.*

Um einen kurzen Satz zu formulieren, der die Lage des Objektes im Raum eindeutig wiedergibt, muss sich das kamerabasierte Objektsuch-System auf Grundlage der gewonnenen räumlichen Informationen zwischen verschiedenen Perspektiven, Bezugssystemen und Nachbarobjekten entscheiden.

Hierfür wurden Regeln getroffen und im Gesamtsystem implementiert.

Bei der Wahl des Nachbarobjektes wurde für die sprachliche Ausgabe in dem Lokation-Modul definiert, welche Gegenstände in der Wohnung als prominente und welche als sekundäre Objekte betrachtet werden. Diese Kategorisierung basiert auf einer Analyse der Form und Größe eines jeden Gegenstandes in der Wohnung.

Ein prominentes Objekt wird definiert als ein Gegenstand in der Wohnung, der eindeutig einem bestimmten Bereich in der Wohnung zugeordnet ist (z. B. das Bett dem Schlafbereich, das Sofa dem Wohnbereich, der Herd der Küche usw). Somit stellt ein prominentes Objekt in der Wohnung eine „Landmarke“ für einen Bereich der Wohnung dar.

Eine sichere Annahme, von der ausgegangen werden kann, ist, dass ein Buch nicht in der Luft schweben kann, da es sich während der Suche nach Voraussetzung in Ruhe befindet.

Eine weitere Annahme, die implementiert wurde, ist, dass bestimmte Gegenstände aufgrund ihrer Form, Größe und Position im Living Place Hamburg keine weiteren Gegenstände unter sich verbergen können. So kann sich z. B. ein Buch nicht unter einer Kommode befinden.

Somit wird die Auswertung der von ISIS gelieferten räumlichen Relationen mit dem Prädikat „below“ in Bezug zu bestimmten Gegenständen (z. B. Kommode, Küchentresen, Sofa) ignoriert bzw. explizit unterbunden.

Neben diesen Annahmen müssen noch einige weitere Annahmen getroffen und in der Implementierung berücksichtigt werden, um einen Satz zu formulieren, der für den Bewohner verständlich ist.

## Ergebnisse

Die Experimente im Living Place Hamburg mit dem 3D-Modell haben gezeigt, dass die gelieferten topologischen (5.4.1) und direktionalen Relationen (5.4.1) ausreichen, um einen beschreibenden Satz wie den folgenden zu formulieren: „Das Buch ist im Schlafbereich auf der Kommode.“

Dieser Satz könnte optional um metrische Informationen (5.4.1) ergänzt werden. Sie werden mit Prädikaten wie „near by“ angegeben und liefern Angaben über den Abstand eines Objektes zu anderen Objekten.

Als Bezugspunkt der metrischen Informationen wird entweder ein weiteres Objekt gewählt (hier das Bett); hierbei würde der Satz wie folgt ergänzt werden: „Das Buch ist auf der Kommode. Die Kommode ist in der Nähe des Bettes“; oder die angegebene Koordinate.

## 5.5. Offene Realisierungen/Themen

Die folgenden Fragestellungen aus dem Bereich „Orten von Objekten in Bilder und Raum“ (4.2 bzw. 4.3) konnten noch nicht erledigt werden:

1. Automatische Ermittlung der 3D-Koordinaten
2. Implementierung einer Planungs- und Koordinationseinheit zum automatischen Durchsuchen der Wohnung und damit verbunden die Steuerung der Kameras.

Hierfür wurden bereits theoretische Lösungsansätze im Rahmen der Masterarbeit ermittelt. Es wurden auch einzelne Teilbereiche implementiert und Experimente durchgeführt.

Um von diesen Experimenten zur Lösung in einer stabilen Anwendung zu gelangen, der auch die Anforderung der oben genannten Themen abdecke, sind aber noch einige Hürden zu überwinden.

Die größte Hürde wird die Steuerung der Kameras im Living Place Hamburg sein. Im Rahmen dieser Masterarbeit wurde eine einfache Kamerasteuerung mit dem Dallmeier Camera Communication Protocol (DCCP) realisiert. Sie integriert die Funktionalitäten zur Steuerung der eingebauten Kameras. Wie in der Abbildung 5.9 dargestellt ist, wurde für die Steuerung eine erste Testanwendung implementiert, die es ermöglichte, Steuerbefehle über das Netzwerk an die einzelnen Kameras zu senden.

Das DCC-Protokoll ermöglicht es, jeden Steuerbefehl (rechts, oben, unten, links) über das Netzwerk an die Kamera zu senden. Jedoch muss jeder Steuerbefehl mit einem Stop-Befehl

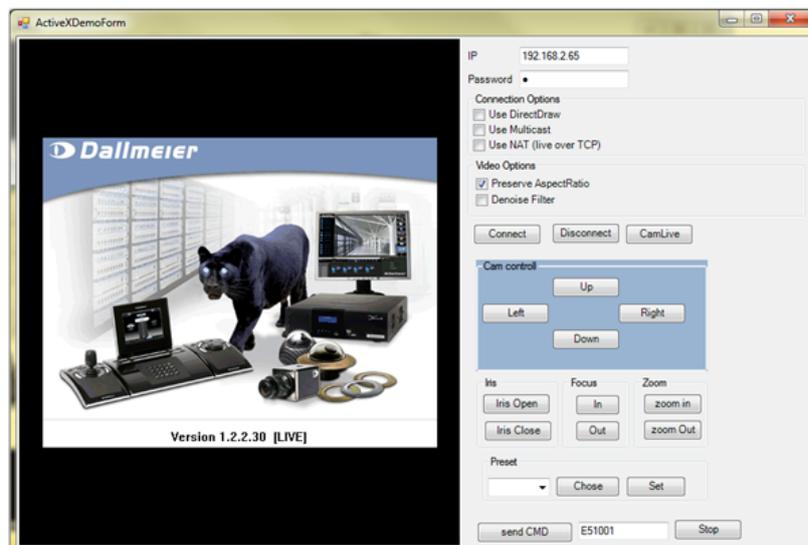


Abbildung 5.9.: Applikation zur Steuerung der PTZ-Kamera [Quelle:Hosnia Najem]

abgeschlossen werden, um die Aktion zu beenden. Auch kann der aktuelle Status der Kameras (Zoomeinstellung, Drehung, genaue Position der Kameras) nicht über das Protokoll abgefragt werden.

Somit ist dieses Protokoll als Schnittstelle zu den Kameras zum Durchsuchen der Wohnung sowie als Planungs- und Koordinationseinheiten der Kameras ungeeignet.

### 5.5.1. Lokalisierung des Objekts im Raum

Für die sprachliche Beschreibung eines Objektes im Raum und somit auch für die Kommunikation mit dem 3D-Gebäudemodell ist die Umrechnung der 2D-Bildkoordinaten in die 3D-Raumkoordinaten fundamental. Denn das Search Module [5.3.3] liefert die Position des gefundenen Objektes in Bildkoordinaten. Um daraus die Position des Objektes im Raum zu bestimmen, müssen diese in die dazugehörigen 3D-Raumkoordinaten umgerechnet werden.

Bislang wurden diese 3D-Koordinaten manuell an Hand einer Kamera über den „Ubisene Simulator“ ermittelt. In Zukunft müssen aber für die Serviceanfragen an den ISIS die zugehörigen Raumkoordinaten automatisch errechnet werden.

Auch bietet die integrierte PView-Software<sup>5</sup> die Möglichkeit, die Kameras manuell entwe-

<sup>5</sup>PView ist das zentrale Steuerelement der Dallmeier Video Management Systeme, das dem Anwender Steuerfunktionalitäten der Kameras anbietet. Vgl. [PView-Software von Dallmeier](#).

der über die PView-Oberfläche (vgl. Abbildung A.8) oder den angeschlossenen Joy-Stick zu steuern.

Für die Bestimmung des Raumpunktes zu einem Objekt werden also keine „magic“-Funktionen benötigt, worunter Lösungs-Methoden und -Verfahren verstanden werden, die noch nicht erforscht und entwickelt sind.

Einige der mathematische Methoden und Verfahren zur Ermittlung der Position des Objektes im Raum sind im Abschnitt „Positionbestimmung im Raum“ 2.3 aufgeführt.

Die Position des Objektes im Raum kann anhand einfache geometrischer Beziehungen von Punkten zueinander berechnet werden. Hierfür sind in OpenCV bereits viele Verfahren der Photogrammetrie und des stereoskopischen Sehens (Stereopsis) integriert. Diese können gut in dem Prototypen eingebunden werden.

Weiterhin sind im kameragesteuerten Objektsuch-System viele weitere Sensor-Techniken integriert, die als Hilfestellung mit herangezogen werden können, um die Bestimmung der 3D-Raumkoordinaten eines Objektes einfacher zu gestalten. Ein Ansatz hierzu ist der Einsatz einer Kinect-Kamera,<sup>6</sup> (*Kinect, Windows SDK, 2011*), mit deren Hilfe die Tiefeninformation eines Objektes bestimmt wird. Die Integration solcher Kameras stellt für dieses System keine große Hürde dar.

### 5.5.2. Die Kamerasteuerung

Diese Suche von Objekten wurde bislang mit nur einer Kamera und mit manueller Steuerung vorgenommen.

Die Integration der PTZ-Kameras im Living Place Hamburg stellt den Entwickler vor eine zusätzliche Herausforderung. Denn bevor die Raumpunkte über die oben genannten Verfahren ermittelt werden können, muss die Voraussetzung geschaffen werden, die Kameras in jeder beliebigen Position zu positionieren.

Für die Steuerung der PTZ-Kameras soll eine Anwendung erstellt werden, die über eine Schnittstelle die Steuerungsfunktionen Zoomen, Schwenken und Lenken automatisch einstellt und steuert.

Laut Hersteller unterstützen die Kameras das standardisierte Protokoll *Open Network Video Interface* (ONVIF) für den Zugriff auf eine IP-Kamera über eine SOAP-Schnittstelle.<sup>7</sup>

---

<sup>6</sup> Xbox: <http://www.xbox.com/de-DE/kinect/>

<sup>7</sup><http://www.dallmeier-electronic.com/de/cctv-ip-videoueberwachung/integration/onvif.html>

Die ONVIF-Schnittstelle würde unter anderem ermöglichen, den Status der Kameras jederzeit abzufragen und jede einzelne Kamera in variabler Geschwindigkeit in jede Position zu schwenken, zu zoomen und zu lenken.

Für die technische Umsetzung stellten sich jedoch im Rahmen dieser Masterarbeit viele Hürden in den Weg, die überwunden werden mussten.

Diese Masterarbeit beschäftigt sich zum erstmals mit den Kamera im Rahmen der Living Place Hamburg . Es musste erst in Erfahrung gebracht werden, welche Funktionalitäten, welche Schnittstellen diese Kameras integrieren.

Nach dem Sichten der Dokumentation von Dallmeier und ONVIF wurde ein Projekt erstellt, in dem die ONVIF-Bibliotheken eingebunden wurden. Im zeitlichen Rahmen der Masterarbeit ist es der Verfasserin nicht gelungen, herauszufinden, wie die Steuerbefehle durch die Einbindung in das Open Network Video Interface (ONVIF) an die Kameras versendet werden könnten. finden ob die Schnittstelle wirklich von der Kamera-Software unterstützte wird.

Zwar führte die Software-Dokumentation der Firma Dallmeier auf, dass diese Schnittstelle von den Kameras unterstützt werden. Daraufhin wurde versucht, diese Schnittstelle über die Anbindung der Bibliotheken in ein C#-Projekt anzusprechen, doch leider ohne Erfolg.

Mit einer Umsetzung dieser Funktionalitäten wäre für die Integration der Kameras in der Smart-Home-Umgebung der Grundstein gelegt. Somit würde der Interaktion der Kameras in weitere Projekte im Rahmen des Living Place Hamburg nichts mehr im Wege stehen.

## 5.6. Fazit

In diesem Kapitel wurde gezeigt, wie das Konzept in einem Prototypen implementiert wurde und welche Teile des Konzepts realisiert werden konnten. Die Abbildung 5.10 stellt die Systemarchitektur des Gesamtsystems dar.

Die Abbildung verdeutlicht, dass für alle vorgestellten Komponenten ein Modul des Prototypen implementiert wurde, der die vorgestellten und dazugehörigen Anforderungen erfüllt.

Für das Szenario „Objekt im System hinterlegen“ 3.2.2 werden im Abschnitt 5.3 die Implementation „TeachObject“ sowie die Benutzeroberfläche „Teach-GUI“ vorgestellt.

Für das Szenarien „Objekt suchen“ 3.2.1 wurden die Anwendungs-Komponente „SearchObject“ und die Benutzeroberfläche „Seach-GUI“ implementiert 5.3.3.

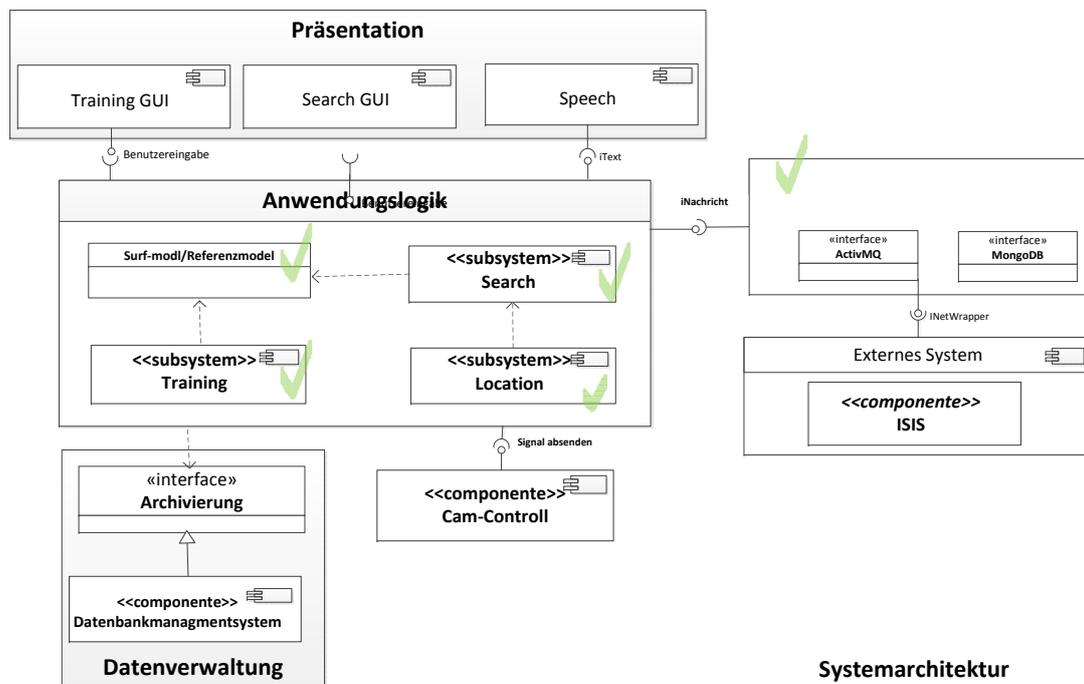


Abbildung 5.10.: Systemarchitektur in Gesamtüberblick

Für diese Umsetzungen wurde in dem Abschnitt „Modellierung und Suche von Objekten“ durch Experimente ermittelt, wie die Referenzmodelle im Kamerasystem hinterlegt werden müssen, um diese ohne großen Aufwand für die Kamerasteuerung in den Bildsequenzen

der Kameras im Living Place Hamburg wiederzuerkennen und in den Bildern zu lokalisieren. Gemäß dem Konzept im Abschnitt „Orten von Objekte im Raum“ 4.3. ist das Location-Modul für die Umsetzung der beschriebenen Anwendungsfälle und Anforderungen zuständig.

Die Einbindung des „ISIS“ als einem Serviceanbieter für die Anfragen über räumliche Informationen und die Relationen zwischen den Objekten ermöglichte die sprachliche Beschreibung der Pose im Raum.

In dem Abschnitt 5.4 wurde gezeigt, wie die gelieferten Prädikate in die lokalisierenden Ausdrücke „im“, „auf“ und „unter“ übersetzt werden, damit das kamerabasierte Objektsuch-Systems daraus einen Satz wie „Das Buch befindet sich im Schlafbereich auf der Kommode“ strukturieren kann. In dem Abschnitt 5.4.1 wurden die verwendeten Annahmen dargelegt.

In der Abbildung 5.10 befinden sich zwei Komponenten, für die kein Prototyp realisiert werden konnte. Die Gründe dafür und auch, welche weiteren Schritte noch folgen sollen, sind in dem folgenden Abschnitt „Zukünftiges Vorhaben/Offene Realisierungen“ 5.5 aufgeführt.

Für die Realisierung des Projektes sind die Grundsteine in der Praxis gelegt wurden und die Stärken und Schwächen der Verfahren erprobt worden.

# 6. Schlussbetrachtung

## 6.1. Zusammenfassung

Dieser Abschnitt der Schlussbetrachtung stellt eine Zusammenfassung dieser Masterarbeit „Kamerabasierte Suche von Objekten in einer Smart Home Umgebung“ dar. Was waren die Ziele dieser Arbeit?

Sie wurde im Kontext des „Smart Home“-Projektes Living Place Hamburg an der HAW Hamburg durchgeführt. Aus der Vision dieser Arbeit: „dem Raum ein Auge zu verschaffen“ wurde das Ziel der Entwicklung eines Konzepts und eines Anwendungsprototypen zur Unterstützung des Bewohners bei der Suche nach bestimmten Objekten im Living Place Hamburg.

Diese Zielsetzung impliziert die Entwicklung eines smarten kameragestützten Systems, bei der die Kamera autonom die Wohnung nach beliebigen Objekten anhand von zuvor erstellten Referenzmodellen durchsucht, die Lage des Objektes im Raum an Hand der Kamerabilder erfasst und dessen Lage im Raum den Bewohner sprachlich mitteilt.

Für den Bewohner bedeutet dies, dass er das Kamerabasierte Objektsuche System fragen kann: „Wo ist mein Buch?“ und damit eine Suche auslöst. Wenn sie erfolgreich ist, so erhält er eine Mitteilung, wie z. B. „Das Buch liegt auf der Kommode im Schlafzimmer.“

Es stellten sich die folgenden Fragen für die Konzeption dieses Services: Wie kann die Serviceanfrage vom System verstanden werden? (Was soll gesucht werden?) – Wie kann die Pose (die Lage und Position) eines Objektes an Hand der Referenzmodelle in den Bildern und im Raum ermittelt werden? (Wo ist das Objekt in Bild und Raum?) – Und wie kann diese sprachlich beschrieben werden? (Also die Frage: Wie sage ich es dem Bewohner?)

Diese Masterarbeit integriert in sich mehrere Schwerpunkte. Jedes Teilgebiet wurde unter bestimmten Forschungsfragestellungen untersucht. Für jeden Schwerpunkt wurden zur Bewältigung der Anforderungen bereits eingesetzte Verfahren verwendet. Diese stellten im einzelnen an sich keine große Hürde dar. Die Herausforderung dieser Masterarbeit war die Vielfalt der Schwerpunkte und die Notwendigkeit, sie zu einem Projekt zu vereinen.

Ein Schwerpunkt der Masterarbeit war die Entwicklung eines Konzepts zur automatischen Suche in der Wohnung nach den zuvor erstellten Referenzmodellen und die anschließende

Verortung des Objekts im Wohnraum mit Hilfe der integrierten Kameras im Living Place Hamburg .

Dann beschäftigte sich diese Masterarbeit unter anderem mit der Frage der Bestimmung der aktuellen Raumpositionen und der Erstellung einer automatischen Planungs- und Koordinationseinheit zum Durchsuchen des Wohnraumes. Ein weiterer Schwerpunkt war schließlich die sprachliche Beschreibung der Lage des Objektes im Raum.

In dem Kapitel „Analyse“ 3.4 werden die Ideen und daraus entwickelten Anforderungen an das Objektsuch-System vorgestellt. Aus Szenarien wurden Anwendungsfälle entwickelt und daraus die Anforderungen, die dann die Grundlage für das Konzept eines Prototypen wurde, das im Kapitel 4 geschildert ist.

Es wurden die ersten Schritte der Implementierung und Anwendung des Systems in der Praxis unternommen. Die Ergebnisse der bisherigen Arbeiten sind im Kapitel 5 „Umsetzung“ dargestellt.

Diese Masterarbeit realisierte als erste im Rahmen des Living Place Hamburg eine Anwendung, in der optische Sensoren verwendet werden. Mit der Erstellung dieser Anwendung auf Basis der integrierten Kameras wurden viele Hürden für die Entwicklung von weiteren Services in diesem Rahmen gemeistert. Es wurde gezeigt, wie eine Anwendung/ein Service dort optische Sensoren integrieren kann.

Welchen Mehrwert solch ein Service dem Benutzer bieten kann, wird sich in Zukunft zeigen.

## 6.2. Ausblick

Mit der Umsetzung dieser Masterarbeit wurde der Living Place Hamburg um einen weiteren Service für den Bewohner erweitert. Er wird in seinem Alltag durch diese integrierten Services in der Wohnung unterstützt.

Der Living Place Hamburg stellt ein „Smart Home“ dar. Es sind viele Companion-Systeme integriert, deren Komplexität und auch nur Anwesenheit dem Bewohner verborgen bleibt. Der Bewohner wird überwacht und sein Verhalten analysiert, um Gewohnheiten zu extrahieren. Auf dieser Basis wird die Wohnung an die Bedürfnisse des Bewohners angepasst.

Mit den optischen Sensoren wird dem Raum ein Auge verliehen, um so die Gewohnheiten des Bewohner noch besser analysieren und die Umgebung entsprechend anpassen zu können.

Wie die Entwicklung des durch diese Masterarbeit implementierten Services, aber auch die der bereits integrierten Services im Living Place Hamburg weiter verläuft, wird sich in Zukunft zeigen.

Es muss geklärt werden, wie viel Verantwortung einem System überlassen werden darf und wie viel Verantwortung dem Bewohner zugesprochen werden muss. Hierbei wird sich in der Zukunft zeigen, wie eine Balance zwischen Benutzer und System aussieht, so dass der Bewohner sich durch die ihn unterstützenden Systeme weder belästigt oder entmündigt fühlt.

Schaut man weiter in die Zukunft, so lässt sich aus der technischer Sicht die Entwicklung so abschätzen: ihr sind (zunächst) keine Grenzen gesetzt. Was aus ökonomischer Sicht noch sinnvoll ist, ist schon weitaus schwieriger zu beantworten. Denn die ökonomischen Auswirkungen und Folgen der Allgegenwärtigkeit des Computers im Alltag sind schwer absehbar.<sup>1</sup>

Mark Weisers Vision der Zukunft ist, dass die Computer aus der Umgebung verschwinden und der Mensch sie nicht mehr wahrnimmt. Es herrsche totale Vernetzung und totale Überwachung. Bereits heute bemerken die Menschen die Kameras nicht mehr, wenn sie in die U-Bahn einsteigen. Die Überwachung hat sich in den Alltag integriert und wird nicht mehr thematisiert. Es ist schon fast verpönt, die neuesten Technologien infrage zu stellen.

Die Wissenschaften sind auf der Suche nach der Antwort auf diese Fragen, was für die Auswirkungen diese totale Vernetzung und ständige Überwachung, also die Allgegenwärtigkeit des Computers, für den Menschen haben wird. Sicher ist bereits, dass die totale Vernetzung nicht mehr aufzuhalten ist, aber welches die Auswirkungen sein werden und ob die Menschen mit den (nicht nur) ökonomischen Folgen umgehen können, wird sich in Zukunft herausstellen.

---

<sup>1</sup>Quellen: [Mattern (2003a), Mattern (2003b), Bohn u. a. (2003)]

# Literaturverzeichnis

- [Barkowsky u. a. 2008] BARKOWSKY, Thomas (Hrsg.) ; KNAUFF, Markus (Hrsg.) ; LIGOZAT, Gérard (Hrsg.) ; MONTELLO, Daniel R. (Hrsg.): *Lecture Notes in Computer Science*. Bd. 4387: *Spatial Cognition V: Reasoning, Action, Interaction, International Conference Spatial Cognition 2006, Bremen, Germany, September 24-28, 2006, Revised Selected Papers*. Springer, 2008. – ISBN 978-3-540-75665-1
- [Bay u.a. 2006] BAY, Herbert ; ESS, Andreas ; TUYTELAARS, Tinne ; VAN GOOL, Luc: Speeded-Up Robust Features (SURF). In: *Computer Vision ECCV 2006* 3951 (2006), S. 404–417. – URL <http://www.vision.ee.ethz.ch/~surf/papers.html>. – ISSN 1077-3142
- [Bohn u. a. 2003] BOHN, Jürgen ; COROAMA, Vlad ; LANGHEINRICH, Marc ; MATTERN, Friedemann ; ROHS, Michael: Allgegenwart und Verschwinden des Computers – Leben in einer Welt smarterer Alltagsdinge. In: GRÖTKER, Ralf (Hrsg.): *Privat! Kontrollierte Freiheit in einer vernetzten Welt*. Heise-Verlag, 2003, S. 195–245. – ISBN 3-936931-01-1
- [Brauer 2010] BRAUER, Henrik: Entwicklung eines Augmented Reality Frameworks auf Basis von Kamera-basierten Trackingverfahren. In: *Studiengang Informatik (Master of Science) am Department Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg* (2010). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/brauer.pdf>
- [Crockford 2006] CROCKFORD, D.: *The application/json Media Type for JavaScript Object Notation (JSON)*. RFC 4627 (Informational). Juli 2006. – URL <http://www.ietf.org/rfc/rfc4627.txt>
- [Dallmeier 2013] DALLMEIER: DDZ4020-YY/HS/HD DOMERA High-Speed PTZ-Dome-Netzwerkkamera, Full HD, 1080p/30, 20 facher optischer Zoom, Autofokus, Wide Dynamic Range, Tag/Nacht, PoE+. In: *Dallmeier electronic GmbH & Co.KG* (2013). – URL <http://www.dallmeier-electronic.com/de/cctv-ip-videoueberwachung/kameras/ptz-dome-kameras/hd-ip/ddz4020-yyhshd-domerar.html>

- [Dreschke 2011] DRESCHKE, Oliver: Entwicklung kontextsensitiver Möbel für intelligente Wohnumgebungen. In: *Masterarbeit eingereicht im Rahmen der Masterprüfung im Studiengang Informatik am Department Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg* (2011), Juli
- [Ellenberg 2011] ELLENBERG, Jens: Ontologiebasierte Aktivitätserkennung im Smart Home Kontext. In: *Master Thesis eingereicht im Studiengang Informatik Master am Department Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg* (2011). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/ellenberg.pdf>
- [Freksa u. a. 2003] FREKSA, Christian ; BRAUER, Wilfried ; HABEL, Christopher ; WENDER, Karl F.: *Lecture Notes in Computer Science*. Bd. 2685: *Spatial Cognition III, Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning*. Springer, 2003. – URL <http://books.google.de/books?id=aglJtr-nhjUC&printsec=frontcover>. – ISBN 3-540-40430-9
- [Freksa u. a. 1998] FREKSA, Christian (Hrsg.) ; HABEL, Christopher (Hrsg.) ; WENDER, Karl F. (Hrsg.): *Lecture Notes in Computer Science*. Bd. 1404: *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*. Springer, 1998. – URL [http://books.google.de/books?id=HR7bCXG5EwAC&printsec=frontcover&dq=Presson+und+Montello++Points+of+reference+in+spatial+cognition:+Stalking+the+elusive+landmark.&source=gbs\\_similarbooks\\_s&cad=1#v=onepage&q&f=false](http://books.google.de/books?id=HR7bCXG5EwAC&printsec=frontcover&dq=Presson+und+Montello++Points+of+reference+in+spatial+cognition:+Stalking+the+elusive+landmark.&source=gbs_similarbooks_s&cad=1#v=onepage&q&f=false). – ISBN 3-540-64603-5
- [Freksa u. a. 2004] FREKSA, Christian (Hrsg.) ; KNAUFF, Markus (Hrsg.) ; KRIEGBRÜCKNER, Bernd (Hrsg.) ; NEBEL, Bernhard (Hrsg.) ; BARKOWSKY, Thomas (Hrsg.): *Lecture Notes in Computer Science*. Bd. 3343: *Spatial Cognition IV: Reasoning, Action, Interaction, International Conference Spatial Cognition 2004, Frauenchiemsee, Germany, October 11-13, 2004, Revised Selected Papers*. Springer, 2004. – ISBN 3-540-25048-4
- [Freksa u. a. 2008] FREKSA, Christian (Hrsg.) ; NEWCOMBE, Nora S. (Hrsg.) ; GÄRDENFORS, Peter (Hrsg.) ; WÖLFL, Stefan (Hrsg.): *Lecture Notes in Computer Science*. Bd. 5248: *Spatial Cognition VI. Learning, Reasoning, and Talking about Space, International Conference Spatial Cognition 2008, Freiburg, Germany, September 15-19, 2008. Proceedings*. Springer, 2008. – ISBN 978-3-540-87600-7
- [Fries 2011] FRIES, Carsten: Kamerabasierte Identifizierung und Lokalisierung von Gegenständen für flexible Roboter. In: *Masterarbeit eingereicht im Rahmen der Masterprüfung im Studiengang Informatik am Department Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg* (2011)

- [Habel 2001] HABEL, Christopher: Sprachproduktion: Informationsvermittlung durch natürlicher Sprache. In: *Vorschlag zur Einrichtung eines DFG-Schwerpunktprogrammes* (2001)
- [Habel und von Sutterheim ] HABEL, Christopher ; SUTTERHEIM, Christine von: *Räumliche Konzepte und Sprachliche Strukturen*. Max Niemeyer Verlag Tübingen 2000
- [Hardenack 2011] HARDENACK, Frank: Das intelligente Bett -Sensorbasierte Detektion von Schlafphasen. In: *Masterarbeit eingereicht im Rahmen der Masterprüfung im Studiengang Master Informatik am Department Informatik der Fakultät Technik und Informatik an der HAW-Hamburg* (2011), Oktober. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/weiss.pdf>
- [Hartley und Zisserman 2004] HARTLEY, R. I. ; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004
- [Hoffmann 2009] HOFFMANN, Robert: Implementierung des SURF-Feature-Detektors auf der GPU mit Hilfe von CUDA. In: *Universität Koblenz, Studienarbeit im Studiengang Computervisualistik* (2009). – URL [http://www.uni-koblenz.de/~cg/Studienarbeiten/SA\\_RobertHofmann.pdf](http://www.uni-koblenz.de/~cg/Studienarbeiten/SA_RobertHofmann.pdf)
- [Jiang und Bunke 1997] JIANG, Xiaoyi ; BUNKE, Horst: *Dreidimensionales Computersehen: Gewinnung und Analyse von Tiefenbildern*. Springer-Verlag, Berlin und Heidelberg, 1997
- [Karstaedt 2011a] KARSTAEDT, Bastian: PO2 - Entwicklung und Integration der Indoor Spatial Information Services in das Living Place Hamburg. In: *Berichte Masterprojekt 2* (2011). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/karstaedt.pdf>
- [Karstaedt 2011b] KARSTAEDT, Bastian: Seminarring - Entwicklung eines Indoor Spatial Information Service für IFC-basierte Gebäudemodelle. In: *Ringvorlesung Seminarvorträge* (2011). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-seminar/karstaedt/bericht.pdf>
- [Karstaedt 2012] KARSTAEDT, Bastian: Kontextinterpretation in Smart Homes auf Basis semantischer 3D Gebäudemodelle. In: *Masterarbeit eingereicht im Rahmen der Masterprüfung im Studiengang Informatik (Master) am Department Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg* (2012). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/karstaedt.pdf>

- [Kinect, Windows SDK 2011] KINECT, WINDOWS SDK: *Kinect, Windows SDK*, 2011. – URL <http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/default.aspx>
- [Lowe ] LOWE, David G.: FLANN - Fast Library for Approximate Nearest Neighbors. In: <http://www.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN>. – URL <http://scholar.google.ca/citations?user=8vs5HGYAAAAJ&hl=en>
- [Lowe 1999] LOWE, David G.: Object recognition from local scale-invariant features. In: *International Conference on Computer Vision, 1999* (1999), S. 1150–1157
- [Lowe 2004] LOWE, David G.: Distinctive image features from scale-invariant keypoints. In: *International journal of computer vision* 60 (2004), Nr. 2, S. 91–110
- [Lowe und Muja 2009] LOWE, David G. ; MUJA, Marius: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration FLANN. In: *International Conference on Computer Vision Theory and Application VISSAPP'09*, INSTICC Press, 2009, S. 331–340
- [Lucas 1981] LUCAS, Takeo: Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI). In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 81)* (1981), S. 674679
- [Mattern 2003a] MATTERN, F.: *Total vernetzt: Szenarien einer informatisierten Welt*. Springer-Verlag GmbH, 2003 (... Berliner Kolloquium der Gottlieb-Daimler-und-Karl-Benz-Stiftung). – URL <http://books.google.de/books?id=1K1q2HPL8tIC>. – ISBN 9783540002130
- [Mattern 2003b] MATTERN, Friedemann: Vom Verschwinden des Computers – Die Vision des Ubiquitous Computing. In: MATTERN, Friedemann (Hrsg.): *Total vernetzt– Szenarien einer informatisierten Welt*. Springer-Verlag, 2003, S. 1–41. – ISBN 3-540-00213-8
- [Meisel 2012] MEISEL, Andreas .: Vortragsfolien zu Grundlagen der 3D-Bildverarbeitung. Studiendepartment Informatik der Fakultät Technik und Informatik an der HAW-Hamburg, 2012. – URL [http://www.informatik.haw-hamburg.de/uploads/media/AW\\_3DBV\\_V03.pdf](http://www.informatik.haw-hamburg.de/uploads/media/AW_3DBV_V03.pdf)
- [Otto und Voskuhl 2010] OTTO, Kjell ; VOSKUHLE, Sören: Entwicklung einer Architektur für den Living Place Hamburg. In: *Master Informatik am Studiendepartment Informatik der Fakultät Technik und Informatik an der HAW-Hamburg* (2010). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/otto-voskuhl.pdf>
- [Premsky 2001a] PREMSKY, Marc: Digital Natives, Digital Immigrants. In: *On the Horizon* 9 (2001), Oktober, Nr. 5

- [Prensky 2001b] PRENSKY, Marc: Do They Really Think Differently? In: *On the Horizon* (2001), Dezember
- [R. Kurazume und Ikeuchi January 2002] R. KURAZUME, Z. Z. ; IKEUCHI, K.: Simultaneous 2D images and 3D geometric model registration for texture mapping utilizing reflectance attribute. In: *Proc. of Fifth Asian Conference on Computer Vision (ACCV)* Vol. I (January 2002), S. 99–106
- [Reinhard Klette 1996] REINHARD KLETTE, Karsten S.: *Computer Vision Räumliche Information aus digitalen Bildern*. Vieweg Technik, 1996
- [Rickheit 1999] RICKHEIT, Gert: *Richtung im Raum interdisziplinäre Perspektiven*. Deutscher Universitäts-Verlag, 1999
- [Rickheit und Strohner 1993] RICKHEIT, Gert ; STROHNER, Hans: *Grundlagen der kognitiven Sprachverarbeitung Modell, Methoden, Ergebnisse*. Francke Verlag Tübingen und Basel, 1993
- [Schuman 2008] SCHUMAN, Alewtina: *Ein einfach benutzbares mobiles Navigationssystem für Fußgänger*, Angewandte Informatik am Studiendepartment Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/schuman.pdf>
- [Tsai 1986] TSAI, R.Y.: An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL* (1986), S. 364–374
- [Vorweg 2001] VORWERG, Constanze: *Raumrelationen in Wahrnehmung und Sprache. Kategorisierungsprozesse bei der Benennung visueller Richtungsrelationen*. Deutscher Universitäts-Verlag, 2001
- [Voskuhl 2011] VOSKUHL, Sören: Modellunabhängige Kontextinterpretation in einer Smart Home Umgebung. In: *Masterarbeit eingereicht im Rahmen der Masterprüfung im Studiengang Informatik (Master) am Department Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg* (2011). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/voskuhl.pdf>
- [Want 2000] WANT, Roy: Remembering Mark Weiser: Chief Technologist, Xerox PARC. In: *IEEE Personal Communications* (2000), February. – URL [http://www.roywant.com/cv/papers/pubs/2000-02%20\(IEEE%20PCom\)%20weiser%20biography%20\(print\).pdf](http://www.roywant.com/cv/papers/pubs/2000-02%20(IEEE%20PCom)%20weiser%20biography%20(print).pdf)

- [Weiser 1991] WEISER, Mark: The computer for the 21st century. In: *Scientific American* Vol. 265 (1991), September, Nr. 3, S. 66–75. – URL <http://www.cse.nd.edu/~cpoellab/teaching/cse40463/weiser.pdf>
- [Weiss 2011] WEISS, Ann-Kathrin: Smart Textiles: Entwicklung textiler Sensoren für intelligente Umgebungen am Beispiel eines Sofas. In: *Bachelorthesis eingereicht im Rahmen der Bachelorprüfung im Studiengang Bekleidung - Technik und Management am Department Design der Fakultät Design, Medien und Information der Hochschule für Angewandte Wissenschaften Hamburg* (2011). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/weiss.pdf>
- [Witt 2011] WITT, KristofferKristoffer: Kontextabhängige multimodale Interaktion mit Schwerpunkt Spracherkennung im Smart-Home Umfeld. In: *Studiengang Master Informatik am Studiendepartment Informatik der Fakultät Technik und Informatik an der HAW-Hamburg* (2011)
- [Xiaoyi Jiang ] XIAOYI JIANG, Horst B.: *Dreidimensionales Computersehen: Gewinnung und Analyse von Tiefenbildern*

# A. Kameras im Living Place Hamburg

Dieser Abschnitt beschreibt den Aufbau der Infrastruktur im Living Place Hamburg Fokus auf die für das Projekt relevanten Eigenschaften und Funktionen der eingebauten Full High-Definition (HD) Pan/Tilt/Zoom(PTZ) High Speed (HS) Dome Kamera der Firma Dallmeier, im Folgenden PTZ-Domekamera genannt.



Abbildung A.1.: Die Kameras im Wohnraum: [Quelle: [Karstaedt \(2012\)](#)]

Im Living Place Hamburg sind sechs der Modell DDZ4010-YY/HS/HD DOMERA® in der Aufputz Variante mit integrierter IP I/O-Schnittstelle montiert.

Das Living Place Hamburg ist in die Bereiche "Entwicklung und Kontrolle" sowie den "Wohnbereich" aufgeteilt, die im 2-D-Gebäudemodell in der Abbildung 1.2 farbig markiert dargestellt sind. Der Wohnraum ist wiederum in einzelne Bereiche unterteilt, in denen jeweils eine Aufputzvariante der PTZ-Domekamera montiert ist.

Jeder Kamera ist einen Name und eine IP-Adresse zugeordnet, wie in der Abbildung [A.1](#) dargestellt.

Kamera eins (IP-CAM1) befindet sich im Wohnzimmer, Kamera zwei (IP-CAM2) und drei (IP-CAM3) sind im Durchgang zwischen den Eingängen montiert. Der Ess- bzw. Arbeitsbereich und der Schlafbereich werden von den Kameras fünf und sechs (IP-CAM5, IP-CAM 6) abdeckt.



Abbildung A.2.: Kameras Position im Schlafbereich [Quelle: [Karstaedt \(2012\)](#)]



Abbildung A.3.: Kameras vom Esszimmer [Quelle: [Karstaedt \(2012\)](#)]

Hier wird erklärt, welcher Switsh eingesetzt wird. Dass zwei Streams von den Kameras abgegriffen wird. Analoger Stream und einmal Digitaler Stream.

Die Abbildung [A.4](#) stellt die Vernetzung der Kameras mit den Switsh im Kontrollraum dar.

Der Kontrollraum ist der wichtigster Bereich im Living Place Hamburg . Über einen Doppelboden verlaufen alle Kamera Kabeln in einen HP PoE Switch ein. Die Kameras sind in einen eigenen Netzwerk angeschlossen, über die Kameras konfiguriert und gesteuert werden. Jede Kamera enthält zwei Anschlüsse, in dem zwei Arten von Video-Streams deffrenziert werden. Einen Analogen Anschluss und einen Digitalen Anschluss.

Von dem Kameras werden zu einem ein Analoger Videostream der Kameras abgegriffen zu anderem ein Digitalen Videostream.

Die analoge Video-Streams sind für die Anwendungen in Echtzeitbereich vorgesehen.Diese sind Anwendungen, die Zeitkritische Aktionen ausführen können den Anloger Videostream benutzen, da die Bilderinformation ohne Verzögerung an den Endgerät ankommen.

Der Digitaler Anschluss ist über den HP PoE Switch mit einem PC- Verbunden, der für die Steuerung der Kameras den PView Software installiert hat, der sogenannter PView-PC.

Über den angeschlossen Joystick am PView-PC lässt sich die Kamera über den PView Software <sup>1</sup>steuern.

Der Digitaler Anschluss sind für Anwendungen vorgesehen, die keine Zeitkritische Aktionen auf Basis der optische Sensordaten durchführen.

Ein Kriterium für die Auswahl der Kameramodelle war die Unterstützung verschiedener standardisierter Technologien und Protokolle, wie Real Time Streaming Protocol (RTSP), um mithilfe eines externen Videoencoders die Echtzeit-Videostreams speichern und auswerten zu können.

Der Modell DDZ4010-YY/HS/HD DOMERA® unterstützten verschiedener standardisierter Technologien und Protokolle die im Folgenden aufgeführt werden.

Im Folgenden werden die wichtigste Eigenschaften und Funktionalitäten der PTZ-Domekamera, die in den Dokumentation von Dallmeier <sup>2</sup> entnommen wurde aufgeführt.

### **A.0.1. Kamera Eigenschaften**

Die PTZ-Domekameras besitzen einen CMOS-Sensor und sind ausgestattet mit einem Zoomobjektiv mit 10-facher optischer Zoom, 12-facher digitaler Zoom.

Die Kamera verfügt über einen 1/3 Full High-Definition CMOS-Bildsensor. Die hochauflösende Sensortechnik liefert klare, kontrastreiche und farbgetreue Bilder. Die Kamera verfügt über einen hochperformanten Pan/Tilt/Zoom-Mechanismus.

---

<sup>1</sup><http://www.dallmeier.com/en/cctv-ip-video-surveillance/product-history/software/pview.html>

<sup>2</sup>Quelle: ? Homepage:<http://www.dallmeier.com/>

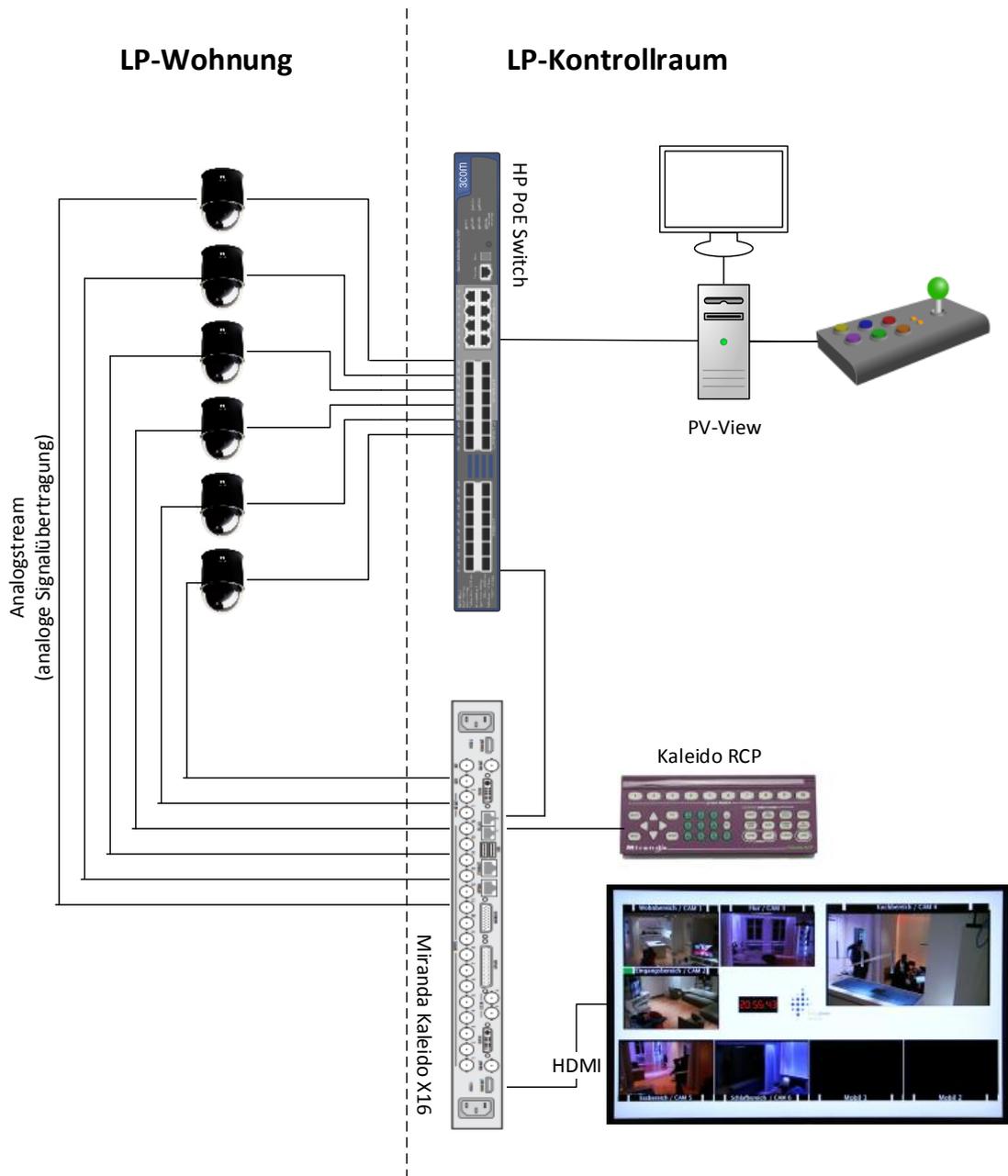


Abbildung A.4.: Vernetzung der Kameras im Living Place Hamburg

Mit der integrierten ICR-Funktion wird eine Tag/Nacht-Funktionalität der Kamera realisiert. Der Umschaltzeitpunkt und der Schwellenwert sind einstellbar. Zur Optimierung der Bildqualität verfügt die Kamera mittels zahlreicher Kontrollfunktionen wie automatischen Weißabgleich (AWB), automatischer Verstärkungsregelung (BLC) und Slow Shutter. Sie verfügt über Auto-Fokus mit manueller Korrektur sowie über digitale Rauschreduzierung. Das Lichtverhältniss wird durch die integrierte Helligkeitsregelung, Rauschreduzierung und weitere Bildverarbeitungsfunktionen und digitale Signalverarbeitungermöglicht.

Sie liefern über das Netzwerk MJPEG-, MPEG-4- und H.264-Video-/Audiokompression hoher Qualität . Sie unterstützen die Übertragung von Videostreams über das IP-Netzwerk an eine Unicast- oder eine Multicast-Adresse in den Standards 720p (1280x720 Pixel) sowie 1080i/p (1920x1080 Pixel) bei einer Bildrate von 25fps bis 60fps (in Abhängigkeit von der Auflösung), also in HD/Full HD-Qualität. Es können Bitraten bis 8 Mbits/s für Video und bis 384 kbit/s für Audio eingestellt werden.

## A.0.2. Spezifikation der Kamera

---

### Sensor

**Typ**  $\frac{1}{3}$ " Full High-Definition CMOS-Bildsensor

**Effektive Pixel** Ca. 2 Megapixel

#### Signalverarbeitung

Pure Digital Signal Processing

**Bilderfassung** Progressive Scan / Interlaced

#### Minimale Beleuchtungsstärke

ICR Off: 2 lux (F1.8, 30 IRE) ICR On: 0,2 lux (F1.8, 30 IRE)

#### Signal-Rausch-Verhältnis (SNR)

Größer als 50 dB

---

### Objektiv

**Typ** F1.8 F2.1 / f = 5,1 mm 51,0 mm (10 optischer Zoom)

---

**Blendensteuerung**

Automatisch (Autoiris)

**Horizontaler Bildwinkel**

Ca. 50,0° (Weitwinkel) bis 5,4° (Tele)

**Minimale Objektdistanz**

10 mm (Weitwinkel) bis 800 mm (Tele)

---

**Funktionen**

**Digitalzoom** 12 (120 mit optischem Zoom)

**Fokussteuerung**

Autofokus (AF-Empfindlichkeit: Normal, Low), One-Push AF, Manuell, Unendlich (8), AF-Naheinstellgrenze wählbar

**Tag/Nacht-Umschaltung**

ICR-Funktion (Umschaltschwelle einstellbar)

**Schwarz-Weiß-Modus**

Automatisch (im Nachtmodus), Manuell (über Einstellung Effekte), Elektronischer (Slow)Shutter ] 1/2 bis 1/10000 s (21 Schritte), Slow Shutter Limit wählbar

**Belichtungssteuerung**

Automatisch (AE), Manuell, Shutter Priority (Zeitvorwahl/Blendenautomatik), Iris Priority (Blendenvorwahl/Zeitautomatik), Bright

**Belichtungskorrektur**

–10,5 dB bis +10,5 dB (in 15 Schritten, in Stufen von 1,5 dB)

**Verstärkungsregelung (Gain)**

Automatisch (AGC), Manuell (-3 bis +18 dB, in 8 Schritten)

**Weißabgleich (WB)**

Auto (AWB), Indoor, Outdoor, One-Push, Manuell

**Gammakorrektur**

Normal, Cinema Type 1, Cinema Type 2

**Gegenlichtkompensation (BLC)**

On / Off

**Flip-Funktion** Ja**Privacy Zone Masking**

Maskieren von bis zu 8 geschützten Zonen (Block, Video Off, Mosaik)

**Effekte** S/W (Schwarz-Weiß), Negativ**Bewegungserkennung (VMD)**

Mit wählbarer Sensitivität (5 Schwellen)

**Alarmmeldung** Ereignisgesteuerte Meldungsfunktionen für Textbenachrichtigung, E-Mail-Versand und FTP-Bilder-Upload**Alarm-Trigger** Statusänderung Kontakt IN, Remote-Statusänderung Kontakt OUT, Videosignalverlust**Alarm-Aktionen**

PTZ-Alarm-Aktion, Statusänderung Kontakt OUT

**Konfiguration IP I/O-Schnittstelle**

Mit Webbrowser

**Konfiguration/Steuerung PTZ-Domekamera**

Über OSD-Menü (On Screen Display Menü)

**Sprachen** Deutsch, Englisch, Französisch, Spanisch, Italienisch

---

**PTZ-Funktionen****Pan (Schwenken) Tilt (Neigen)**

360° kontinuierlich / 0° 180°

**Variable Geschwindigkeit:**

Pan: 120 %/s, Tilt: 120 %/s (Turbo: 420 %/s)

**Preset Geschwindigkeit:**Pan: 420 %/s, Tilt: 420 %/s ( $\pm 0,1^\circ$ )

**Programmierbare Presets**

248 Positionen mit Kamerastatus (Titel mit bis zu 16 Zeichen)

**Programmierbare Scans**

16 Scans (inklusive Diagonal-Scans), 1 Auto Pan (360° kontinuierliches Schwenken)

**Programmierbare Rundgänge**

8 Touren (Überwachungsabläufe), 8 Pattern (benutzerdefinierte Bereichsabfahrten, bis zu 480 Sekunden)

**Alarm-Aktionen bei Kontakt in Statusänderung**

Automatisches Ausführen von gespeicherten PTZ-Funktionen (Presets, Scans oder Rundgänge), Statusänderung Kontakt OUT

---

**Format und Encodierung**

**Videostandard** SDTV (PAL/NTSC), HDTV (SMPTE 296M, SMPTE 274M)

**Auflösung SD** 4CIF (704 576), CIF (352 288), QCIF (176 144), 4SIF (704 480), SIF (352 240), QSIF (176 112), VGA (640 480), QVGA (320 240)

**Auflösung HD** 1080p (1920 1080), 1080i (1920 1080), HD H (1920 540), HD D (1440 720), 720p (1280 720)

**Videokompression**

H.264, MJPEG

**Audiokompression**

MPEG-1/2 Layer 2, G.711, G.722.1

**Bildrate** Bis zu 60 fps bei SD-Auflösung und 720p Bis zu 30 fps bei 1080p, 1080i, HD H, HD D

**Video-Bitrate** 0,25 8 Mbit/s, CBR (konstante Bitrate) oder VBR (variable Bitrate)

**Audio-Bitrate** 32 160 kbit/s (MPEG-1/2 Layer 2), 64 kbit/s (G.711), 16 kbit/s (G.722.1)

**Video-Streaming**

Simultanes Dual-Streaming: Videokompression, Auflösungen, Bild- und Bit-Raten der einzelnen Streams unabhängig einstellbar (maximale Gesamtbildrate aller Streams 30 fps mit maximaler Auflösung)

**Übertragungsformat**

Progressive (Vollbild)

**Live-Streaming Übertragungsmodus**

Unicast, Multicast

**Schnittstellen Audio IN (Stereo)**

1 3,5 mm Klinke (Line IN)

**Audio OUT (Stereo)**

1 3,5 mm Klinke

**Kontakt IN 3****Kontakt OUT 1****Serielle Schnittstelle**

1 RS485 (Vollduplex, Halbduplex)

**Lokaler Videospeicher**

USB-/SDHC-Port integriert

**Ethernet 1 RJ45, 10BASE-T-/100BASE-TX-PoE****Programmierschnittstelle**

Offene API für Integration von Drittsystemen (ONVIF-konform)

---

**Protokolle****Ethernet-Protokolle**

IPv4, TCP, UDP, ARP, ICMP, DHCP, NTP, HTTP, RTSP, IGMP V2, FTP, SMTP, RTP, RTCP

---

**Sonstiges Videopuffer**

64 MB RAM Ringspeicher (Vor- und Nachalarm, Netzausfallüberbrückung)

---

**Elektrische Merkmale****Spannungsversorgung**

PTZ-Kamera: Netzteil 1 mit 24/28 V AC  $\pm 10\%$  (50/60 Hz) oder über PoE+ Lüfter/Heizung (Wetterschutzgehäuse): Netzteil 2 mit 24 V AC (50/60 Hz), max. 3 A

**PoE-Standard** IEEE 802.3at

**Leistungsaufnahme**

Aufputzvariante: max. 25 W

---

**Mechanische Daten**

**Abmessungen** Aufputzvariante: ca.  $\varnothing$  166 H 244 mm

**Gewicht** Aufputzvariante: ca. 1755 g

---

**Umgebungsbedingungen****Betriebstemperatur**

Aufputzvariante:  $-10^{\circ}\text{C}$  bis  $+40^{\circ}\text{C}$  (Indoor)

**Relative Luftfeuchtigkeit**

0% – 90% RH, nicht kondensierend

**Schutzart** IP67 (Wetterschutzvariante)

---

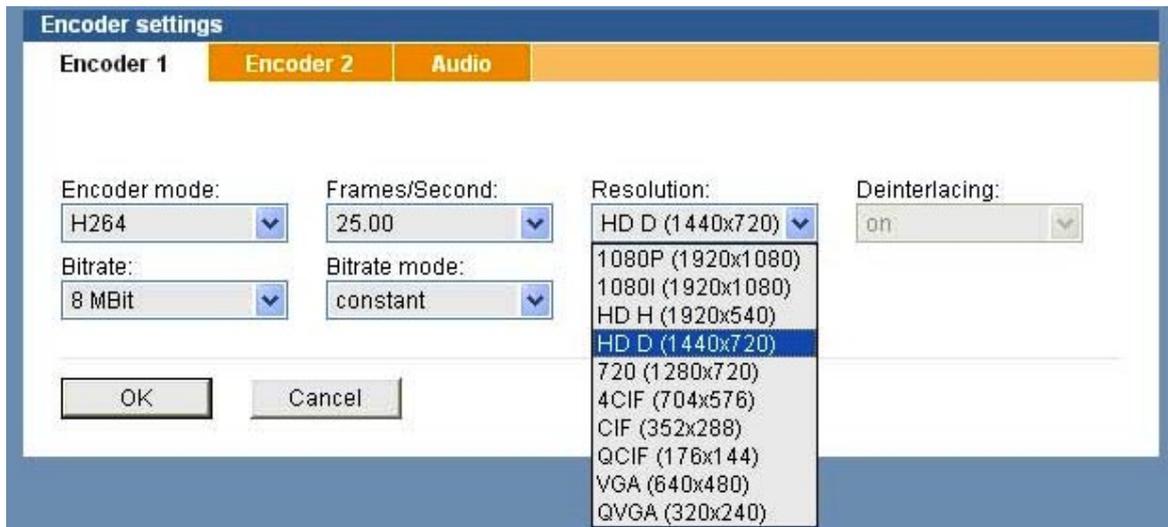


Abbildung A.5.: Encoder-Einstellung in PTZ-Domekameras

### A.0.3. Kamera – Funktionalitäten

Zusätzlich zu den oben beschriebenen Eigenschaften besitzt die Kameras zahlreiche Funktionen, Schnittstellen und Spezifikationen, sie sind der Dokumentation [Quelle: [Dallmeier \(2013\)](#)] des Herstellers zu entnehmen.

#### DualStreaming der Kamera

Die Kamera verfügt über ein simultanes Multi-Streaming mit (H.264 und MJPEG) mit unabhängig einstellbaren Auflösungen der Bild- und Bit-Raten in den einzelnen Streams.

Die Kameras sind mit zwei Encoder ausgestattet. Sie ermöglichen dem Empfänger des Video-Streams die Nutzung der Funktion DualStreaming.

Jede Kamera ist mit zwei Encodern ausgestattet, deren Einstellungsmöglichkeiten in Abbildung [A.5](#) dargestellt sind.

Die Encoder der Kameras können zum Erzeugen und Versenden von zwei Videostreams mit unterschiedlicher Qualität konfiguriert werden, wobei Bildrate (fps) und Auflösung von Encoder 2 in Abhängigkeit von Encoder 1 eingestellt werden.

In der Regel wird Encoder 1 auf eine hohe und der Encoder 2 auf eine niedrige Qualität eingestellt. Auch diese Einstellungen werden über das Webbrowser manuell gesetzt werden.

In dem Whitepaper der Firma Dallmeier HD-Kameras DualStreaming enthält die Beschreibung, wie die Encoder der PTZ-Domekamera zu Konfigurationen werden können.

Über die Webbrowser zur Konfiguration der Kameras besteht die Möglichkeit viele Einstellungen für die Übertragung der Video-Streams vorzunehmen.

Für jeder Video-Streams Übertragung bietet der Encoder die Einstellung der Encoder mode(), die Framerate, die Bitrate, der Bitmode und die Auflösung (Resolution).

Dabei werden die Auflösungen kategorisiert unter den Bezeichnungen HD, VGA, NTSC und PAL. Diese Einstellungen nehmen Einfluss auf die Übertragung der Video Stream.

<b>HD</b>		<b>VGA</b>	
Bezeichnung	Auflösung	Bezeichnung	Auflösung
1080p	1920 × 1080	VGA	640 × 480
1080i	1920 × 1080	QVGA	320 × 240
HD H	1920 × 540		
HD D	1440 × 720		
720p	1280 × 720		

<b>NTSC</b>		<b>PAL</b>	
Bezeichnung	Auflösung	Bezeichnung	Auflösung
4SIF	704 × 480	4CIF	704 × 576
SIF	352 × 240	CIF	352 × 288
QSIF	176 × 120	QCIF	176 × 144

Abbildung A.6.: Bezeichnung der Auflösungen

Der Zugriff auf jede Video-Stream erfolgt über den Aufruf der IP-Adresse der Kameras ergänzt mit Slug../livestream0 für den Encoder1 und /livestream1 für den Encoder 2

. Encoder 1 `rtsp://IP-Adresse/encoder1`

Encoder 2 `rtsp://IP-Adresse/encoder2`

Diese Funktionalität ermöglicht den Zugriff der Video-Stream in einzelnen Frame, bei der der Benutzer die Möglichkeit hat, jederzeit die Einstellungen manuell zu verändern.

### Preset Position

Eine weitere hilfreiche Funktionalität der Kameras sind die programmierbaren Preset-Positionen. Bis zu acht verschiedene Positionen inkl. Zoom-, Blenden- und Fokuseinstellungen der

Kamera können gespeichert werden. Bei Auswahl einer der Preset Position schwenkt die Kamera dorthin. Die Preset Geschwindigkeit kann manuell eingestellt werden.

Eine zusätzliche Funktion, der sogenannte virtueller Rundgang, dieser ermöglicht, dass eine Kamera vordefinierte Routen/ Bewegungen Durchläuft und dabei die Umgebung mit voreingestellten

Eine Route besteht aus mehreren Koordinaten Punkten, welche die Kamera nacheinander abfährt. Für den Weg von einem Routenpunkt zum nächsten kann die Kameraeinstellungen vordefiniert werden. Die Schwenkgeschwindigkeit, die Zoomeinstellung, die Neigung und viele mehr Einstellungen können vorgenommen werden.

Diese Funktionalität eignet sich z.B. für Videoaufnahmen einer bestimmten Szene, in dem Analysen auf Basis bestimmte Aktionen in einer Szene durchgeführt werden.

#### **A.0.4. PTZ Konfiguration**

Ein weiteres wichtiges Funktionalität der PTZ-Domekamera sind die integrierten Pan/Tilt/Zoom-Mechanismen. Diese Mechanismen ermöglichen Bewegungen der Kamera in fast alle Richtungen: das Schwenken um 360 Grad bei Neigungen zwischen 0 und 180 Grad, wobei die Geschwindigkeit der Bewegung variabel einstellbar ist.

Die manuelle Konfiguration und die Steuerung der PTZ-Domekamera erfolgt über einen Webbrowser. Über den Webbrowser kann jede Kamera individuell konfiguriert und eingestellt. Diese Einstellungen kann gespeichert werden, und bei Bedarf auf jede andere Kamera in der Wohnung übertragen.

Die PTZ-Konfiguration bzw. die Steuerung der Kameras kann auf einer grafischen Oberfläche über den Webbrowser für jede einzelne Kamera (siehe Abb. A.7) oder über die von Dallmeier zur Verfügung gestellte PView Software<sup>3</sup> vorgenommen werden.

Im Dialog PTZ-Konfiguration des Webbrowsers (vgl. B in der Abb. ??) können für die Kameras die Zoom-, Blenden- und Fokuseinstellungen vorgenommen werden. Mit den vier Richtungstasten können die Kameras manuell in alle Richtungen geschwenkt werden.

Die Benutzerverwaltung ermöglicht die Definition verschiedener Zugriffs- und Konfigurationsrechte für drei unterschiedliche Benutzergruppen. Falls erforderlich können den Benutzergruppen zudem einzelne Benutzer zugeordnet werden.

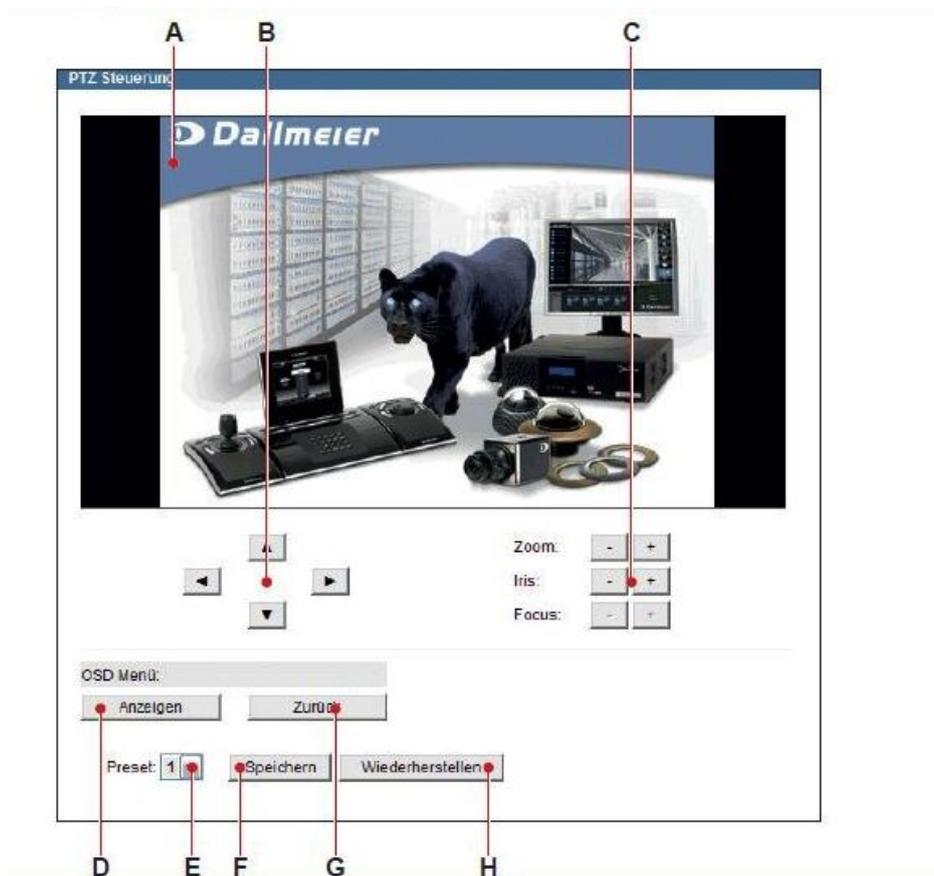


Abbildung A.7.: Kamera Steuerung über den Webbrowser

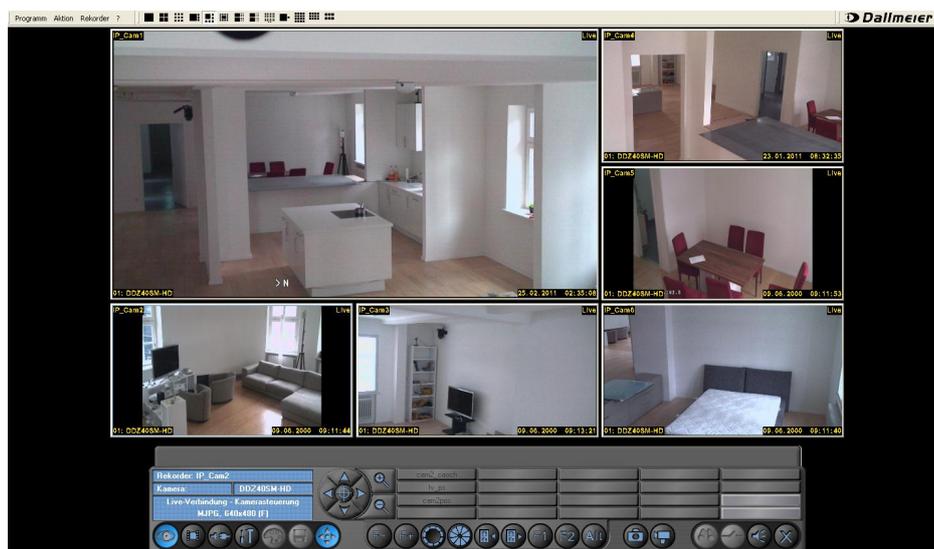


Abbildung A.8.: PTZ-Steuerung über die PView Software .

## **PView Softwares**

Die PView Software bietet ein Bedienpanel für die PTZ-Steuerung, dargestellt in der Abbildung [A.8](#), das sich über Maus oder Joystick bedienen lässt. Sie ermöglicht die manuelle Steuerung der Kameras.

Weiterhing bietet die PView Software viele weitere Funktionen und Einstellungsmöglichkeiten wie z. B. die Speicherung von Videostreams über eine Rekorder-Funktion, den Bildexport oder die Backup-Erstellung, die aus der Dokumentation der PView Software <sup>4</sup> zu entnehmen sind.

---

<sup>3</sup><http://www.dallmeier-electronic.com/de/cctv-ip-videoueberwachung/software/pview-7.html>

<sup>4</sup><http://www.dallmeier.com/en/cctv-ip-video-surveillance/product-history/software/pview.html>

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg,  
\_\_\_\_\_  
Ort, Datum

\_\_\_\_\_  
Unterschrift