



Technische Universität Ilmenau Fakultät
für Informatik und Automatisierung
Fachgebiet Neuroinformatik und Kognitive
Robotik

Entwicklung und Integration von Interaktionsstrategien zur Erkennung und Behandlung von Notfallsituationen im häuslichen Umfeld durch Service-Roboter

Masterarbeit zur Erlangung des akademischen Grades Master of Science

Benjamin Schütz

Betreuer:
Dipl.Inf. Steffen Müller
M.Sc. Thomas Schmiedel

Verantwortlicher Hochschullehrer:
Prof. Dr. H.-M. Groß, FG Neuroinformatik und Kognitive Robotik

DOI: 10.22032/dbt.47959

URN: urn:nbn:de:gbv:ilm1-2021200028

Die Masterarbeit wurde am 02.01.2017 bei der Fakultät für
Informatik und Automatisierung der Technischen Universität Ilmenau
eingereicht.

Inhaltsverzeichnis

1	Einleitung	1
2	Stand der Technik	5
2.1	Notfallerkennung	5
2.2	Positionsbestimmung einer Person	11
2.3	Alarmierungskonzept	16
3	Aufbau und Funktion des MIRA-Frameworks	19
4	Roboterplattform Tweety	23
5	Implementierung einer Strategie zur Notfallbehandlung	27
5.1	Aufbau eines Services zur Notfallbehandlung	29
5.2	Genutzte Roboterskills	33
5.2.1	Erkennung liegender Personen	33
5.2.2	Navigation	35
5.2.3	Detektion stehender Personen	36
5.3	Module der Notfallerkennung	37
5.3.1	Erzeugung eines NoActivity-Alarms	38
5.3.2	Erzeugung eines Immobility-Alarms	41
5.4	EMERGENCYSEARCH-Behavior	44
5.5	VERIFYFALLENPERSON-Behavior	48
5.6	Ablaufsteuerung / Dialog	53

6 Funktionstest	57
6.1 Aufbau der Versuchsumgebung	58
6.2 Durchführung des Funktionstests	61
6.3 Ergebnisse und Ergebnisdiskussion	64
7 Zusammenfassung und Ausblick	75
7.1 Zusammenfassung	75
7.2 Ausblick	79
A Protokolle des Funktionstests	83
Literaturverzeichnis	97

Kapitel 1

Einleitung

Ein gesundheitlicher Notfall tritt plötzlich und ohne Vorwarnung ein. Dabei handelt es sich um eine lebensbedrohliche Störung des Bewusstseins, der Atmung, des Herz-Kreislaufs, des Stoffwechsels oder Störungen mit traumatologischen Ursachen. Dazu gehören Apoplex, Herzinfarkt, Blutungen, Krämpfe, Frakturen, plötzliche Querschnittslähmungen bei bestehenden Tumorerkrankungen oder Exazerbationen [WIESE et al., 2011]. Vor allem ist es ein beängstigendes Erlebnis für die Betroffenen. Auch Stürze, die aus altersbedingte Bewegungseinschränkungen resultieren, führen, selbst wenn sie nicht gefährliche Verletzungen hervorrufen, dazu, dass Senioren unsicher werden und sich weniger zutrauen. In der Folge verlassen sie seltener ihre Wohnungen, wodurch sich auch die Teilhabe am gesellschaftlichen Leben reduziert. Diese Isolation begünstigt dabei jedoch nicht nur den körperlichen Abbau. Auch die kognitive Leistungsfähigkeit leidet unter der fehlenden sozialen Interaktion.

Die steigende Angst vor eintretenden Notfällen und die sich daraus ergebende Unsicherheit führt folglich dazu, dass sich Senioren selbst oder auf Anraten von Angehörigen entscheiden, in Pflegeeinrichtungen zu ziehen. Das Zurücklassen der eigenen Wohnung kann dabei den bereits in Gang gebrachten körperlichen und geistigen Abbau zusätzlich vorantreiben.

Es ist daher wünschenswert, Senioren ein selbstbestimmtes und selbstständiges Leben innerhalb ihrer vertrauten Umgebung zu ermöglichen. Dabei sollen sie bei der Verrichtung alltäglicher Tätigkeiten unterstützt, aber auch körperlich und kognitiv ge-

fördert und gefordert werden. Um dieses Ziel zu erreichen gibt es seit einigen Jahren unterschiedliche Ansätze, wie Roboter genutzt werden können, um Senioren im Zusammenleben wertvolle Hilfe leisten zu können. Da die für diesen Zweck eingesetzten Roboter Unterstützung durch Anbieten von Services liefern, werden sie als Service-Roboter bezeichnet. Darüber hinaus können Roboter auch emotionale Stütze sein und werden dadurch zu einem Begleiter im Leben der Senioren. Ein Roboter, der mit dem Nutzer interagiert und sich auf dessen Bedürfnisse einstellen kann und dabei durch Mimik oder sein Verhalten Emotionen transportiert, wird als Companion bezeichnet. Durch den emotionalen Bindungsaufbau steigt die Akzeptanz beim Nutzer.

Das Fachgebiet für Neuroinformatik und kognitive Robotik der TU Ilmenau hat zu diesem Zweck zusammen mit weiteren Projektpartnern einen sozialen Service-Roboter entwickelt. Ziel des von 2012 bis 2015 laufenden Projektes SERROGA (**S**ervice-**R**obotik für die **G**esundheits**A**ssistenz)¹ war es, die Lebenssituation älterer Menschen durch die Entwicklung eines Service-Roboters zu verbessern. Dieser sollte dabei die Aufgaben eines Bewegungsmotivators und die eines Assistenten für Kommunikation und Terminübersicht erfüllen.

Das seit 2015 und bis 2018 laufende Projekt SYMPARTNER (**S**ymbiose von **P**aul und **R**obo**T**er Compa**N**ion für eine **E**motionssensitive Unte**R**stützung)² schließt an SERROGA an und verbindet die Funktionen des Service-Roboters mit den Funktionen des PAUL-Systems (**P**ersönlicher **A**ssistent für **u**nterstütztes **L**eben). Das PAUL-System liefert Funktionen zur Erinnerung und zur Haussteuerung, ist jedoch auf einen PC oder ein Tablet beschränkt. Der Assistenz-Roboter hat nur einen geringen Umfang alltagsrelevanter Funktionen, bietet aber emotional-soziale Kommunikation und ist in der Lage, sich selbstständig durch seine Einsatzumgebung zu bewegen und damit dem Nutzer Funktionalitäten direkt anzubieten.

Die erfolgversprechende Symbiose beider Systeme bietet damit viele Möglichkeiten, Senioren im alltäglichen Leben zu unterstützen. Jedoch fehlt bislang eine verlässliche Strategie, um Notfälle zu erkennen und bei deren Auftreten angemessen und schnell

¹<http://www.serroga.de/>

²<http://www.sympartner.de/>

darauf zu reagieren. Vor allem im Alter treten hingegen deutlich häufiger Situationen auf, die lebensbedrohlich sein können. An der Spitze der häufigen Notfälle stehen Herzinfarkte und Schlaganfälle. Eine schnelle Hilfe ist in diesen Fälle wichtig, um die Folgen für die Betroffenen möglichst gering zu halten. Leben Senioren mit einem Partner zusammen, ist die schnelle Reaktion des Partner essentiell, um schwerwiegende Folgen abzuwenden. Schwieriger wird es, wenn Menschen allein leben. Vom Auftreten des Notfalls bis zum Auffinden der Personen können unter Umständen wertvolle Minuten bis in zu Stunden vergehen. Im Folgenden berichten drei Patienten, wie sie ihren Schlaganfall wahrgenommen haben. Dabei wird das Gedächtnisprotokoll einer Mitarbeiterin des Therapeutenteams des neurologischen Rehabilitationszentrums Magdeburg herangezogen. Aus den Aussagen wird deutlich, wie lange die Betroffenen hilflos waren bis sie gefunden wurden.

„Ich bin frühmorgens aufgewacht und mir war schon ganz komisch, schwindlig und so. Nachmittags bin ich zum Kühlschrank gegangen und dann wurde mir schwarz vor Augen. Ich habe noch gemerkt, wie ich am Kühlschrank zusammengesunken bin. Dann weiß ich nichts mehr. Meine Frau hat mich nach der Arbeit dann gefunden.“ (Herr K., 63 Jahre)

„Das kam ja ganz plötzlich. Ich habe abends noch mit meinem Sohn telefoniert. Der wollte mir am nächsten Tag noch was vorbeibringen. Ein Glück. Ich bin einfach so umgefallen. Mein Sohn hat mich dann gefunden. Aber wann, weiß ich nicht.“ (Herr Sch., 57 Jahre)

„Ich wollte gerade mein Fahrrad in den Keller räumen und dann kam der Schlag. Ich habe noch laut gerufen, aber das hat niemand gehört. Dann wurde ich ohnmächtig. Im Krankenhaus bin ich wieder aufgewacht. Mich haben ein paar Hausmitbewohner gefunden, aber ich lag dort bestimmt zwei Tage.“ (Frau M., 81 Jahre)

Um Notfällen dieser Art in Zukunft schnell und verlässlich zu begegnen, muss ein, in Wohnungen von Senioren eingesetzter, Service-Roboter in der Lage sein, Notfälle zu erkennen und weitere Maßnahmen einzuleiten.

Ziel dieser Masterarbeit ist es, eine Interaktionsstrategie zur Erkennung und Behandlung von Notfallsituationen im häuslichen Umfeld zu entwickeln. Der dabei entstehende Service soll den Roboter nutzen und modular aufgebaut sein. Der Vorteil der angestrebten Modularität ist, dass der entstehende Ablauf einfach in zukünftige Anwendungen integriert werden kann und dabei jedoch problemlos erweiterbar bleibt. Um diesen Anforderungen gerecht zu werden, sollen Module durch den Autor erzeugt bzw. bereits existierende Module genutzt werden, die die notwendigen Funktionalitäten zur Erfüllung des geforderten Gesamtkonzeptes enthalten. Das Gesamtkonzept soll dabei aus der Erkennung von Notfällen, der Suche, dem Finden und der Verifikation des Nutzers sowie einem anschließenden Dialog zur Klärung der Situation bestehen.

Die vorliegende Masterarbeit ist wie folgt gegliedert. Kapitel 2 gibt einen Überblick über den aktuellen Stand der Technik. Dabei werden Konzepte zur Erkennung von Notfällen in häuslichem Umfeld sowie zur Positionsbestimmung der Bewohner vorgestellt. Zusätzlich wird gezeigt, welche Möglichkeiten der Alarmierung genutzt werden können. Kapitel 3 stellt das in dieser Arbeit genutzte Framework vor, mit dem Roboterapplikationen erzeugt, zusammengeführt und im Ablauf koordiniert werden können. In Kapitel 4 wird die Roboterplattform vorgestellt, für die die Strategie zur Notfallbehandlung implementiert wird. Das Kapitel 5 enthält die Implementierung des Notfallservices. Dabei soll in Kapitel 5.1 der allgemeine Aufbau einer Strategie zur Notfallbehandlung und die Integration der verschiedenen Module erläutert werden. Module, die bereits existieren und vom Roboter genutzt werden können, werden in Kapitel 5.2 vorgestellt. Die folgenden Kapitel beschreiben die Entwicklung der Module zur Erkennung von möglichen Notfällen (Kapitel 5.3), der Personensuche (Kapitel 5.4), der Verifikation einer liegender Person (Kapitel 5.5) und des Dialogs (Kapitel 5.6) und bilden damit den Kern der vorliegenden Masterarbeit.

In Kapitel 6 wird die Integration der in dieser Arbeit entwickelten Module mittels durchgeführter Funktionstests überprüft. Es wird der Ablauf beschrieben und am Anschluss werden die Testergebnisse vorgestellt und diskutiert. Das abschließende Kapitel (Kapitel 7) fasst die vorliegende Masterarbeit zusammen und gibt einen Ausblick über zukünftige Weiterentwicklungen.

Kapitel 2

Stand der Technik

Eine große Sorge allein lebender Senioren ist es, in eine Notsituation zu geraten, in der sie sich nicht bemerkbar machen oder Hilfe anfordern können.

Automatische Gesundheitsüberwachungssysteme sollen dabei helfen, Senioren auch im Alter ein eigenständiges Leben zu ermöglichen, indem ihnen und ihren Angehörigen die Angst davor genommen werden soll, in eine Notsituation zu geraten, die von anderen unbemerkt bleibt. Diese Systeme sind in der Lage, Notfälle zu erkennen und Angehörige bzw. das Pflegepersonal einer Einrichtung für betreutes Wohnen zu alarmieren. Allen diesen Systemen ist ein grundsätzlich ähnlicher, modularer Aufbau gemein. Es lassen sich in der Handlungskette solcher Überwachungssysteme drei große Module identifizieren: die *Notfallerkennung*, die *Positionsbestimmung* der hilfsbedürftigen Person sowie die *Alarmierung* bzw. die Absetzung eines Notrufes. Je nach betrachtetem System befinden sich die einzelnen Module mehr oder weniger stark im Fokus und sind deshalb in den verschiedenen Konzepten unterschiedlich stark ausgeprägt. Entsprechend der genannten Dreiteilung soll auch der Aufbau dieses Kapitels strukturiert sein.

2.1 Notfallerkennung

Die denkbar einfachste Form der Notrufgenerierung, die gegenwärtig auch die verbreitetste ist, ist ein tragbarer Notrufknopf. Fühlt sich der Träger unwohl, kann er über den Notrufknopf auf sich aufmerksam machen. Das gesendete Signal wird von

einer Empfangsstation erfasst und weitergeleitet. Mehrere Empfangsstationen zusammen bilden ein Notrufsystem an dessen Ende meist ein Mitarbeiter eines Pflegedienstes die eingehenden Notrufe entgegennimmt. In privaten Haushalten sind solche Systeme selten zu finden, da sie einen Menschen voraussetzen, der die Informationen vieler separater Systeme empfängt und die notwendige Hilfe organisiert. In Pflegeheimen und Einrichtungen des betreuten Wohnens ist die Vernetzung mehrerer solcher Systeme in einem Gebäude sowie die zentrale Auswertung dagegen problemlos möglich. Ein großer Nachteil tragbarer Notrufknöpfe ist darin zu sehen, dass hierbei eine aktive Mitarbeit des hilfsbedürftigen Menschen notwendig wird. Ist der Nutzer aufgrund eines plötzlichen Ereignisses nicht mehr in der Lage, rechtzeitig Alarm auszulösen oder trägt er das Gerät in diesem Augenblick nicht am Körper, bleibt der Notfall unbemerkt.

Um nicht auf die aktive Alarmierung durch den Menschen angewiesen zu sein, werden Systeme entwickelt, die einen Notfall dann erkennen, wenn aktuelle Beobachtungen vom normalen Verhalten der Person abweichen. Die Erzeugung von Beobachtungen kann dabei auf unterschiedlichen Wegen erfolgen. So gibt es Systeme, die am Handgelenk getragen werden können und Vitaldaten, wie Puls und Hauttemperatur des Trägers erfassen. Diese Informationen dienen der Beurteilung des allgemeinen Gesundheitszustandes und deuten auf einen Notfall, wenn z.B. die Herzschlagfrequenz über einen bestimmten Zeitraum zu hoch oder zu niedrig ist. Zusätzlich erfasst das Gerät die Bewegungen des Nutzers und wertet die Bewegungsmuster aus. Dadurch kann ein Sturz anhand seines typischen Musters (schnelle Lageänderung mit anschließender Phase der Bewegungslosigkeit) erkannt und ein Notruf generiert werden. Durch die hohe Genauigkeit in der Erkennung von Stürzen sowie kritischer Gesundheitszustände sind tragbare, passive Geräte nützliche Hilfsmittel zur Beurteilung von Notsituationen. Ein Beispiel ist hierfür ist das Wrist Device, das im Rahmen des EU-Projektes EMERGE zur Erkennung von Notfällen entwickelt wurde [KLEINBERGER et al., 2011]. Ein Nachteil, den sich diese Geräte mit tragbaren Notrufknöpfen teilen, bleibt, dass sie die Mithilfe des Nutzers erforderlich machen. Sie müssen angelegt, abgenommen, eventuell auch eingestellt oder aufgeladen werden. Die Nutzerfreundlichkeit und damit die Akzeptanz des Gerätes, das in Form und Größe vergleichbar einer Armbanduhr ist,

wird zwar als sehr gut beurteilt, jedoch bleibt der Nachteil, dass es keine Möglichkeit zu Erzeugung eines Alarms gibt, wenn das Gerät nicht durch den Nutzer erreichbar ist [KLEINBERGER et al., 2011]. Ein ähnliches Prinzip, bei dem ebenfalls Stürze anhand der typischen Bewegungsmuster erkannt werden, die ein Beschleunigungssensor erfasst, wird in [TABAR et al., 2006] vorgestellt.

Um diesem Nachteil zu überwinden, den alle Geräte haben, die vom Nutzer getragen werden müssen, setzen andere Systeme auf die Auswertung von Daten, die ohne mitzuführende Geräte erfasst werden können. Eine Möglichkeit bietet die Nutzung einer sensitiven Fußbodenunterlage aus leitfähigem Textil [TUZINSKI und DRESCHER, 2008]. Diese Unterlage ist so strukturiert, dass 32 einzelnen Sensorflächen auf einem Quadratmeter entstehen, die den Druck erfassen, der auf sie ausgeübt wird. Jeweils acht Sensorflächen sind mit einem in die Unterlage integrierten Empfangsmodul verbunden. Jedes Modul überwacht die Zustände der angrenzenden Sensorflächen und übermittelt diese an einen zentralen Funkempfänger, der die Auswertung für einen Raum übernimmt. In die Sendemodule integrierte Temperatursensoren ermöglichen, neben der orts- und zeitaufgelösten Erfassung von Bewegungen, auch Aussagen zur Temperaturverteilung im Raum. Durch dieses System ist es möglich, den Aufenthaltsort der Bewohner zu ermitteln und entsprechende Komponenten der Hausautomation zu schalten. Darüber hinaus kann die sensitive Fußbodenunterlage auch der Erhöhung der Sicherheit allein lebender Personen dienen. Auch hier werden Bewegungsmuster ausgewertet, um kritische Situationen zu erkennen [TUZINSKI und DRESCHER, 2008]. Stürzt eine Person und bewegt sich darauf hin nicht, wird ein Notruf abgesetzt. Auch eine längere Inaktivität oder das lange Verweilen in einem dafür untypischen Ort, kann als Indiz für eine Notsituation genutzt werden. Obwohl das Konzept der sensitiven Fußbodenunterlage ein weites Feld an Anwendungen ermöglicht, erschweren die hohen Kosten und der Aufwand für die Installation die flächendeckende Verbreitung. Daraus ergibt sich, dass Konzepte zur Notfallerkennung im häuslichen Umfeld den Forderungen nach einer einfachen Installation, ohne große Veränderung an der bestehenden Einrichtung, sowie einer kostengünstigen Lösung genügen muss.

In [PFLÜGER et al., 2012] wird ein System zur Notfallerkennung, basierend auf Daten einer 3D Kamera, vorgestellt. Ziel dabei war es, ein Unfallerkennungssystem zu entwickeln, das eine schnelle Erkennung von Notfällen gewährleistet, wenig Fehlalarme produziert und Stürze erkennen kann. Da optische 2D Sensoren zu anfällig gegen Beleuchtungsänderungen und Schattenwürfe sind und Wärmebildkameras sowie Laserscanner als zu teuer ausscheiden, bieten 3D Sensoren das größte Potenzial. Die kostengünstigste Lösung fanden die Autoren im Stereo-Video-Prinzip. Um Modularität in der Hardware zu gewährleisten können, wurde bei der Entwicklung der Auswertelgorithmen darauf geachtet, dass diese unabhängig vom verwendeten Sensor sind. Die Auswertung erfolgt daher nicht im aufgenommenen Bild, sondern es werden aus den Daten sensorunabhängige 3D-Punktwolken berechnet. Die in der Szene enthaltenen Personen werden durch dynamische Hintergrundsegmentierung detektiert. Wurden Personen gefunden, werden die Hauptachsen und Schwerpunkte anhand der Personenumrisse bestimmt wodurch die Lage der Person im Raum ermittelt werden kann. Um in einer Szenen einen Notfall zu erkennen, wird die Bewegung des Schwerpunktes analysiert. Eine schnelle Änderung der Höhe kann dabei auf einen Sturz hindeuten. Zusätzlich werden die Hauptachsen verwendet, um im Raum liegende Personen zu erkennen. Ausgehend von diesen Beobachtungen kann das Notrufsystem in Alarmbereitschaft versetzt werden [PFLÜGER et al., 2012].

Auch wenn Konzepte, die Kameras zur Datenerfassung nutzen, den Vorteil bieten, dass mit ihnen schnell und robust die Personenerkennung in der erfassten Szene erfolgen kann, ist zu bedenken, dass eine, in privaten Räumen installierte Kamera, zum Gefühl der Überwachung beim Nutzer führen kann. Zwar betonen [PFLÜGER et al., 2012], dass die aufgezeichneten Rohdaten der Kameras die Wohnung nicht verlassen, da die Verarbeitung und Beurteilung direkt in der Auswerteeinheit in der Wohnung stattfindet. Lediglich der Notruf, infolge einer erkannten Notsituation, verlässt anschließend die Wohnung. Trotzdem wird bei vielen Nutzern die Frage bleiben, ob es nicht doch möglich ist, von außen auf die Kameras zuzugreifen, was zu einer verringerten Akzeptanz gegenüber solchen Systeme führt. Um von vornherein solchen Sorgen zu begegnen, sind Systeme vorzuziehen, bei denen sich Fragen zu Datenschutz und Privatsphäre nicht in

dem Maße wie bei Kameras stellen.

Um die Robustheit des Notfallsystems zu steigern, haben [PFLÜGER et al., 2012] ihrem System eine akustische Auswertung hinzugefügt. Auf Basis von in der Wohnung installierten Mikrofonen und der Analyse durch Hidden-Markov-Modelle sollen typische Hilfeschreie detektiert werden können. Es wird die Wahrscheinlichkeit für einen vorliegenden Hilfeschrei berechnet, die als zusätzlicher Parameter in die Auswertung der Bilddaten eingeht. Da eine Trennung des gesuchten akustischen Signals von störenden Umgebungsgeräuschen nicht ohne weiteres möglich ist, ist eine Systemkonzeption nur auf Basis akustischer Sensoren nicht zuverlässig genug. Die akustische Auswertung ist im genannten Konzept deshalb nur als eine ergänzende Komponente zu verstehen.

Die Integration von Bewegungsmeldern ist eine weitere Möglichkeit, ein System zur Notfallerkennung einzurichten, welches zum einen kostengünstig zu realisieren ist und zum anderen die Privatsphäre der Bewohner besser schützt, als es mit kamerabasierten Systemen möglich wäre. In [FLOECK und LITZ, 2008] wird ein System beschrieben, bei dem neben Bewegungsmeldern weitere Sensoren das Öffnen und Schließen von Fenstern und Türen sowie das Betätigen der elektrischen Rollläden und der Wasserhähne registrieren. Als Bewegungsmelder dienen dabei PIR-Sensoren (passive infrared sensors), die die Wärmestrahlung sich bewegendere Personen und Objekte erfassen. Mit diesen Informationen wurde die tageszeitabhängige Aktivität der Bewohner erfasst. Es zeigte sich, dass die Erfassung der Aktivitätsprofile einige Nachteile brachte, die durch die Betrachtung der Inaktivitäten umgangen werden konnte. Das ermöglichte, neben der Nutzung einer einfach zu erzeugenden, kontinuierlichen Zeitachse, eine einfache Zusammenführung von Daten unterschiedlicher Sensoren. Die zugrunde liegende Idee ist einfach: Die Inaktivität, d.h. die Zeit, in der keine Aktivität des Bewohners registriert wurde, wird kontinuierlich hochgezählt. Sobald eine Aktivität erfasst wird, wird der Inaktivitätszähler zurückgesetzt. Dabei ist es unwichtig, welcher Sensor die Aktivität erfasst hat. Auf diese Weise lassen sich Inaktivitätsgraphen für einzelne Tage erzeugen. Um einen Inaktivitätsgraphen als Referenz zu erhalten, wurden die Inaktivitätsgraphen von 31 Tage gemittelt. Aus dieser Referenz werden Inaktivitätsschwellen erzeugt, bei deren Überschreiten ein Alarm ausgelöst wird. Um bei Abwesenheit des

Bewohners keinen Alarm infolge vermeintlicher Inaktivität auszulösen, werden die Türkontakte genutzt, um zwischen Inaktivität bei An- und Abwesenheit zu entscheiden. Die zulässige Inaktivitätsschwelle bis zur Auslösung eines Alarms wird entweder auf einen beliebigen, festen Wert gesetzt oder kann abhängig von den aufgezeichneten Inaktivitätsmustern des Bewohners variieren. Um die Zahl der Fehlalarme auf eine akzeptables Maß zu beschränken, darf die Alarmschwelle nicht beliebig reduziert werden. Deshalb ist es jedoch nicht möglich, Notfälle innerhalb sehr kurzer Zeitspannen zu erkennen. Neben der Notfallerkennung kann das System verwendet werden, um ungewöhnliche Situationen erkennen, wie z.B. angelehnte Wohnungstüren oder laufende Wasserhähne [FLOECK und LITZ, 2008].

Auch [CUDDIHY et al., 2007] nutzen PIR-Bewegungsmelder, um Notfälle anhand ungewöhnlich langer Inaktivitäten zu erkennen. Im Unterschied zu einer einfachen, festen Inaktivitätsschwelle für den gesamten Tag, werden für jeden Zeitbereich individuelle Schwellen berechnet. In die Berechnung geht der um Ausreißer bereinigte Verlauf der vergangenen Inaktivitäten ein sowie ein Minimumpuffer und ein Term, der die Sensitivität während Zeiten hoher Aktivität erhöht und sie bei niedriger Aktivität verringert. Der Minimumpuffer wird genutzt, um Fehlalarme in Zeitspannen mit hoher Aktivität zu vermeiden. Zudem werden zu jeder Zeitspanne auch die benachbarten Zeitspannen betrachtet, um eine kleinere zeitliche Verschiebungen in der Aktivität des Bewohners zu berücksichtigen, ohne sofort ein Alarm ausgelöst wird. Trotz des Konzeptes der individuellen Anpassung der Alarmschwelle für jeden Zeitabschnitt und der Adaption der Schwellen über den Zeitraum des Betriebs an die Aktivitätsmuster des Bewohners, bleibt auch hier das Problem, dass Alarmschwellen nicht beliebig herabgesetzt werden können, da sonst die Zahl der Fehlalarme auf ein nicht tolerables Maß steigt. Der Ansatz, die beobachteten, individuellen Inaktivitätsmuster zu nutzen, um tageszeitabhängige Alarmgrenzen einführen zu können, kann jedoch dazu führen, Notfälle deutlich schneller zu erkennen, als es mit festen Alarmgrenzen möglich wäre. Das Ziel soll dabei die Balance zwischen vielen Fehlalarmen (falsch-positive Alarme) durch zu hohe Sensitivität und dem Übersehen eines tatsächlichen Alarms (richtig-positiver Alarm) durch zu geringe Sensitivität sein [CUDDIHY et al., 2007].

2.2 Positionsbestimmung einer Person

Um in einem Notfall mittels mobiler Roboterplattform schnell Personen finden zu können, bedarf es eines Systems, mit dem die Position der Person in der Wohnung bestimmt werden kann. In der Literatur gibt es unterschiedliche Sensorkonzepte, die zur Positionsbestimmung in geschlossenen Räumen genutzt werden. An dieser Stelle sei die Bestimmung mittels Bewegungsmelder ([CRANDALL und COOK, 2011], [LI et al., 2015], [LUO und CHEN, 2013], [YANG et al., 2015]), Kameras ([ABOWD et al., 2002a], [ABOWD et al., 2002b], [CUCCHIARA et al., 2007], [YU et al., 2007]), Mikrofone [ABOWD et al., 2002a], Drucksensoren [YU et al., 2007], RFID-Marken [KOCH et al., 2007], Signalstärkenmessung ([TABAR et al., 2006], [KALTIOKALLIO et al., 2012]) und Messung der Signallaufzeiten [BHARADWAJ et al., 2014] genannt. Einige Systeme dienen der konkreten Erkennung von Notfällen, während andere genutzt werden, um die Aktivitäten des täglichen Lebens zu erfassen. Diese Erfassung wird genutzt, um Abweichungen in Tagesabläufen oder typischen Aktivitäten oder Änderungen des körperlichen und geistigen Zustandes des Nutzers zu erkennen.

Häufige Verwendung finden Bewegungsmelder (passive infrared, PIR). In [LI et al., 2015] werden acht PIR-Sensoren zu einem an der Decke angebrachten Sensornetz angeordnet, um Bewegungen in einer 25 m^2 großen Testumgebung zu erfassen. Der Abstand zwischen zwei benachbarten Sensoren beträgt $1,5\text{ m}$, wodurch Überlappungen der Erfassungsbereiche entstehen. Bewegt sich eine Person durch die Testumgebung, kann aus der Kombination der ausgelösten Sensoren die Position der Person bestimmt werden. Aus der Sequenz der Sensorkombinationen kann auf diese Weise der zurückgelegte Weg der Person nachvollzogen werden. Untersuchungen zeigten jedoch teilweise Abweichungen des errechneten Pfads vom tatsächlichen Pfad. Somit eignet sich das System nur für eine grobe Bestimmung der Personenposition [LI et al., 2015]. Bewegungsmelder werden auch in [CRANDALL und COOK, 2011] genutzt, um Positionen von Personen und deren Bewegung in Testumgebungen zu bestimmen. Hierfür wurden im CASAS-Projekt zwei Testumgebungen mit jeweils 44 bzw. 52 Bewegungsmeldern ausgestattet. Es werden zwei Algorithmen verwendet, die der Lokalisierung und der Verfolgung der Bewegungen in der Testumgebung dienen. Der erste Algorithmus nutzt

dafür festgelegte Regeln, wie sich Personen bewegen und die Kenntnis der möglichen Routen, die sich zwischen den Sensoren aufgrund der Sensoranordnung ergeben. Der zweite Algorithmus erweitert den ersten Ansatz, indem aus Trainingsdaten die Wahrscheinlichkeiten für die Übergängen zwischen den Bereichen ermittelt wurden. Mithilfe dieser Konzepte lässt sich die Position von Personen in der Testwohnung mit einer Genauigkeit von 72,2 % bzw. 85 % bestimmen. Schwerpunkt war dabei jedoch nicht nur die genaue Positionsbestimmung der Personen, sondern die Fähigkeit zu jedem Zeitpunkt Aussagen über Personenanzahl in der Wohnung, sowie über deren zeitlichen Bewegungsverlauf machen zu können. Im Gegensatz zu anderen Ansätzen kann das System die Bewegungen mehrerer Personen, die sich gleichzeitig in der Wohnung aufhalten, unterscheiden [CRANDALL und COOK, 2011].

Auch in [LUO und CHEN, 2013] werden PIR-Sensoren zur Positionsbestimmung genutzt. Da die Bewegungsmelder jedoch für die exakte Positionsbestimmung zu ungenau sind, wurde das System um zusätzliche Hochfrequenz-Empfängermodule zur Auswertung der Received Signal Strength (RSS), der sich im Raum ausbreitenden Hochfrequenzsignale, ergänzt. Durch die Fusion der verschiedenen Konzepte konnte die Genauigkeit des Systems bei der Positionsbestimmung gesteigert werden [LUO und CHEN, 2013].

In [YANG et al., 2015] werden ebenfalls PIR-Sensoren zur Erfassung der Nutzerbewegungen verwendet. Da die Erfassungsbereiche der Sensoren überlappend angeordnet sind, ergeben sich aus deren Überschneidungen viele einzelne Bereiche die unterschieden werden können. Dadurch wird die Genauigkeit in der Positionsbestimmung erhöht. Gleichzeitig werden in einer Karte Bereiche, in denen Möbel stehen, als nicht zugänglich vermerkt. Damit können die möglichen Aufenthaltsorte der Person deutlich eingeschränkt werden. Mittels Partikelfilter werden die Informationen zur Begehbarkeit der Bereiche und die Sequenz der erfassten Bewegungen zu einer Schätzung über die Position der Person zusammengefasst [YANG et al., 2015].

Auch kamerabasierte Konzepte werden zu Positionsbestimmung genutzt. In [CUCCHIARA et al., 2007] werden in der Wohnung Kameras verteilt, sodass jedes Zimmer mit mindestens einer Kamera ausgestattet ist. Aus den Bildern wird die Silhouette der

sich bewegenden Person extrahiert. Damit soll es möglich sein, die Bewegung der Person entlang aneinander angrenzender Räume zu verfolgen. Hidden-Markov-Modelle werden verwendet, um eine Positionsschätzung zu erzeugen, für den Fall dass die Person aufgrund von Verdeckungen nicht mehr im Bild zu sehen ist [CUCCHIARA et al., 2007].

In [ABOWD et al., 2002b] werden ebenfalls Kameras genutzt, um die Position und die Orientierung von Personen zu bestimmen. Dabei befinden sich die Kameras an der Decke der Wohnung und haben einen Öffnungswinkel von 120° . Nachdem die Kameras in der Umgebung installiert und kalibriert sind, erkennen sie Personen durch Hintergrundsubtraktion im Bild. Die Bilder der einzelnen Kameras werden daraufhin zur Auswertung zusammengeführt, um die räumlichen und zeitlichen Informationen zu erhalten.

Autoren wie [ABOWD et al., 2002a] und [YU et al., 2007] kombinieren mehrere Sensor-konzepte, um die Genauigkeit der Positionsbestimmung zu erhöhen. [ABOWD et al., 2002a] nutzt neben Kameras auch Mikrofone, die sich in allen Räumen der Wohnung befinden. Dabei sind alle Mikrofone ständig auf Empfang und erzeugen in regelmäßigen Abständen Audioaufnahmen von 5 s Länge. In diesen Aufnahmen sucht eine Spracherkennungssoftware nach Mustern, die auf gesprochene Sprache hinweisen. Werden solche Muster gefunden, vermerkt das System die Anwesenheit einer Person in diesem Bereich. Ergänzt wird das System durch RFID-Antennen, die in Fußmatten integriert sind. Der Nutzer trägt eine passive RFID-Marke in Höhe des Unterschenkels. Wird eine Fußmatte überquert, kann anhand der Marke der Nutzer identifiziert sowie seine Bewegung verfolgt werden [ABOWD et al., 2002a].

In [YU et al., 2007] werden 40 Druckssensoren verwendet, um die Position der Bewohner einer Wohnung zu bestimmen. Da die Positionsbestimmung mit Drucksensoren jedoch ungenau wird, sobald sich mehrere Personen nah beieinander befinden, reichen diese Sensoren zur alleinigen Bestimmung nicht aus. Ergänzt wird das System deshalb durch vier Kameras, die in den Ecken der Räume angebracht sind. Die Personenpositionen werden jeweils einzeln aus den Daten der Drucksensoren und der Kamerabilder ermittelt. Durch die Fusion der Daten aus unterschiedlichen Systemen sollen die ne-

gativen Einflüsse von Verdeckungen auf die Positionsbestimmung reduziert werden. Um die Erkennung noch robuster zu gestalten, wird der Condensation-Algorithmus genutzt. Dieser ist von Bayes-Filtern abgeleitet und ermöglicht die Positionsverfolgung mehrerer Personen. Durch Nutzung der Drucksensoren, der Kameras sowie des Condensation-Algorithmus' konnte in 78,8% der Fälle eine Genauigkeit von bis zu 30 cm erreicht werden [YU et al., 2007].

Neben Systemen zur Positionsbestimmung bei denen die verwendete Sensorik größtenteils in der Wohnung installiert ist, gibt es Systeme, bei denen der Nutzer die Sensoren am Körper oder in der Kleidung trägt. In [KOCH et al., 2007] wird ein System vorgestellt, bei dem der Nutzer ein RFID-Lesegerät am Schuh trägt. Unter dem Teppich der Wohnung befindet sich ein Netz aus 4000 Transpondern, die jeweils mit einem Abstand von 12,5 cm zueinander angeordnet sind. Die Transponder können bis auf eine Distanz von 10 cm vom Lesegerät erfasst werden, wodurch sich eine sehr genaue Positionsbestimmung ergibt [KOCH et al., 2007].

Auf der Infrarot-Entfernungsmessung basiert das Konzept nach [SCHINDLER et al., 2006]. An einem Gestell, ähnlich einem Kopfhörer, trägt der Nutzer zwei Infrarot-Entfernungsmesser, die je zu beiden Seiten des Kopfes ausgerichtet sind. Sie erfassen, wenn sich Objekte im Abstand von 10 cm bis 60 cm vor ihnen befindet. Ein zusätzlicher Beschleunigungssensor registriert die Bewegungen des Nutzers. Bevor die Position des Nutzers bestimmt werden kann, muss eine Karte der Umgebung erzeugt werden. Dazu bewegt sich der Nutzer durch die Räume der Wohnung. Geht der Nutzer durch eine Tür, wird dies anhand des erhaltenen Entfernungsprofils erkannt. Gleichzeitig kann der Nutzer die betretenen Räume mit Labeln versehen. Dazu zeigt er vor dem anderen Entfernungsmesser eine Anzahl Finger. Aus der Sequenz der erfassten Entfernungen wird diese Geste erkannt und dem Raum zugeordnet. Der Beschleunigungssensor zählt die Schritte, die der Nutzer zwischen den Räumen zurücklegt. Aus dieser Information wird die topologische Karte der Umgebung erzeugt, die die Beziehungen der Räume zueinander abbildet. Für die Lokalisation wird ein Partikelfilter verwendet, um aus den Messungen der Entfernungen und der Schritte, die Bewegung der Person entlang der Kanten der erzeugten Karte und damit ihre aktuelle Position zu schätzen

[SCHINDLER et al., 2006].

Ein weiteres tragbares Gerät zur Positionsbestimmung wird in [TABAR et al., 2006] beschrieben. Neben zwei Beschleunigungssensoren zur Erkennung eines Sturzes enthält das System ein Modul zur Auswertung der empfangenen Signalstärke. Für erste Untersuchungen wurden in einem Raum drei Signalempfänger installiert. Die am Körper des Nutzers getragene Einheit enthält neben einem Beschleunigungsmesser zur Sturzerkennung auch einen Hochfrequenzsender. Dieser sendet periodisch Signale, die von den Empfangseinheiten im Raum erfasst werden. Je größer der Abstand zwischen Sender und Empfänger, umso mehr wird die Signalstärke gedämpft. Über Triangulation wird aus der jeweils empfangenen Signalstärke die Person lokalisiert. Dabei müssen die Koordinaten der Empfangsknoten bekannt sein, um die Position im Raum bestimmen zu können [TABAR et al., 2006].

Auch in [KALTIOKALLIO et al., 2012] wird über die Auswertung der empfangenen Signalstärke die Position im Raum bestimmt. In einer Testwohnung wurden dafür 33 Sendemodule zu einem Wireless-Netzwerk zusammengeführt. Die Module senden Datenpakete im Hochfrequenzbereich ($2,4\text{ GHz}$), die von einem Empfänger erfasst werden. Über eine Identifikationsnummer können sie identifiziert und dem jeweiligen Sender zugeordnet werden. Befindet sich eine Person in der Wohnung, wird die Ausbreitung der Hochfrequenzsignale beeinflusst. Eine Person, die sich in der Nähe eines Senders befindet, behindert die Ausbreitung des gesendeten Signals, wodurch das Signal stark gedämpft empfangen wird. Auf diese Weise lässt sich aus der Dämpfung der Signale auf die Position der Person in der Wohnung schließen. Damit die Messung nicht nur von einer initialen Kalibrierung abhängt und über die Zeit ungenau wird, wurde ein Verfahren vorgestellt, welches eine Nachkalibrierung durchführt, sobald das System erkannt hat, dass sich der Nutzer aus dem jeweiligen Bereich entfernt hat [KALTIOKALLIO et al., 2012].

Ein ähnliches Verfahren nutzen [BHARADWAJ et al., 2014]. Hierbei werden Ultra-Breitband-Antennen in der Kleidung des Nutzers befestigt. Jede Antenne sendet ein Signal, das von einer Empfangseinheit erfasst wird. Durch Auswertung der Zeiten, die die Signale zur Empfangsstation benötigten, kann auf deren Position geschlossen

werden. Die Genauigkeit in der Positionsbestimmung liegt damit bei 3 *cm*. Nicht nur die Position der Person im Raum sollte auf diese Weise ermittelt werden. Durch Nutzung mehrerer Sendeantennen, verteilt am Oberkörper und den Armen des Probanden, kann, aufgrund der hohen räumlichen Auflösung, auch die Lage von Oberkörper und Armen zueinander bestimmt werden [BHARADWAJ et al., 2014].

2.3 Alarmierungskonzept

Nachdem ein Notfall erkannt und die Position bestimmt wurde, an der sich die Person zu diesem Zeitpunkt befindet, muss der Notfall aus der Wohnung nach außen signalisiert werden. Die Alarmierung darf jedoch nicht ungefiltert geschehen, da zu häufige Fehlalarme Kapazitäten bei Pflegepersonal und anderen alarmierten Stellen binden. Fehlalarme führen dadurch zu hohen Kosten (z.B. durch das Aussenden von Rettungswagen) und zur Verunsicherung des Nutzers [RODNER et al., 2011]. Das Ziel des Konzeptes ist es, zuverlässig zu alarmieren, wenn ein Notfall vorliegt, aber auch Fehlalarme zu erkennen und in diesen Fällen die Alarmierung zu verhindern.

Zu diesem Zweck stellen [RODNER et al., 2011] ein mehrstufiges Alarmierungskonzept vor. Während des normalen Betriebs befindet sich das System in einem neutralen Normalzustand. Die Erfassung einer potenziellen Notsituation bringt das System in einen vorläufigen Alarmierungszustand, bei dem überprüft werden soll, ob es sich bei der vorliegenden Situation um einen Notfall oder einen Fehlalarm handelt. Wird ein potentieller Alarm als Fehlalarm bestätigt, gelangt das System in den Normalzustand zurück. Andernfalls wird der potentielle Alarm als tatsächlicher Alarm angesehen, wodurch das System in den Alarmzustand gelangt und Hilfe verständigt wird. Der prinzipielle Aufbau dieses mehrstufigen Alarmierungskonzept ist in Abbildung 2.1 dargestellt.

In der konkreten Anwendung wird zur Überprüfung von Fehlalarmen zweimal versucht in der Wohnung des Nutzers anzurufen, sobald eine potentielle Notsituation vorliegt. Von einem Notfall wird ausgegangen, wenn die von Bewegungsmeldern erfassten Bewegungen in der Wohnung des Nutzers für einen unzulässig langen Zeitraum ausblieben. Auch das manuelle Betätigen einer Schaltfläche auf einem tragbaren End-

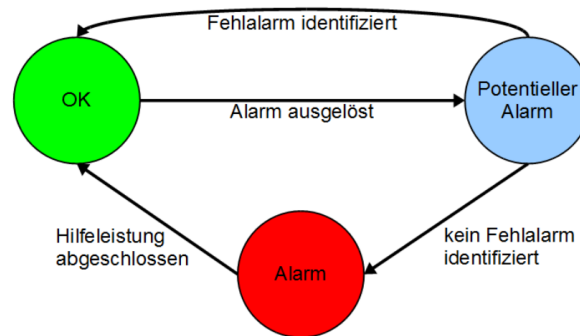


Abbildung 2.1: Das mehrstufige Alarmierungskonzept befindet sich im Normalzustand „OK“. Durch das Vorliegen eines potentiellen Notfalles findet der Übergang in den Zustand „Potenzieller Alarm“ statt. Hier wird überprüft, ob ein Fehlalarm vorliegt. Ist dies der Fall wird die Alarmierung beendet, andernfalls gelangt das System in den Zustand „Alarm“ wodurch die Alarmierung von Rettungskräften stattfindet [RODNER et al., 2011].

gerät erzeugt einen solchen potenzielle Alarm. Als Reaktion auf diesen Alarm erfolgt ein automatischer Anruf in der Wohnung des Nutzers. Eine Ansage weist den Nutzer auf die vorliegende Alarmierung hin. Der Nutzer hat nun die Möglichkeit durch Drücken einer Schaltfläche auf dem Endgerät den Fehlalarm zu bestätigen. Unterbleibt diese Rückmeldung für eine gewisse Zeit, gelangt das System in die nächste Stufe der Überprüfung. Der Alarm gelangt zu einem Mitarbeiter eines Wohlfahrtsverbandes, der ebenfalls versucht, den Bewohner durch einen Anruf in seiner Wohnung zu erreichen. Durch diesen Anruf kann der Zustand des Bewohners erfragt werden und passende Hilfe eingeleitet werden [RODNER et al., 2011].

Auch [PFLÜGER et al., 2012] nutzen ein mehrstufiges Alarmierungssystem, um Fehlalarme zu vermeiden. Die Autoren bezeichnen es als „Eskalierbare Alarmierung“, womit ein stufenweise ansteigendes Alarmkonzept gemeint ist. Ein Alarm wird in diesem Konzept durch die Auswertung von Kamerabildern erzeugt. In der betrachteten Szene wird die Person segmentiert. Anhand der Form des Körpers werden die Körperachse und der Schwerpunkt bestimmt. Ändert sich durch einen Sturz die Lage der Körperachsen oder die Position des Schwerpunktes, wird von einem Notfall ausgegangen. Zusätzlich erfolgt eine akustische Auswertung der Szene auf Basis von Hidden-Markov-

Modellen, durch die Hilfeschreie erkannt werden sollen. Im Ergebnis findet bei Verdacht einer Notsituation ebenfalls ein automatischer Kontrollanruf in der Wohnung des Nutzers statt. Ist die Person nicht in der Lage, den Anruf zu beantworten, nimmt das System eine hohe Notfallwahrscheinlichkeit an, in deren Folge eine Liste von hinterlegten Telefonnummern (z.B. von Angehörigen oder Pflegediensten) alarmiert wird [PFLÜGER et al., 2012].

Das System zur Notfallbehandlung von [TABAR et al., 2006] setzt ebenfalls auf eine Rückkopplung des Alarms, damit der Nutzer das Senden von Fehlalarmen vermeiden kann. Ein Sturz wird hierbei durch einen Beschleunigungssensor erkannt, der vom Nutzer am Körper getragen wird. Zusätzlich enthält das die Erfassungseinheit neben dem Beschleunigungssensor ein Modul zur Sprachübertragung. Damit kann eine bidirektionale Verbindung zwischen Nutzer und Pflegedienststelle hergestellt werden. Nachdem ein Sturz erkannt wurde, wird diese Verbindung aufgebaut. Der Nutzer hat vorher die Möglichkeit, den Anruf abbrechen. Erfolgt kein Abbruch, kann das Pflegepersonal die vorliegende Situation erfragen und weitere Maßnahmen einleiten. Um die Situation besser beurteilen zu können, werden über die Telefonverbindung Daten der installierten Kamera übertragen, die sich in dem Raum befindet, in dem sich der Sturz ereignet hat [TABAR et al., 2006].

Kapitel 3

Aufbau und Funktion des MIRA-Frameworks

In der Robotik wächst mit der Zahl der Fähigkeiten und der Funktionen, die ein Roboter haben soll, die Komplexität der Software, die diese Funktionalitäten bereitstellt. Damit die Wartung und Überarbeitung sowie die Integration neuer Funktionen in bestehende Software auch mit wachsender Komplexität einfach bleibt, wird das Ziel verfolgt, die Software möglichst in einzelne, kleine und überschaubare Module aufzuteilen. Dabei übernimmt jedes Modul jeweils eine konkrete Aufgabe. Beispielsweise kann ein Modul die Algorithmen zur Pfadplanung enthalten, während ein anderes Modul die Personenerkennung übernimmt. Diese Modularität ermöglicht den einfachen Austausch von Modulen in bestehenden Anwendungen und erhöht die Wiederverwertbarkeit erstellter Algorithmen. Dadurch lassen sich die Funktionalitäten komplexer Robotikapplikationen durch Zusammenführung verschiedener bestehender Module erzeugen. Dabei sorgt ein Framework dafür, dass die notwendigen Module in einen Gesamtprozess integriert werden und über verschiedene Mechanismen miteinander kommunizieren können. Für diesen Zweck wurde von der TU Ilmenau in Zusammenarbeit mit der MetraLabs GmbH die Middleware for Robotic Applications¹ (MIRA) entwickelt. Die einzelnen Module lassen sich nach Komplexität und Aufgabe in verschiedene hierarchisch geordnete Schichten (Layer) unterteilen. Abbildung 3.1 zeigt diese hierarchische Struktur.

¹<http://www.mira-project.org/MIRA-doc/>

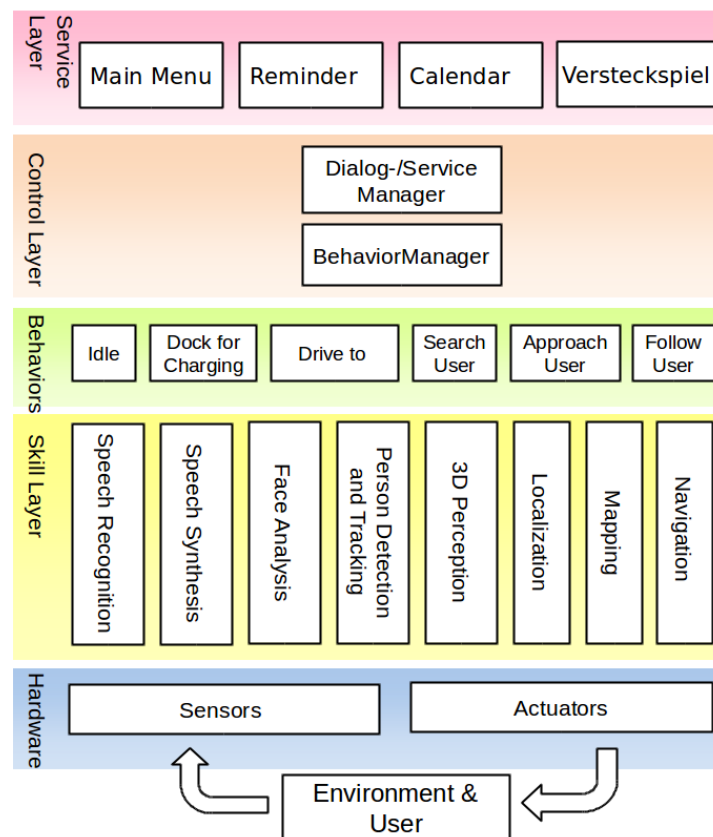


Abbildung 3.1: Module können nach Art und Verwendung in die einzelnen Schichten (Layer) der Applikationsarchitektur geordnet werden. Module des Skill Layers stellen einzelne Funktionalitäten dar, während in den Behaviors diese Funktionalitäten zu komplexeren Anwendungen zusammen geführt werden [MÜLLER, 2016, 67].

Die oberste Schicht enthält die Services (Service Layer), die aus einer oder mehreren Konfigurationsdateien (Frames) bestehen. In den Frames wird die Reihenfolge festgelegt, in der die Behaviors abgearbeitet werden. Dies erfolgt in Abhängigkeit von Events. Dazu zählen Sensorinformationen aus der Umwelt genauso wie die Änderungen interner Zustände. Die Events gelangen über sogenannte Slots zu den entsprechenden Frames des Services. In den Frames wird definiert, welche Aktionen ausgeführt werden, wenn neue Events in den Slots zur Verfügung stehen. Beispielsweise können Events der Slots Bedingungen sein, um Module zu starten oder Bildschirm- bzw. Sprachausgaben zu erzeugen. Auch Timeouts können genutzt werden, um Aktionen auszulösen.

Die Steuerung der Services geschieht durch den Dialog-/ServiceManager. Er sorgt dafür, dass Services nacheinander aktiviert und deaktiviert werden. Auch kann er aktuell laufende Services pausieren oder abbrechen, wenn die Abarbeitung eines neuen Services eine höhere Priorität hat. Das ist beispielsweise der Fall, wenn der Ladezustand kritisch ist und der Roboter zur Ladestation fahren soll. Auf diese Weise kommunizieren die Services nicht direkt miteinander, können aber auf andere Services verweisen, die daraufhin vom ServiceManager gestartet werden. Die einzelnen Funktionen eines Services werden durch die Behaviors bestimmt. Ein Behavior ist eine konkrete Aufgabe, die ein Roboter ausführen soll. Eine solche Aufgabe kann das Suchen eines Nutzers (*Drive to*), das Folgen eines Nutzers (*Follow User*) oder das Warten auf ein Event (*Idle*) sein. Aus diesen Behaviors lassen sich Services wie z.B. das Versteckspiel in Abbildung 3.1 erzeugen. So wie der Ablauf der Services über den ServiceManager gesteuert wird, werden die Behaviors vom BehaviorManager kontrolliert. Dabei meldet der aktivierte Service, welches Behavior vom BehaviorManager gestartet werden soll. Dieser beendet daraufhin laufende Behaviors und verhindert somit, dass verschiedene Behaviors gleichzeitig den Roboter bewegen wollen und damit Konflikte in der Navigation auslösen. Da der BehaviorManager die Behaviors exklusiv aktiviert und nur das aktive Behavior auf die Navigation zugreifen darf, können Konflikte ausgeschlossen werden. Es ist deshalb sinnvoll, jedes Modul, welches die Navigation des Roboters nutzt, als Behavior zu implementieren.

Neben den Behaviors gibt es die Schicht der Skills. Hierbei handelt es sich um Basisfunktionalitäten des Roboters. Sie können genutzt werden, um in Kombination mit anderen Skills neue Behaviors zu erzeugen. Skills können entweder indirekt über die Behaviors gesteuert werden oder sie laufen parallel im Hintergrund. So kann ein Skill im Hintergrund aktiv sein und aus Sensorinformationen der Umwelt Events generieren, die den Ablauf eines Services beeinflussen oder den Behaviors zur Verfügung stehen. Beispielsweise ist ein Modul zur Erzeugung und Aktualisierung von Umgebungskarten ebenso ein Skill, wie ein Modul zur Personendetektion.

In MIRA werden Skills als Units oder MicroUnits realisiert. Die Entscheidung, ob ein Modul als Unit oder MicroUnit implementiert wird, hängt davon ab, ob das Modul

reaktiv angelegt werden soll (MicroUnit), d.h. ob es gestartet werden soll, sobald ein neues Event vorliegt oder ob es sich periodisch selbst starten soll (Unit). Eine Unit kann regelmäßig eigene Berechnungen durchführen, um damit neue Events zu erzeugen.

Die Kommunikation zwischen verschiedenen Units bzw. MicroUnits kann in MIRA auf zwei Wegen erfolgen: Entweder über Channels oder über Remote Procedure Calls (RPC). Channels sind Strukturen, auf die Daten geschrieben und von denen Daten gelesen werden können. Damit ein Modul Daten von einem Channel lesen kann, muss es dafür auf dem Channel zum Lesen registriert werden (Subscribe auf einen Channel). Soll ein Modul Daten auf einen Channel schreiben, so muss dies ebenfalls dem Channel mitgeteilt werden (Publish auf einen Channel). Auf dem Channel liegen die mit einem Zeitstempel versehenen Daten vor. Dadurch können sie zugeordnet werden, auch wenn sie erst zu einem späteren Zeitpunkt abgerufen werden. Durch das Konzept der Channels können Daten unterschiedlichen Modulen zur gleichen Zeit bereitgestellt werden.

Während beim Konzept der Channels viele Module von einem Channel lesen oder auf diesen schreiben, erfolgt eine direkte Kommunikation zwischen zwei Modulen via RPC. Dabei wird die Methode eines Moduls durch ein anderes Modul nach dem request-reply-Prinzip aufgerufen. Ein Zugriff erfolgt über die *callService()*-Methode aus dem aufrufenden Modul. Zusätzlich können auf diesem Weg auch Parameter in die aufgerufene Methode übergeben werden. In Abbildung 3.2 sind die beiden Wege der Kommunikation zwischen Units bzw. MicroUnits dargestellt.

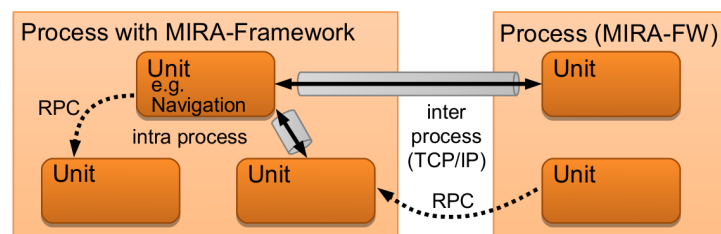


Abbildung 3.2: Die Kommunikation zwischen zwei Units kann über Channels oder Remote Procedure Calls (RPC) erfolgen. Dabei können die Units auch zu unterschiedlichen Prozessen gehören [EINHORN und LANGNER, 2012, 2593].

Kapitel 4

Roboterplattform Tweety

Roboter können unter anderem dabei helfen, Senioren möglichst lange ein selbstständiges Leben in ihren eigenen Wohnungen zu ermöglichen. Dabei lassen sich zwei Arten von Robotern unterscheiden. Auf der einen Seite stehen die Service-Roboter. Sie unterstützen die Senioren in ihren häuslichen Umgebungen, indem sie ihnen Aufgaben bei der Bewältigung ihres Haushalts abnehmen. Ein bekanntes Beispiel hierfür sind Roboter, die autonom in der Wohnung staubsaugen. Auf der anderen Seite existieren Konzepte, bei denen nicht die Ausführung von Services im Vordergrund steht. Diese Roboter sollen die Lebensqualität dadurch verbessern, dass sie anwesend sind, berührt und gestreichelt werden können und durch Mimik oder akustische Ausgaben Emotionen ausdrücken. Besonders bei allein lebenden Menschen soll diese Art von Robotern dazu führen, dass sich die Bewohner nicht alleine fühlen und etwas haben, um das sie sich kümmern bzw. mit dem sie sich beschäftigen können. Diese Roboter nehmen, ähnlich wie Haustiere, die Rolle von Lebensgefährten (Companion) ein und werden daher als Companion-Roboter bezeichnet [ROBINSON et al., 2014].

Im Fachgebiet für Neuroinformatik und kognitive Robotik der TU Ilmenau wurde im Rahmen des Projekts „SERvice-RObotik für die GesundheitsAssistenz“ (SERROGA, 2012-2015) und des Vorgängerprojektes „CompanionAble“ (2008-2012) der Roboter „Tweety“ entwickelt. Abbildung 4.1 zeigt den baugleichen Roboter „Max“. Ziel der Entwicklung war es, die Vorteile eines Service- und eines Companion-Roboters zu vereinen. Der in diesem Projekt entstandene Roboter bewegt sich autonom durch die

Wohnung und ist dadurch in der Lage, den Bewohner aktiv aufzusuchen, um seine Services anzubieten. Die Funktionen umfassen das Bereitstellen von Kommunikationsmöglichkeiten (z.B. Videotelefonie), das Erinnern an Termine oder an die Medikamenteneinnahme sowie das miteinander Agieren als Motivator zur körperlichen Aktivität. Tweety basiert auf der Roboterplattform SCITOS G3. Der 120 *cm* große Roboter besitzt als Antrieb ein Differential Drive, bei dem die beiden Räder sowohl unabhängig voneinander als auch entgegengesetzt bewegt werden können. Dadurch ist das Drehen auf der Stelle möglich. Zusammen mit der geringen Grundfläche von ca. 50 *cm* kann er selbst durch enge Bereiche gut manövrieren. Mit einer Akkuladung ist er in der Lage bis zu 10 *h* autonom in der Wohnung zu operieren, bis er aufgeladen werden muss. Dabei gehört das selbstständige Aufsuchen seiner Ladestation bei geringer Akkukapazität zu einer seiner Basisfunktionen. Seine Umgebung nimmt der Roboter über einen Laserscanner, eine 180°-Farbkamera sowie über zwei unterschiedlich geneigte Asus-Tiefenkameras wahr. Auf Basis der Sensorinformation der Tiefenkamera findet die Erkennung liegender Personen statt (vgl. Kap. 5.2.1). Über die in Fahrtrichtung ausgerichtete 180°-Kamera können stehende Personen erkannt werden (vgl. Kap. 5.2.3). Die schriftsprachliche Interaktion mit dem Roboter geschieht vordergründig über ein großes, neigbares Touchdisplay, das sich an der Front des Roboters befindet. Je nachdem, ob die Interaktion mit einer sitzenden oder stehenden Person geschieht, kann das Display angekippt werden, um dem Benutzer die Bedienung zu erleichtern. Auf der verbal auditiven Ebene kommuniziert der Roboter über eingebaute Lautsprecher. Dabei können Audiodateien situationsabhängig ausgegeben werden. Die Erzeugung der Audiodateien erfolgt mithilfe eines Sprachsyntheseprogramms. Durch die Wahl der Tonhöhe sowie der Geschwindigkeit können zusätzlich zum Inhalt der Sätze auch Emotionen übermittelt werden. Die Möglichkeit, akustisch über die internen Zustände des Roboters informieren zu können, vereinfacht die Verständigung mit dem Nutzer. Weitere Möglichkeiten der Interaktion bieten die Berührungssensoren, die zum einen am Gehäuse des Roboters angebracht sind (vgl. Abb. 4.1) und sich zum anderen in einem Kunstfell am Kopf des Roboters befinden. Über dieses sogenannte „Streichelfell“ soll der Nutzer animiert werden, den Roboter zu berühren, um dadurch eine emotionale Verbindung zu ihm herzustellen. Dabei registriert der Roboter die Art der

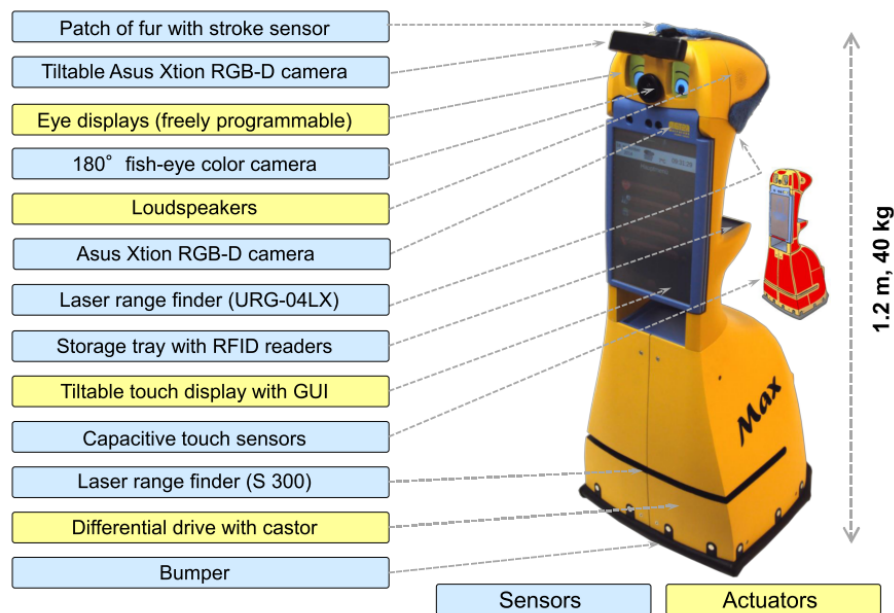


Abbildung 4.1: Roboter „Max“ mit seinen Sensoren und Aktuatoren. Die baugleichen Roboter „Max“ und „Tweety“ basieren auf der Roboterplattform SCITOS G3. Die roten Bereiche zeigen die Positionen der kapazitiven Berührungssensoren. Zur Personenerkennung werden die geeignete Asus Xtion RGB-D Kamera sowie die 180°-Farbkamera verwendet [GROSS et al., 2015, 5993].

Berührung und kann beispielsweise Streicheln, Kitzeln oder einen Schlag unterscheiden. Entsprechend der Berührung reagiert er mit Schnurren, Lachen oder Schreien. Zusätzlich kann die Art der Berührung genutzt werden, um vom Roboter ausgeführte Aktionen zu bewerten. Durch dieses Feedback passt der Roboter sein Verhalten mit der Zeit den Wünschen des Nutzers an.

Als weitere Form der Kommunikation kann der Roboter seine animierten Augen nutzen. Dadurch bekommt er eine Erscheinung, die dafür sorgt, vom Nutzer als Interaktionspartner akzeptiert zu werden. Über die Augen können Emotionen transportiert und über aktuelle interne Zustände des Roboters informiert werden. So signalisieren offene Augen, dass Interaktionen möglich und gewünscht sind, während geschlossene Augen anzeigen, dass Funktionen im Moment nicht bereitstehen, z.B. wenn sich der Roboter an der Ladestation befindet. Auf diese Weise stehen dem Roboter auch nonverbale

Kommunikationsmöglichkeit zur Verfügung.

Abbildung 4.1 zeigt den ebenfalls auf der Roboterplattform SCITOS G3 basierenden Roboter „Max“. Der in dieser Arbeit verwendete „Tweety“ gleicht jedoch sowohl in der Ausstattung wie auch in Form und Farbe dem abgebildeten Roboter.

Ausführlichere Beschreibungen zu den verwendeten Roboterplattformen und ihren Funktionalitäten, die im *SERROGA*-Projekt bzw. in dessen Vorgängerprojekt *CompanionAble* (2008-2012) entstanden, lassen sich in [GROSS et al., 2015], [GROSS et al., 2012] und [GROSS et al., 2011] finden.

Kapitel 5

Implementierung einer Strategie zur Notfallbehandlung

Um Senioren in ihrem Wohnumfeld zu unterstützen, werden in Zukunft häufiger mobile Roboterplattformen eingesetzt. Deshalb ist es sinnvoll, wenn der Roboter neben seinen hauptsächlichen Serviceaufgaben auch eine Strategie zur Erkennung und Behandlung von Notfällen besitzt. Ein Service, welcher diese Aufgabe übernimmt, soll dabei zusammen mit allen weiteren Anwendungen starten, sich jedoch, solange keine Abweichungen in den Alltagsroutinen des Bewohners auftreten, die auf einen Notfall hindeuten können, vom Nutzer unbemerkt im Hintergrund der laufenden Anwendung befinden. Erst wenn Komponenten des Services einen möglichen Notfall detektieren, muss das Modul in den Vordergrund treten und die Kontrolle über das Roboterverhalten erlangen. Dabei müssen derzeit laufende Prozesse abgebrochen oder pausiert werden, damit modulspezifische Aufgaben des Notfallservice wie z.B. Navigation oder Verifikation durchgeführt werden können. Es bietet sich daher an, das Modul zur Notfallbehandlung als einen Service zu implementieren. Wie im Kapitel 3 beschrieben, kann ein Service verschiedene Units und Behaviors enthalten, wobei letztere durch interne oder externe Events gestartet und gestoppt werden können. Gesteuert werden diese Vorgänge durch den BEHAVIORMANAGER.

Die Erzeugung eines geeigneten Ablaufs zur Notfallbehandlung beinhaltet die Integration von Modulen, die jeweils unterschiedliche Aufgaben erfüllen. Neben Modulen, die

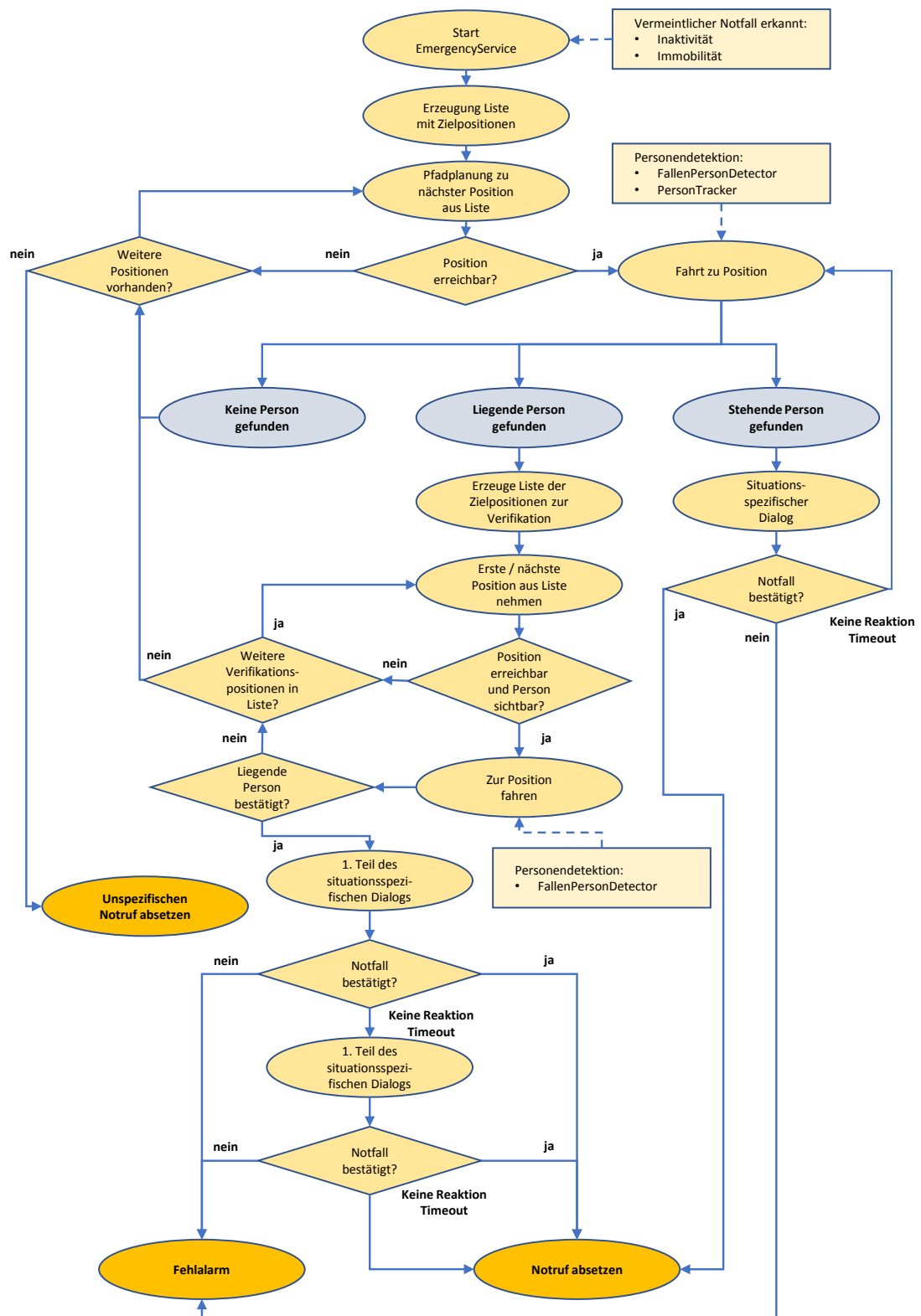


Abbildung 5.1: Programmablaufplan des Services zur Notfallbehandlung

im Rahmen dieser Arbeit entwickelt werden sollen, existieren bereits einige Module, die genutzt und in den Gesamttablauf integriert werden können. Dazu gehören die Module zur Detektion stehender und liegender Personen sowie das der Navigation einschließlich der Pfadplanung und der Motorsteuerung. Da diese Module Fähigkeiten beschreiben, die der Roboter besitzt und die von verschiedenen Anwendungen verwendet werden können, bilden sie die SKILLS des Roboters. In Kapitel 5.2 werden sie vorgestellt.

Den Kern dieser Arbeit bildet die Entwicklung und Integration von Modulen zur Erzeugung eines Services zur Notfallbehandlung. Die Beschreibung des allgemeinen Aufbaus des Services erfolgt in Kapitel 5.1. In den anschließenden Kapiteln wird die Konzeption und Entwicklung der Module vorgestellt, die im Rahmen der vorliegenden Arbeit entwickelt wurden. Die Reihenfolge der Kapitel folgt dabei dem Ablauf der Notfallbehandlung. Enthalten ist die *Notfallerkennung* (Kapitel 5.3), die *Personensuche* (Kapitel 5.4), die *Verifikation liegender Personen* (Kapitel 5.5) und abschließend der *Dialog* (Kapitel 5.6). Die Gesamtheit aller Module zur Erkennung und Behandlung von Notfällen in häuslicher Umgebung soll im Folgenden als EMERGENCYSERVICE bezeichnet werden.

5.1 Aufbau eines Services zur Notfallbehandlung

Der EMERGENCYSERVICE startet mit der Erkennung eines Notfalls. Hierfür sollen PIR-Sensoren (*passive infrared*, Bewegungsmelder) genutzt werden, die die Bewegungen von Personen erfassen. Alle Bewegungsevents werden von einer Unit der Notfallbehandlung registriert und ausgewertet. Werden untypisch lange Inaktivitätsphasen bemerkt, soll das System in Alarmbereitschaft versetzt werden, woraufhin die Personensuche und alle daran anschließenden Abläufe gestartet werden. Abbildung 5.1 zeigt den allgemeinen Ablauf der Notfallbehandlung. Wie aus der Abbildung ersichtlich, startet sowohl eine untypisch lange Inaktivität, wie auch eine zu lange Immobilität die Notfallbehandlung. Daraufhin wird eine Liste geeigneter Suchziele erstellt, die nacheinander angefahren werden sollen, bis die Person gefunden wurde. Wird während der Suchfahrt die Person gefunden, wird die Fahrt beendet und Kontakt zu der Person

aufgenommen. Dabei soll unterschieden werden, ob es sich um eine stehende oder liegende Person handelt. Im Falle einer stehenden Person tritt der Roboter sofort mit der Person in Kontakt, um herauszufinden, ob ein Notfall vorliegt und Hilfe benötigt wird.

Wird hingegen eine liegende Person entdeckt, muss der Roboter zuvor verifizieren, ob es sich mit hoher Wahrscheinlichkeit auch tatsächlich um eine liegende Person handelt oder ob ein Gegenstand fälschlicherweise als Person klassifiziert wurde. Diese zusätzliche Überprüfung ist notwendig, da der Roboter während seiner Suchfahrt schon geringe Hypothesen für das Vorliegen einer Person akzeptiert. Damit soll sichergestellt werden, dass eine Person nicht durch den Roboter übersehen werden kann.

Je nach Situation kann es vorkommen, dass eine hilfsbedürftige Person in einer ungünstigen Position vor dem Roboter liegt, während sich dieser annähert. In diesem Fall erzeugt das Modul zur Erkennung liegender Personen (`FALLENPERSONDETECTOR`) eine Personenhypothese mit geringer Wahrscheinlichkeit (*Score*) für das Vorhandensein einer liegenden Person. Während die Schwelle zur Erkennung liegender Personen bei der Suchfahrt absichtlich niedrig gewählt wird ($Score > 0,3$), um die Sensitivität zu erhöhen und damit schon bei geringer Wahrscheinlichkeit die Suchfahrt unterbrochen wird, soll das Vorhandensein einer Person zuerst mit hoher Wahrscheinlichkeit ($Score > 0,9$) durch das Verifikationsmodul bestätigt sein, bevor der Roboter weitere Maßnahmen ergreift.

Konnte eine liegende bzw. stehende Person gefunden und bestätigt werden, startet der Roboter den situationsabhängigen Dialog, der so konzipiert ist, dass der Nutzer die Möglichkeit hat, einen Notfall zu bestätigen. Der Roboter kann darauf reagieren, indem er z.B. einen Notruf absetzt. Jedoch muss der Nutzer auch die Möglichkeit haben, das Absetzen des Notrufes zu verhindern, um Fehlalarme zu vermeiden. Wird ein Notfall von dem Nutzer bestätigt, können Alarmierungsmechanismen dafür sorgen, dass beispielsweise Angehörige oder Rettungsstellen über den vorliegenden Notfall informiert werden.

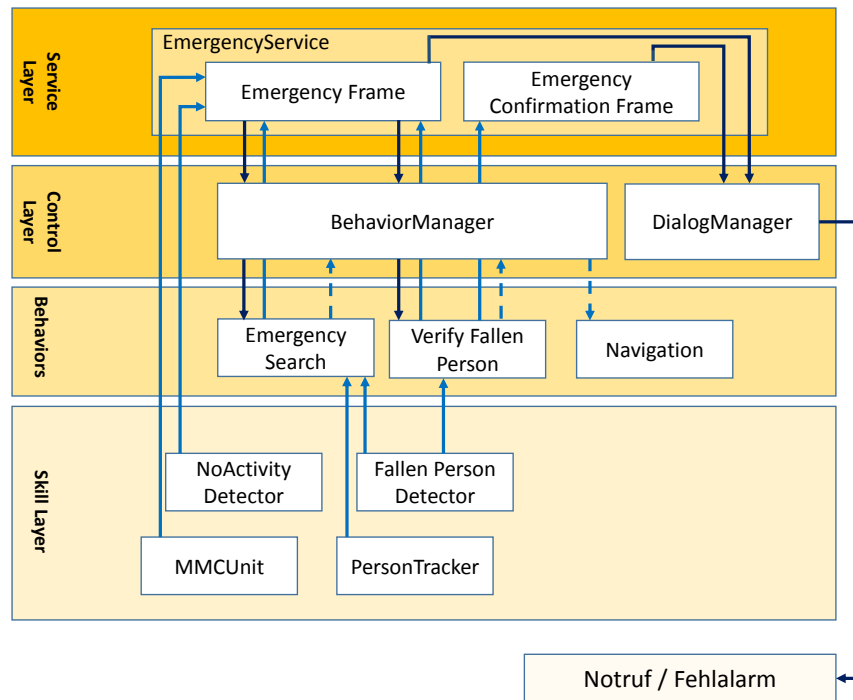


Abbildung 5.2: Das Schichtenmodell zeigt die Struktur des Service zur Notfallbehandlung zusammen mit den Wegen, auf denen Behaviors aktiviert (dunkelblaue Pfeile) und Informationen in Form von Events transportiert werden (hellblaue Pfeile). Die Behaviors nutzen die Navigation (gestrichelte Pfeile). Der BehaviorManager steuert die Aktivierung der einzelnen Behaviors.

Die Gestaltung eines Services zur Notfallbehandlung und die sich daraus ergebende Interaktion der Behaviors und Units ist in Abbildung 5.2 dargestellt.

Um auf vorliegende Notfälle schließen zu können, werden die Bewegungen des Nutzers durch PIR-Sensoren erfasst und ausgewertet. Konnten untypisch lange Phasen von Inaktivität bzw. Immobilität in der Wohnung registriert werden, wird dies durch die Erzeugung entsprechender Events der Skills NOACTIVITYDETECTOR (vgl. Kap. 5.3.1) bzw. MMCUNIT (vgl. Kap. 5.3.2) signalisiert, wodurch der erste Frame des EMERGENCYSERVICE aktiviert wird. Dieser startet die Ausführung des EMERGENCYSEARCH-Behaviors (vgl. Kap. 5.4) zu Suche der Person, wobei die Aktivierung durch den BehaviorManager geschieht, der die Koordination der verschiedenen Behaviors übernimmt.

Im EMERGENCYSEARCH-Behavior wird die Reihenfolge festgelegt, in der Zielpositionen innerhalb der Wohnung angefahren werden sollen. Der Roboter fährt daraufhin die einzelnen Bereiche ab, bis eine stehende oder liegende Person gefunden wurde. Dazu werden die Informationen analysiert, die aus dem PERSONTRACKER (vgl. Kap. 5.2.3) bzw. FALLENPERSONDETECTOR (vgl. Kap. 5.2.1) zum EMERGENCYSEARCH-Behavior gelangen. Konnte eine stehende oder liegende Person gefunden werden, wird dies über die Events *StandingPersonDetected* bzw. *FallenPersonDetected* dem EMERGENCYFRAME mitgeteilt.

Im Falle einer stehenden Person startet der Frame mithilfe des DialogManagers den Nutzerdialog, um die Person anzusprechen und zu klären, ob ein Notfall vorliegt. Wurde hingegen eine liegende Person erkannt, wird das VERIFYFALLENPERSON-Behavior (vgl. Kap. 5.5) zur Verifikation der Personenhypothese gestartet. Die Ausführung wird durch den BehaviorManager gesteuert, der zuvor das EMERGENCYSEARCH-Behavior beendet, um Konflikte zwischen den Behaviors zu verhindern. Die Verifikation nutzt ebenfalls den FALLENPERSONDETECTOR. Eine sichere Hypothese für das Vorliegen einer liegenden Person aktiviert den EMERGENCYCONFIRMATIONFRAME, der den Dialog zur Ansprache einer liegenden Person steuert. Konnte hingegen die Hypothese einer liegenden Person nicht bestätigt werden, wird mit der Suche in Wohnung fortgefahren. Dazu wird das EMERGENCYSEARCH-Behavior erneut gestartet, um die Fahrt zu den verbleibenden Zielpositionen zu steuern.

Der Dialog (vgl. Kap. 5.6) dient der Klärung des vorliegenden Sachverhalts. An seinem Ende steht die Information, ob es sich um einen Notfall handelt oder ob ein Fehlalarm ausgelöst wurde und keine Hilfe benötigt wird.

Neben den bereits im Zusammenhang mit der Abbildung 5.2 genannten Events, die in den Skills erzeugt werden und über Channels die vermeintlichen Notfälle oder das Vorhandensein von Personen signalisieren und damit direkt die Aktivierung der Behaviors beeinflussen, gibt es noch weitere Channels, über die interne Zustände des Systems repräsentiert werden.

So dienen die auf dem MERGEDMAP- und GLOBALMAP-Channel liegenden Karten dazu, mögliche Zielpositionen auf Befahrbarkeit zu prüfen, während der Channel MI-

LEAGE über den vom Roboter zurückgelegten Weg informiert. Diese internen Zustände beeinflussen den Ablauf der Services nicht direkt, werden jedoch benötigt, um die inneren Ablauf zu steuern.

5.2 Genutzte Roboterskills

Zum Gesamtablauf einer effektiven Notfallbehandlung gehören die Navigation sowie eine zuverlässige Erkennung von Personen. Die Navigation umfasst die Pfadplanung und die Fahrsteuerung. Zur Personenerkennung gehören die Erkennung von stehenden und liegenden sowie sitzenden Personen, wobei die Erkennung stehender und sitzender Personen durch dasselbe Modul geschieht. Da es sich bei diesen Modulen um Fähigkeiten des Roboters handelt, die ihm, unabhängig vom gewünschten Gesamtverhalten, zur Verfügung stehen, gehören sie zur Gruppe der Roboterskills. Je nach Anwendung können verschiedene Skills miteinander kombiniert werden, um gewünschte Funktionen des Roboters zu erzeugen. Im Folgenden sollen diese, im Rahmen dieser Arbeit genutzten, Skills kurz vorgestellt werden.

5.2.1 Erkennung liegender Personen

Die Erkennung liegender Personen wurde im Rahmen einer Masterarbeit an der TU Ilmenau entwickelt [LEWANDOWSKI, 2016]. Als Input dienen Tiefenbilder aus 3D-Sensoren wie Laser-Entfernungsmesser oder Tiefenkameras. Die erfassten Bilder werden anschließend zu einer dreidimensionalen Gesamtkarte zusammengeführt. Durch die kontinuierliche Erzeugung der Karte können neue Objekte in die Karte integriert werden. Es ist auch möglich, Teile aus der Karte zu entfernen, wenn sie für längere Zeit nicht in der Szene gesehen werden. Die somit gewonnene 3D-Karte ist stets eine aktuelle Darstellung der Umgebung, deren Auflösung durch die gewählte Zellgröße bestimmt wird. Aus den einzelnen NDT-Karten werden im folgenden Schritt Features extrahiert, um Zellen zu finden, die zu einer Person gehören können. Damit die Berechnungszeit sowie die Anzahl der falsch-positiv klassifizierten Zellen verringert wird, wird zuvor die Bodenebene im Bild geschätzt und die Zellen, die zum Boden gehören, aus dem Bild entfernt.

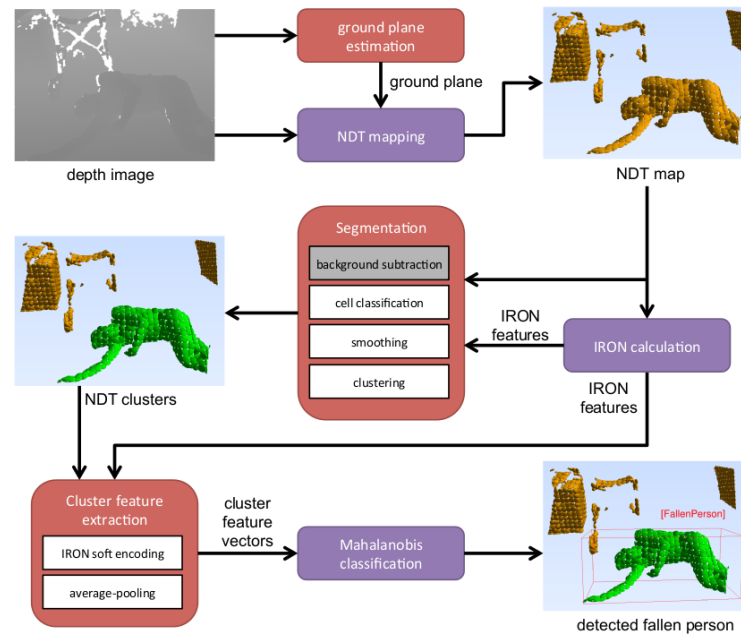


Abbildung 5.3: Die Abbildung zeigt den Ablauf zur Erkennung gestürzter Personen in Bildern. Dazu werden aus den erfassten Tiefenbildern Features zur Klassifikation extrahiert. Das Ergebnis sind BoundingBoxes, die um die Bereiche der gefundenen Personen gebildet werden. Ein Score zu jeder BoundingBox gibt an, mit welcher Wahrscheinlichkeit es sich dabei um eine Person handelt [LEWANDOWSKI, 2016, 52].

Neben den extrahierten Features werden auch die Abstände zu den Zellen der Nachbarschaft für die Klassifikation genutzt. Auf diese Weise entstehen verschiedene Zellcluster, für die jeweils im nächsten Schritt bestimmt wird, ob es sich hierbei um einen liegenden Menschen handelt. Wurde eine liegende Person im aktuellen Bild erkannt, wird eine Box, die sogenannte *Bounding Box*, entsprechend der Größe des Clusters um dieses gebildet. Eine detaillierte Darstellung der Segmentierungs- und Klassifizierungsschritte ist in der Abbildung 5.3 gegeben.

Enthält die Szene mehrere Personen, wird für jede Person eine Bounding Box erzeugt, für die die Koordinaten der Eckpunkte und des Clusterschwerpunktes sowie ein Score für die Wahrscheinlichkeit der Detektion angegeben wird. Diese Informationen bilden eine Personenhypothese. Die einzelnen Hypothesen pro Bild werden erst in einem Hypothesenvektor zusammengefasst und anschließend auf den Channel FALLENPER-

SONHYPOTHESES geschrieben und stehen damit allen Modulen zur Verfügung, die auf sie zugreifen wollen [LEWANDOWSKI, 2016].

5.2.2 Navigation

Die Navigation ist ein wesentlicher Bestandteil in der mobilen Robotik. Ihre Aufgabe ist es, Bewegungskommandos so zu erzeugen, dass sich der Roboter schnell und zielgerichtet von einer Startposition zu einem Ziel bewegt. Dabei müssen dynamische Änderungen in der Umgebung erkannt werden, um Kollisionen mit Objekten und Menschen zu vermeiden. Wird der Roboter im direkten Kontakt mit Menschen eingesetzt, können weitere Einschränkungen wichtig werden, die von der Navigation berücksichtigt werden müssen.

Der Zugriff auf das Navigationsmodul erfolgt von außen durch das Setzen von *Tasks*. Ein Task besteht wiederum aus mehreren Subtasks. Diese sind modular und können teils beliebig kombiniert werden, um ein gewünschtes Navigationsverhalten zu erzeugen. Es lassen sich z.B. die Zielposition (*PositionTask*), die gewünschte Orientierung des Roboters (*OrientationTask*) sowie die bevorzugte Fahrtrichtung (*PreferredDirectionTask*) festlegen. Weitere Subtasks beeinflussen die zulässige Annäherung an Personen, die maximale Geschwindigkeit während der Fahrt und ob einer Person gefolgt werden soll. Bei den am häufigsten verwendeten Subtasks zur Wahl der Zielposition und einer Orientierung können Toleranzen angegeben werden, ab wann das Ziel als erreicht gilt. Das Navigationsverhalten nimmt die Tasks entgegen und führt die entsprechenden Fahrkommandos (einschließlich der Pfadplanung und Kollisionsvermeidung) aus. Zur Kollisionsvermeidung nutzt das Navigationsmodul zum einen die Karte der statischen Hindernisse (*GlobalMap*), zum anderen die Daten der 3D-Sensoren, über die der Roboter verfügt. Damit können dynamische Hindernisse während der Fahrt erkannt werden. Für den Fall, dass ein Hindernis nicht rechtzeitig erkannt wurde, da es beispielsweise außerhalb des Erfassungsbereichs der Sensoren lag, verfügt der Roboter über eine Stoßsensorleiste (Bumper). Beim Auslösen wird die Fahrt sofort gestoppt. Dies geschieht nicht durch Software, sondern durch Hardware, die direkt die Bewegung der Motoren unterbricht.

Über den aktuellen Zustand, in dem sich die Navigation befindet, informieren Events, die auf den `PILOTEVENT`-Channel geschrieben werden. Konnte ein Ziel erfolgreich angefahren werden, wird dies über den Zustand *GoalReached* signalisiert. Der Zustand *PlanAndDrive* zeigt an, dass sich der Roboter noch auf der Fahrt zum Ziel befindet. Befindet sich das Ziel in einem Bereich, der vom Roboter als frei bzw. als nicht belegt angenommen wurde, jedoch durch ein Hindernis nicht erreichbar ist, gelangt der Roboter nach einem Timeout in den Zustand *NoPathPlannable*. Die Auswertung der Zustände wird im Folgenden genutzt, um den Programmablauf zu steuern.

5.2.3 Detektion stehender Personen

Die Behandlung liegender Personen steht in dieser Arbeit im Vordergrund, da bei gestürzten Personen eher von einem Notfall ausgegangen werden muss. Dennoch bildet, neben der Erkennung liegender Personen, die Erkennung stehender Personen eine weitere wichtige Eigenschaft des Roboters, die im Konzept der Strategie zur Behandlung von Notfällen genutzt werden soll. Trifft der Roboter während seiner Suchfahrt auf eine stehende Person, kann meist davon ausgegangen werden, dass kein Notfall vorliegt. Damit Fehlalarme vermieden werden, wird der Dialog genutzt, um zu klären, ob Hilfe benötigt wird oder nicht.

Die Erkennung stehender Personen kann durch viele Faktoren erschwert werden. So können Personen durch Objekte verdeckt sein, sich in unterschiedlichen Abständen zum Roboter befinden oder sich bewegen. Auch schlechte oder wechselnde Beleuchtung kann dazu führen, dass die Personenerkennung misslingt.

Zur Detektion stehender Personen nutzt der Roboter „Tweety“ zwei unterschiedliche Ansätze. Zum einen wird mittels Laserscan kurz über dem Boden nach Abstandsprofilen gesucht, die auf Beine hinweisen. Zum anderen wird eine bildbasierte Detektion mit einer frontal ausgerichteten 180°-Farbkamera verwendet. Dazu werden Histogramms of Oriented Gradients (HOG) genutzt. Diese Prinzip wurde in [DALAL und TRIGGS, 2005] beschrieben. Die Grundidee ist, dass Objekte in Bildern nicht über ihr Aussehen klassifiziert werden, sondern über die Verteilung der Richtungen der Kanten im Bild, die diese Objekte aufweisen. Anhand der entsprechenden Verteilungen können

auf diese Weise auch stehende Personen erkannt werden. Dazu wird das aufgenommene Bild in einzelnen Bereiche unterteilt, für die jeweils ein Histogramm gebildet wird. Aus der Kombination der einzelnen Histogramme entsteht eine Repräsentation des Bildes, welche zur Klassifikation genutzt wird [DALAL und TRIGGS, 2005]. Eine genaue Beschreibung der Funktionsweise des auf dem Roboter verwendeten Personendetektors zur Erkennung stehender bzw. sich bewegender Personen, erfolgt in [VOLKHARDT et al., 2013] und [WENGEFELD et al., 2016].

Durch Nutzung der Farbkamera können Personen erkannt werden, die sich vor dem Roboter in einem Abstand bis zu 4 m befinden. Da der Algorithmus auf die Erkennung von Oberkörpern abzielt, kann er ebenso verwendet werden, um neben stehenden auch sitzende Personen zu erkennen.

Die kamerabasierte Erkennung wird in dieser Arbeit genutzt, um stehende Personen während der Suchfahrt zu erkennen. Für den Fall einer Fehldetektion, muss diese im situationsabhängigem Dialog erkannt werden, damit die Suche zu weiteren Zielen fortgesetzt werden kann.

5.3 Module der Notfallerkennung

Da die Fähigkeit eines Roboters, Notfälle zu erkennen, stets von dem genutzten Sensorkonzept abhängt, wird in dieser Arbeit der Schwerpunkt auf die Auswertung von Bewegungs- bzw. Ruhephasen, wie sie durch PIR-Sensoren erfasst werden können, gelegt. Ein ausgelöster Sensor sendet seine Sensor-ID zusammen mit dem Zeitpunkt der erfassten Bewegung. Damit ist es möglich, die Bewegungen in einer Wohnung nicht nur zeitlich sondern auch räumlich aufzulösen.

Im Folgenden werden zwei Mechanismen zur Notfallerkennung vorgestellt. Dabei sollen zum einen lange Zeiten der Bewegungslosigkeit erkannt werden. Diese Ereignisse können ortsunabhängig sein und sollen im weiteren als Inaktivität (*NoActivity*) bezeichnet werden. Dem gegenüber gibt es Situationen, in denen sich die Nutzer für einen langen Zeitraum nicht aus einem Bereich herausbewegen, jedoch bei Bewusstsein sind. So können sich Personen nach einem Sturz zwar bewegen, jedoch nicht alleine aufstehen

und deshalb einen Bereich für eine untypisch lange Zeit nicht verlassen. Der zuvor beschriebene Alarmierungsmechanismus bei Inaktivität würde in diesem Fall mit jeder erfassten Bewegung von normalen Nutzerbewegungen ausgehen und keinen Alarm auslösen. Die Ereignisse, bei denen Bewegungen erfolgen, die Mobilität zwischen verschiedenen Bereichen aber für eine gewisse Zeit ausbleibt, soll demnach als Immobilität (*Immobility*) bezeichnet werden.

Im Folgenden wird die Erfassung der Inaktivität sowie die der Immobilität erläutert. Es wird dabei davon ausgegangen, dass sich der Nutzer in seiner Wohnung befindet und den typischen Aktivitäten des alltäglichen Lebens nachgeht. Des Weiteren wurde bei der Entwicklung einer Notfallstrategie von einer allein lebenden Person ausgegangen. Die Erzeugung der Events, die auf eine erkannte Inaktivität bzw. Immobilität schließen lassen, geschieht durch die Units `NOACTIVITYDETECTOR` und `MMCUNIT`.

5.3.1 Erzeugung eines NoActivity-Alarms

Bewegt sich ein Bewohner innerhalb seiner Wohnung, so löst diese Bewegung jedes Mal beim Passieren des Erfassungsbereiches eines PIR-Sensors ein Event aus. Auch wenn eine Person sitzt und relativ kleine Bewegungen vollzieht (wie sie z.B. beim Essen oder Zeitunglesen geschehen), führt dies zur Auslösung des Sensors. Solange sich die Person so bewegt, dass ein Sensor ausgelöst wird, kann im Sinne der Inaktivitätsüberwachung davon ausgegangen werden, dass keine Situation vorliegt, in der der Nutzer Hilfe durch den Roboter benötigt. Jedes neu Event setzt einen Timer zurück, der die verstrichene Zeit seit dem letzten Event erfasst. Zusätzlich wird periodisch verglichen, ob der Timer einen zuvor festgelegten Schwellwert überschritten hat. Ist dies der Fall, wird ein entsprechendes Event auf dem NoActivity-Channel generiert.

Dafür wurde die Unit `NOACTIVITYDETECTOR` entwickelt, die die aktuellen Bewegungsevents empfängt. Die Bewegungsevents können anhand ihrer ID den entsprechenden Sensoren zugeordnet werden. Außerdem ist jedes Event mit einem Zeitstempel versehen. Durch die Erfassung eines neuen Events, wird der Inaktivitätszähler zurückgesetzt. Damit die Informationen über die erhaltenen Events nicht verloren gehen und auch für zukünftige Datenanalysen zur Verfügung stehen, werden sie in einer

geeigneten Form mit Uhrzeit, Sensor-ID und Wochentag gespeichert. Zusätzlich werden für jede Stunde des Tages alle Zeiten zwischen aufeinander folgenden Bewegungen gespeichert. Ziel dieser Erfassung ist es, dynamische Schwellen für zulässige Inaktivitäten zu generieren. Diese Schwellen sollen tageszeitabhängig sein und die typischen Aktivitätsmuster der Nutzer widerspiegeln. Durch Optimierung muss dabei ein Kompromiss zwischen einer schnellen Erkennung von Notfällen bei gleichzeitig niedriger Rate an Fehlalarmen gefunden werden. So sollte eine dynamische und tageszeitabhängige Schwelle längere Inaktivitäten während der Nacht tolerieren, lange Inaktivitäten am Vormittag jedoch als untypisch erkennen.

Zur Berechnung einer dynamischen, tageszeitabhängigen Schwelle wird in der vorliegenden Arbeit ein Algorithmus genutzt, der in [CUDDIHY et al., 2007] vorgestellt wurde. Bei diesem Konzept werden ebenfalls Bewegungsmelder genutzt, um Inaktivitäten des Nutzers zu erfassen. Diese werden nach Erfassung den jeweiligen Zeitintervallen des Tages zugeordnet. Dazu wurde der Tag in 48 Intervalle von jeweils 30 Minuten unterteilt. Der Algorithmus erzeugt mit jeder erfassten Bewegung die zulässige Inaktivitätsdauer für den aktuellen Intervall. Die Bestimmung erfolgt auf Basis aller erfassten Inaktivitäten des Intervalls. Die erzeugte Alarmschwelle enthält einen *Uniform Buffer*, der eine Mindestzeit festlegt, bevor ein Alarm ausgelöst wird. Ein *Variable Buffer* erhöht die Empfindlichkeit des Systems während typischer Aktivitätsphasen und verringert sie in Phasen, die durch lange Inaktivitäten gekennzeichnet sind. Um leichte zeitliche Verschiebungen in den Tagesabläufen des Nutzers zu berücksichtigen, wird die aktuelle Alarmschwelle in Abhängigkeit der erfassten maximalen Inaktivitäten der Intervalle vor und nach dem aktuellen Intervall bestimmt [CUDDIHY et al., 2007].

Um die Eignung des Algorithmus für das in dieser Arbeit vorgestellte Konzept zu überprüfen, wurde er als Teil des NOACTIVITYDETECTOR-Moduls implementiert und auf dem Datensatz *Nr. 19, Aruba, Daily Life 2011-2012*¹ getestet. Der Datensatz entstand im Rahmen des *WSU CASAS Smart Home Project* und enthält die Bewegungsevents, die in der Wohnung einer allein lebenden Frau über 220 Tage aufgezeichnet wurden. Die Wahl der Parameter folgte den Empfehlungen von [CUDDIHY et al., 2007, 93], die

¹<http://ailab.wsu.edu/casas/datasets/>

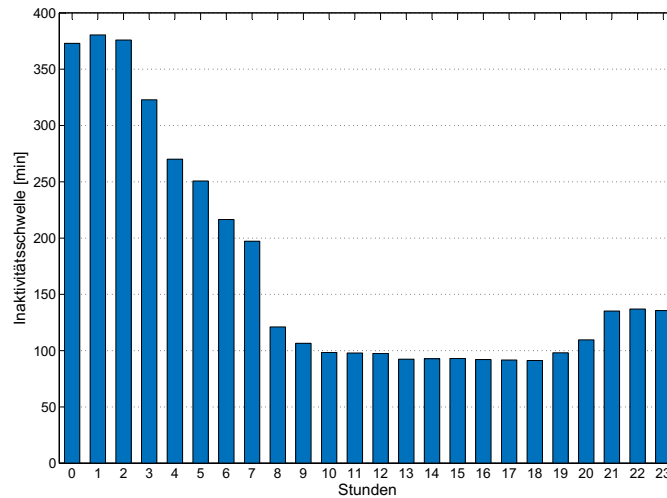


Abbildung 5.4: Durch die Erzeugung einer dynamischen Alarmschwelle können die tageszeitabhängigen Aktivitätsphasen der Nutzer berücksichtigt werden. Die zulässigen Inaktivitäten sind dabei in den Nachtstunden höher als während des Tages.

damit auf dem von ihnen verwendeten Datensatz in 93 % der Versuche ihr Ziel, nicht mehr als einen Fehlalarm pro Woche zu erzeugen, erreichen konnten. Unter Anwendung des Algorithmus' auf den *WSU CASAS Aruba*-Datensatz wurden unter Beibehaltung der empfohlenen Parameter 67 Alarmer für den gesamten Datensatz erzeugt. Da der Datensatz keinen Notfall enthält, sind diese Alarmer falsch-positiv und damit Fehlalarme. Abzüglich der ersten 30 Tage, die für die Bildung einer Datenbasis verwendet wurden, ergeben sich 190 Tage, an denen die Aktivitäten überwacht wurden. Daraus resultiert, dass im Durchschnitt alle 2,84 Tage ein Fehlalarm erzeugt wurde. Abweichend von [CUDDIHY et al., 2007] wurde für die Untersuchung in dieser Masterarbeit der Tag in 24 Intervalle unterteilt. Die dafür erzeugten Alarmgrenzen sind in Abbildung 5.4 dargestellt. Es zeigt sich der typische Inaktivitätsverlauf, bei dem geringe Inaktivität während des Nachmittags und hohe Inaktivität während der Nacht zu beobachten ist.

Ausgehend von dieser Untersuchung soll der vorgestellte Ansatz genutzt werden, um in künftigen Anwendungen Alarmer infolge erfasster Inaktivität zu erzeugen. Dazu muss in der ersten Phase der Inbetriebnahme des Roboters, das Aktivitätsmuster des Bewohners über den ersten Zeitraum erfasst werden. Damit in dieser Zeit dennoch eine Notfallüberwachung stattfinden kann, wird in dieser Phase des Lernens eine feste, für

alle Stunden gleiche, Inaktivitätsschwelle gesetzt werden, deren Überschreiten einen Alarm auslöst. Diese feste Schwelle wird auf 350 Minuten gesetzt und folgt damit den aus der Abbildung 5.4 erhaltenen Schwellen der Nacht. Im späteren Betrieb wird diese feste Schwelle durch die dynamische Schwelle ersetzt, die, entsprechend des beschriebenen Konzepts, für die aktuelle Stunde, aus den Beobachtungen der Vergangenheit erzeugt wird. Durch die fortlaufende Erfassung der Bewegungsevents während der Nutzung können langfristige Veränderungen im Tagesablauf (z.B. späteres Aufstehen oder späteres Mittagessen) erkannt und berücksichtigt werden. Damit prägt sich über die Zeit für jeden Nutzer ein individuelles Schwellwertprofil aus.

5.3.2 Erzeugung eines Immobility-Alarms

Wie auch bei der Erfassung des Inaktivitätsevents, werden für die Erkennung der Immobilität ebenfalls PIR-Sensoren genutzt, um Bewegungen zwischen den Bereichen zu registrieren. Mit jedem neuen Event wird der Übergang vom Bereich des letzten Events zum Bereich des aktuellen Events gespeichert. Dies erfolgt in Form einer Matrix der Größe $N \times N$, wobei N die Anzahl der verwendeten PIR-Sensoren angibt. Tabelle 5.1 zeigt exemplarisch die Struktur der Matrix, in der die Anzahl der Übergänge für den bisherigen Erfassungszeitraum enthalten sind. Die Zeilen der Tabelle repräsentieren dabei die Bereiche aus denen ein Übergang erfolgte, während die Spalten die Bereiche bezeichnen, in denen der Übergang endete. Die Werte in den Zellen geben die Anzahl der beobachteten Transitionen zwischen den jeweiligen Bereichen an. Mit jedem neuen Übergang wird der korrespondierende Eintrag erhöht.

Aus dieser Art der Datenrepräsentation lässt sich gut erkennen, zwischen welchen Bereichen häufige Übergänge (z.B. Übergang von Bereich 5 in Bereich 3) stattfinden und welche Bereiche nicht in direkter Verbindung zueinander stehen (z.B. Übergang von Bereich 3 in Bereich 6). Erwartungsgemäß sind die Einträge der Hauptdiagonale groß, da sie die Übergänge der Bereich in sich selbst repräsentieren. Der Aufenthalt in einem Bereich für einen längeren Zeitraum sorgt somit automatisch für hohe Werte der Hauptdiagonale. Werden die einzelnen Einträge einer Zeile auf die Gesamtanzahl der Übergänge dieser Zeile normiert, erhält man ein Maß für die Wahrscheinlichkeit,

		neuer Bereich					
		1	2	3	4	5	6
vorheriger Bereich	1	169	9	16	51	18	242
	2	12	27	20	17	91	7
	3	19	22	167	134	281	8
	4	56	14	142	268	76	64
	5	20	99	278	68	111	8
	6	264	17	39	136	38	274

Tabelle 5.1: Die Tabelle zeigt exemplarisch die genutzte Markov Mobilty Chain 1. Ordnung. Die Einträge in den Zellen geben die Anzahl der bereits erfassten Übergänge zwischen den jeweiligen Bereichen an. Werden die einzelnen Einträge auf die Summe einer Zeile normiert, ergibt sich die Übergangswahrscheinlichkeit von einem Bereich in einen anderen.

mit der die Übergänge aus einem Bereich in einen anderen stattfinden. Dieses Maß soll später genutzt werden, um ausgehend von einem Bereich weitere angrenzende Bereiche auszuwählen, in denen die Suche nach einer Person sinnvoll sein kann.

Die Idee, Übergänge zwischen verschiedenen Bereichen zu erfassen und daraus zukünftig besuchte Bereiche vorhersagen zu können, wurde bereits in [GAMBS et al., 2012] beschrieben. Die Autoren nutzen die sogenannten *Mobility Markov Chains* (MMC), um aus der Folge von besuchten Bereichen auf die nächsten Ziele zu schließen. Im Gegensatz zum Konzept der vorliegenden Masterarbeit, verwendeten die Autoren nicht nur Mobilty Markov Chains 1. Ordnung (*1-MMC*), bei denen nur der aktuelle und der vorherige Bereich betrachtet werden, sondern nahmen bis zu vier vorherige Bereiche (*4-MMC*) für die Prädiktion des nächsten Ziels. Die Untersuchungen zeigten, dass die Verwendung von Ordnungen $n > 2$ keinen wesentlichen Gewinn in der Genauigkeit der Vorhersage bringt [GAMBS et al., 2012]. Die zu verwaltende Datenmenge erhöht sich hingegen mit $N^{(n+1)}$, wobei N die Bereichszahl und n die Ordnung der MMC angibt. Die Erfassung der Übergangswahrscheinlichkeiten in der dargestellten Form wird in Kapitel 5.4 relevant, wenn die Reihenfolge der abzusuchenden Bereiche bestimmt

wird.

Da die beschriebene Erfassung der Übergangszahl noch keinen Zeitbezug hat, wird zusätzlich die Aufenthaltsdauer (Zeit vom Betreten bis zum Verlassen eines Bereiches) erfasst. Zur Bestimmung der Aufenthaltsdauer wird erfasst, wann ein Bereich betreten wird. Events, die aus dem bereits betretenen Bereich stammen, werden ignoriert. Erst bei Betreten eines neuen Bereiches wird die Aufenthaltsdauer für den letzten Bereich ermittelt. Die Aufenthaltsdauern werden für jeden Bereich getrennt erfasst. Dazu werden die summierten Aufenthaltszeiten zusammen mit deren Anzahl gespeichert. Neue Zeiten werden mit den alten Werten verrechnet. Aus der Division der summierten Aufenthaltszeiten mit der Anzahl ergibt sich dadurch ein gleitender Mittelwert der Aufenthaltszeiten für jeden Bereich.

Die auf diese Weise erhaltenen mittleren Aufenthaltszeiten werden genutzt, um Schwellen für die Alarmierung zu bestimmen. Dazu muss ein Kompromiss zwischen geringer Anzahl an Fehlalarmen und einer möglichst schneller Erkennung von Notfällen gefunden werden. Zur Zeit gibt es noch keine Daten, auf denen die Überprüfung einer entwickelten variablen Alarmierungsschwelle bei Immobilität durchgeführt werden kann. Der Grund dafür ist, dass die derzeit genutzte Versuchsumgebung im *Fachgebiet für Neuroinformatik und kognitive Robotik* an der TU Ilmenau durch die Mitarbeiter frequentiert wird, wodurch keine realistischen Daten über typische Aufenthaltszeiten während verschiedener Tätigkeiten des täglichen Lebens aufgezeichnet werden können. Diese sind jedoch die Grundlage bei der Entwicklung geeigneter Algorithmen zur Bestimmung zulässiger Verweildauern.

Ein vorläufiges Konzept zur Erzeugung einer Alarmierungsschwelle nutzt die erfassten mittleren Aufenthaltsdauern. Damit ein geringfügig längerer Aufenthalt als die mittlere Aufenthaltszeit nicht sofort einen Alarm auslöst, wird im derzeitigen Konzept die dreifache Aufenthaltszeit als Schwellwert genutzt. Ein zusätzlicher zeitlicher Puffer sorgt dafür, dass auch bei sehr kurzen erfassten Aufenthaltszeiten eine Mindestzeit des Aufenthalts durch das System akzeptiert wird. Die Idee eines Puffers wurde dem Konzept von [CUDDIHY et al., 2007, 91f.] zur Bestimmung einer Alarmierungsschwelle bei Inaktivität entliehen. Der Pufferzeit ist derzeit das Mittel der Aufenthaltszeiten

aller Bereiche.

Um in Zukunft eine Schwelle zu bestimmen, die die zeitliche Dynamik der Aufenthaltszeiten in den jeweiligen Bereichen besser berücksichtigt, als das derzeit verwendete Konzept, muss zuvor ein aussagekräftiger Datensatz erfasst werden, der Informationen über den Verlauf realistischer Aktivitäten in einer Wohnung einer allein lebenden Person enthält. Anstatt die mittleren Aufenthaltszeiten zu speichern, kann zukünftig die statistische Verteilung der Verweildauern für jeden Bereich erfasst werden. Aus dieser Datenrepräsentation ließen sich verlässlichere Aussagen zu akzeptablen Aufenthaltszeiten pro Bereich treffen.

Die Erzeugung eines Alarmevents läuft unabhängig vom verwendeten Konzept der Schwellenberechnung nach dem gleichen Prinzip ab. Jedes Mal, wenn ein neuer Bereich betreten wird, wird die jeweilige Alarmierungsschwelle für den betretenen Bereich bestimmt und die Zeit des Betretens gespeichert. Bei neuen Events aus demselben Bereich wird die bisherige Aufenthaltszeit aus der aktuellen Zeit und der gespeicherten Startzeit bestimmt und mit der Schwelle der zulässigen Aufenthaltszeit verglichen. Wird die Alarmschwelle überschritten, erfolgt eine Alarmierung durch das Schreiben auf den `IMMOBILITYALERT`-Channel. Wird hingegen ein neuer Bereich betreten, erfolgt die Bestimmung einer neuen bereichsspezifischen Alarmschwelle und die aktuelle Zeit wird als Beginn des neuen Aufenthalts vermerkt.

Die Erfassung der Bewegungen zwischen den Bereichen der Wohnung erfolgt in der `MMCUNIT` des `EMERGENCYSERVICE`. In ihr werden, aus den erfassten mittleren Verweildauern, die Schwellen für die zulässigen Aufenthaltszeiten pro Bereich gebildet. Zusätzlich erfolgt hier die Erfassung und Aktualisierung der Übergänge der Mobility Markov Chain. Entsprechend des in Kapitel 3 vorgestellten Schichtenmodells (vgl. Abb. 3.1) stellen sowohl die `MMCUNIT` wie auch der `NOACTIVITYDETECTOR` einen Skill dar, der von anderen Units oder Behaviors genutzt werden kann.

5.4 EmergencySearch-Behavior

Im vorangegangenen Abschnitt wurden die beiden Mechanismen zur Erkennung von Notfällen vorgestellt. Diese bilden die Skills des `EMERGENCYSERVICE` und laufen, von

den übrigen Bestandteilen des Services unbemerkt, im Hintergrund ab. Erst wenn die analysierten Bewegungs- bzw. Ruhesequenzen die Annahme eines Notfalls bestätigen, signalisieren sie dies dem Frame des EMERGENCYSERVICE durch senden eines *NoActivity*- bzw. *ImmobilityAlert*-Events auf den entsprechenden Channeln. Wird auf diese Weise ein Notfall erkannt, wird das EMERGENCYSEARCH-Behavior gestartet, welches die Suche nach der hilfsbedürftigen Person übernimmt. Da dieses Modul den Roboter bewegen muss und deshalb Zugriff auf alle Komponenten der Navigation benötigt, wird es als Behavior implementiert.

Für eine möglichst effektive Suche müssen geeignete Zielpositionen gefunden werden, die vom Roboter nacheinander angefahren werden können. In dieser Arbeit wird davon ausgegangen, dass durch die genutzten Bewegungssensoren eine kontinuierliche Erfassung der Aufenthaltsorte der Person gegeben ist. Daher kann angenommen werden, dass der Sensor, der zuletzt auslöste, bevor ein Notfall signalisiert wurde, auch der Aufenthaltsort der Person ist. Aus diesem Grund nimmt das EMERGENCYSEARCH-Behavior den Ort des letzten Bewegungsevents als erste Zielposition der Suche. Dazu wird vom Channel LATESTSENSORID die ID des zuletzt ausgelösten Sensors gelesen. Die Umwandlung der Sensor-ID in die Zielkoordinaten erfolgt über eine Liste von vorgegebenen, in der Wohnung verteilten Positionen, die auf dem Channel NAVPOINTS liegen. Diese Koordinaten sind so gewählt, dass der Roboter von ihnen aus den jeweiligen Bereich der Wohnung bestmöglich überblicken kann, damit Personen, sich in diesem Bereich befinden, zu erkennen. In Abbildung 6.1 im Kapitel 6 sind die festgelegten Beobachtungspositionen des Roboters dargestellt.

Wie im Kapitel 6.1 zum Setup der Testumgebung näher erläutert, ist es möglich, dass Bewegungsfolgen in der Wohnung aufgrund der Latenz, die die PIR-Sensoren haben, nicht exakt erfasst werden können. Dadurch kann es vorkommen, dass eine Person einen Bereich, in dem sie einen Sensor ausgelöst hat, bereits verlassen hat, ohne dass der Sensor des neu betretenen Bereichs ein neues Event generieren konnte. Es genügt daher nicht, bei der Personensuche lediglich den Bereich des letzten Sensor-Events anzufahren. Aus diesem Grund sind weitere Zielpositionen nötig, die nicht willkürlich gewählt sind, sondern aus den Häufigkeiten der erfassten Übergängen zwischen

den Bereichen bestimmt werden sollen. Dazu wird die Übergangsmatrix (*Mobility Markov Chain*) genutzt, die in Kapitel 5.3.2 bereits vorgestellt wurde. In ihr sind die Häufigkeiten der Übergänge zwischen den Bereichen erfasst. Diese werden genutzt, um mithilfe der Übergangswahrscheinlichkeiten die Reihenfolge festzulegen, in der die Bereiche vom Roboter abgefahren werden sollen.

Sollte bei der Fahrt zum Bereich des letzten ausgelösten Sensors die Person nicht gefunden worden sein, wird der Bereich mit der höchsten Übergangswahrscheinlichkeit als neues Ziel der Navigation übergeben. Auf diese Weise werden nacheinander alle Bereiche der Wohnung abgefahren, wobei die wahrscheinlichsten zuerst besucht werden und die unwahrscheinlichsten zuletzt. Die Suche endet entweder wenn die Person gefunden wurde oder wenn alle Bereiche erfolglos angefahren wurden. Sollte der Fall eintreten, dass trotz Suche in allen Bereichen der Wohnung die Person nicht gefunden werden konnte, wird das Event *NoPersonFound* erzeugt, welches über die erfolglose Suche informiert. Befindet sich die Person in der Wohnung, konnte jedoch nicht vom Roboter gefunden werden, wird ein unspezifischer Notruf abgesetzt.

Während der gesamten Suche sind die Module zur Erkennung von stehenden und liegenden Person aktiv. Sie überprüfen, ob sich eine Person in dem Bereich vor dem Roboter befindet. Das Vorhandensein einer Person unterbricht die Suchfahrt und meldet dies dem Frame des EMERGENCYSERVICE über das Event *StandingPersonDetected* bzw. *FallenPersonDetected*. Die Information, ob eine liegende Person detektiert wurde, liegt auf dem Channel `FALLENPERSONHYPOTHESES` und enthält eine Datenstruktur, in der die, die Person umgebende Bounding Box mit den Koordinaten der Eckpunkte, die Koordinaten des Schwerpunktes sowie ein Maß für die Wahrscheinlichkeit (Score) einer Personendetektion enthalten sind. Die Wahrscheinlichkeit der Personenhypothese muss einen festgelegten Grenzwert überschreiten, damit das EMERGENCYSEARCH-Behavior eine liegende Person erkennt und die Suchfahrt daraufhin abbricht.

Über den Navigationsstatus informieren Events auf dem Channel `PILOTEVENT`. Das Event *GoalReached* signalisiert das Erreichen der Zielposition und führt dazu, dass ein neues Ziel gesetzt wird, wenn noch keine Person gefunden wurde. Das Event *NoPa-*

thPlannable wird ausgelöst, wenn es der Pfadplanung nicht möglich war, einen vom Roboter befahrbaren Weg zwischen der aktuellen Roboterposition und der vorgegebenen Zielposition herzustellen. Dieses Event sorgt dafür, dass das aktuelle Ziel verworfen und das nächste Ziel gesetzt wird.

Während der Entwicklung fiel auf, dass es gelegentlich vorkommt, dass der Roboter beginnt einen Pfad zum Ziel zu planen, aber aus unterschiedlichen Gründen dieser Prozess nicht terminiert. Der Roboter befindet sich dann im Zustand *PlanAndDrive*, erreicht jedoch nicht den Zustand *NoPathPlannable*. Für diesen Fall wurde ein Timer ergänzt, der mit dem Setzen einer neuen Zielkoordinate gestartet wird. Bewegt sich der Roboter (intern zu erkennen an der Änderung der zurückgelegten Strecke auf dem Channel MILEAGE) wird der Timer mit jeder Bewegung zurückgesetzt. Bewegt sich der Roboter nicht mehr oder hat er sich nach dem Setzen der Koordinaten gar nicht erst bewegt, beendet ein Timeout die missglückte Zielfahrt und es wird ebenfalls mit der Fahrt zum nächsten Ziel der Liste fortgefahren. Die Zeit bis zum Timeout ist dabei beliebig einstellbar.

Die Information, ob das EMERGENCYSEARCH-Behavior eine Person gefunden hat oder nicht und ob es sich dabei um eine liegenden oder stehende Person handelt, gelangt in Form eines Behavior-Events nach außen zum Serviceframe, wo der eventabhängige weitere Programmablauf gesteuert wird. Wurde eine Person während der Suche detektiert, die jedoch im weiteren Ablauf nicht bestätigt werden konnte, wird das EMERGENCYSEARCH-Behavior erneut gestartet und setzt die Suche an den verbleibenden Zielen fort. Das ist zum einen der Fall wenn eine gefundene liegende Person im Zuge der Verifikation nicht bestätigt werden konnte. Zum anderen sorgt ein Timeout im Dialog dafür, dass weitere Ziele angefahren werden, wenn der Roboter fälschlicherweise eine stehende Person erkannt hat. Wird der, vom Roboter gestartete Dialog, nicht durch einen Nutzer erwidert, wird beim Erreichen des Timeouts davon ausgegangen, dass es sich um eine Fehldetektion gehandelt haben muss.

5.5 VerifyFallenPerson-Behavior

Nachdem die Personensuche durch eine Personenhypothese unterbrochen wurde, soll mithilfe des VERIFYFALLENPERSON-Behaviors das Vorhandensein einer liegenden Person bestätigt oder widerlegt werden. Da auch dieses Modul Aufgaben der Navigation und der Roboterbewegung übernimmt, ist es ebenfalls von der Behavior-Klasse abgeleitet, so dass es durch den BehaviorManager gesteuert werden kann. Dieser startet es, sobald das EMERGENCYSEARCH-Behavior das Event *FallenPersonDetected* sendet. Um die Hypothese, eine gestürzte Person gefunden zu haben, überprüfen zu können, muss der Roboter geeignete Beobachtungspositionen einnehmen. Aus diesem Grund sollen rund um die Person Positionen angefahren werden, von denen aus der Roboter die Szene aus verschiedenen Blickwinkeln überprüfen kann. Der Hypothesenchannel liefert zusätzlich zur Bounding Box die Koordinaten des Personenschwerpunktes. Damit der Roboter möglichst zielgerichtet auf die Person zufährt, soll die erste Beobachtungsposition auf der Verbindungslinie zwischen der aktuellen Roboterposition und dem Mittelpunkt der Person liegen. Die erste Beobachtungsposition wird so bestimmt, dass sie sich in einem Abstand von 1 m zum Personenmittelpunkt befindet. Entsprechend der gewählten Anzahl von Beobachtungspositionen werden nun um die Person weitere Beobachtungspositionen erzeugt. Sie liegen alle auf einem Kreis mit einem Radius von 1 m um den Personenmittelpunkt. Zusätzlich wird die Orientierung des Roboters in jeder Beobachtungsposition so berechnet, dass er immer in Richtung des Kreismittelpunkts blickt. Die so ermittelten Roboterposen (x, y, phi) bilden die groben (vorläufigen) Beobachtungspositionen. Im nächsten Schritt werden aus den groben Positionen, die lediglich einer ersten Platzierung um die Person dienen, die konkreten Zielpositionen bestimmt.

Die Bestimmung der Zielpositionen erfüllt den Zweck, eine Position zu finden, die einen möglichst geringen Abstand zur Person aufweist. Das ist wichtig, damit die Person im späteren Dialog über Berührungen am Robotergehäuse signalisieren kann, ob Hilfe benötigt wird oder nicht. Um die endgültigen Zielpositionen des Roboters zu bestimmen, werden Positionen auf dem Strahl gesucht, der vom Personenmittelpunkt durch die jeweilige Grobposition führt. Ein festgelegter Bereich (z.B. von 0.5 m bis 2 m)

gibt den minimalen und maximalen Abstand vom Personenmittelpunkt an, innerhalb dessen eine mögliche Zielposition gesucht wird.

Zur Überprüfung werden zwei verschiedene Karten genutzt. Die eine Karte (*GlobalMap*) enthält alle Hindernisse, die über die Zeit als unveränderlich vom Roboter gelernt wurden. Die andere Karte ist eine Repräsentation, in der die festen Hindernisse mit den dynamischen Hindernissen aus den aktuellen Sensorbeobachtungen zusammengeführt werden (*MergedMap*). Beim Test auf Kollisionsfreiheit prüft der Roboter auf beiden Karten, ob er die betrachtete Zielposition einnehmen kann ohne dass er mit seiner Grundfläche in Bereiche, die von Hindernissen belegt sind, gelangt. Ist die, auf diese Weise kontrollierte, Position frei, wird sie als Fahrkommando der Navigation mitgeteilt. Hat der Roboter die Position erreicht, wird eine neue Zielposition auf dem Strahl gesucht, der durch die nächste Grobposition führt. Ist die Position jedoch nicht frei, wird die nächste Position auf dem Strahl überprüft. Dabei wird für jede neue Position der Abstand zum Mittelpunkt um 5 cm erhöht, bis entweder eine freie Position gefunden wurde oder der maximale zulässige Abstand entlang dieses Richtungsvektors überschritten wurde.

Die Bestimmung von Zielpositionen ausgehend von den zuvor erzeugten Grobpositionen erfüllt dabei zwei Aufgaben. Zum einen ermöglicht sie eine maximale Annäherung an die Person, zum anderen ist es möglich, eine befahrbare Zielposition auf dem Richtungsvektor zu finden, wenn die ursprünglich ermittelte Grobposition nicht befahrbar ist, da sie z.B. im Bereich von Hindernissen liegt. In Abbildung 5.5 wird die Bestimmung von Beobachtungspositionen um eine liegende Person schematisch dargestellt.

Damit der Blick des Roboters nicht durch ein Hindernis (z.B. eine Wand oder einen Raumteiler) verstellt ist, wird vor dem Anfahren der zuvor bestimmten Zielposition überprüft, ob sich zwischen der Zielposition und dem Personenmittelpunkt Hindernisse befinden. Dazu wird für jede Zelle der *GlobalMap*, die sich auf der Verbindungslinie der beiden Punkte befindet, ein Kollisionstest durchgeführt. Nur wenn alle Zellen als nicht belegt verifiziert wurden, wird die Zielposition vom Roboter angefahren. Andernfalls muss von einer möglichen Sichtbehinderung ausgegangen werden und die nächste Zielposition auf dem nächsten Strahl wird bestimmt. Die *GlobalMap* wird genutzt, da

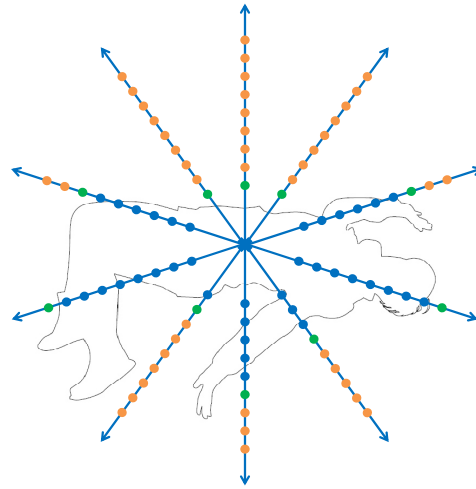


Abbildung 5.5: Um geeignete Positionen für die Verifikation zu finden, werden 10 Strahlen ausgehend vom Personenmittelpunkt erzeugt. Auf den Strahlen wird für verschiedene Beobachtpositionen überprüft, ob die Position durch den Roboter befahren werden kann. Die Überprüfung erfolgt dabei unter Beachtung der Größe der Grundfläche des Roboters auf den beiden verwendeten Karten. Positionen die der Roboter nicht einnehmen kann sind im Schema blau dargestellt. Von den erreichbaren Positionen (orange), werden jeweils die innersten gewählt (grün). Damit folgt der Roboter während der Verifikation der Silhouette der Person.

sie im Gegensatz zur *MergedMap* nur feste Hindernisse, wie z.B. Wände oder Möbel, jedoch nicht die liegende Person enthält.

Eine Problematik ergab sich bei der Anfahrt von Positionen, die sich vom Roboter aus gesehen hinter der Person befanden. Dem Roboter ist es in diesem Fall nicht möglich, die genaue Größe eines Objektes zu ermitteln, wenn er sich diesem nur von einer Seite nähert. Wenn der Roboter auf der Fahrt zu einer, vom *EMERGENCYSEARCH*-Behavior gesetzten, Beobachtposition ein Hindernis mit Personenhypothese wahrnimmt, wird die Anfahrt gestoppt und die Verifikation gestartet. Bis zu diesem Zeitpunkt erfolgte die Navigation anhand der *GlobalMap*, die nur die festen, unveränderlichen Hindernisse enthält. Durch Zusammenführen der *GlobalMap* mit den, vom Roboter erkannten, dynamischen Hindernissen entsteht die *MergedMap*.

Anhand der *MergedMap* soll im *VERIFYFALLENPERSON*-Behavior zu den Beobach-

tungsposition um die vermeintliche Person herum navigiert werden. Da der Roboter das Hindernis jedoch nur von einer Seite sieht, kann dessen genauer Umriss nicht in die *MergedMap* übernommen werden. Das kann dazu führen, dass während der Erzeugung der Verifikationspositionen hinter der Person, deren Befahrbarkeit nicht exakt geprüft werden kann, wodurch es zu Problemen kommt.

Versucht der Roboter, Positionen anzufahren, die durch die Person belegt sind oder sehr nah am Hindernis liegen, kommt es zu einem Wechselspiel der Bewegungssteuerung und der Kollisionsvermeidung. Die Bewegungssteuerung versucht, die vorgegebene Position zu erreichen, während gleichzeitig die Bewegung unterbrochen wird, um Kollisionen mit dem Hindernis zu vermeiden. Dieses Wechselspiel aus Bewegung und Bewegungsabbruch führt zu einem „Zappeln“ des Roboters, ohne dass das Ziel erreicht wird. Der Roboter ist in einer Endlosschleife gefangen, die er nicht beenden kann.

Damit der Roboter nicht in den beschriebenen Zuständen gefangen bleibt, wird die Verlauf der Roboterpositionen der letzte 3 s betrachtet. Dazu wird der Abstand zwischen den entsprechenden x - und y -Koordinaten der aktuellen Position und der 3 s zurückliegenden Position gebildet. Zusätzlich wird die Winkeldifferenz der beiden Roboterposen berechnet.

Startet der Roboter zu einem neuen Ziel, legt er in 3 s jeweils eine weitere Strecke zurück bzw. dreht sich schneller, als wenn er sein Ziel fast erreicht hat und nur noch kleine Positionskorrekturen vornimmt. Die Beträge der Richtungsvektoren bzw. die Winkeldifferenzen sind daher während der Fahrt größer als wenn sich der Roboter dem Ziel angenähert hat. Befindet sich der Roboter nahe am Ziel, werden diese Werte immer kleiner, da sich in derselben Zeiteinheit die Positionen nur noch geringfügig ändern.

Befindet sich der Roboter im beschriebenen Zustand, in dem er versucht eine belegte Position zu erreichen, die Bewegung jedoch wiederholt abgebrochen wird, sind die Positionsänderungen minimal. Bei Unterschreiten einer gesetzten Schwelle für die Länge des Richtungsvektors (Abstand zwischen den Positionen) und gleichzeitigem Unterschreiten einer Winkeldifferenz, wird die Positionsanfahrt abgebrochen und die Fahrt zum nächsten Ziel fortgesetzt.

Damit kommt der Roboter so nah wie möglich an die Person heran, hängt jedoch nicht mehr in der Endlosschleife fest. Da sich nun der Roboter hinter der Person befindet, kann von dieser Position aus die *MergedMap* um die neuen Informationen zur tatsächlichen Größe des Hindernisses ergänzt werden. Dadurch können die weiteren Positionen mithilfe der aktualisierten Karte bestimmt werden.

Hat der Roboter seine Beobachtungsposition erreicht (das wird über das Event *GoalReached* auf dem *PILOTEVENT*-Channel signalisiert) kann vom *FALLENPERSONHYPOTHESES*-Channel die Wahrscheinlichkeit für eine gestürzte Person in der vorliegenden Szene gelesen werden. Da im enthaltenen Score stets auch die Scores aus den letzten Beobachtungen enthalten sind, wird der Score (und damit die Sicherheit einer Hypothese) genauer, je mehr Ansichten erfasst werden.

Überschreitet die Hypothesenwahrscheinlichkeit einen festgelegten Schwellwert, wird das Behavior-Event *FallenPersonConfirmed* gesendet. Weitere Beobachtungspositionen werden nicht mehr angefahren und das *VERIFYFALLENPERSON*-Behavior wird vom BehaviorManager beendet. Erreicht die vom Channel gelesene Wahrscheinlichkeit nicht den Schwellwert, wird die nächste Beobachtungsposition angefahren. Wurde an keiner Beobachtungsposition der gesetzte Schwellwert erreicht, d.h. die Verifikation konnte von keiner Position eine Person bestätigen, meldet dies das Event *FallenPersonRejected*. Der BehaviorManager beendet daraufhin das *VERIFYFALLENPERSON*-Behavior, startet dafür aber erneut das *EMERGENCYSEARCH*-Behavior, um die Suche, der Reihe nach, an den verbliebenen Positionen fortzusetzen. Dieser Vorgang wiederholt sich so lange, bis eine Person sicher an einer Position bestätigt wurde oder bis alle Bereiche überprüft wurden. Wie auch beim *EMERGENCYSEARCH*-Behavior informiert der Channel *PILOTEVENT* über den Status der Navigation und der Zielfahrt. Die Events *GoalReached* und *NoPathPlannable* starten die Bestimmung des nächsten Ziels. Außerdem wird mit jedem neu gesetzten Ziel ein Timer aktiviert, der ein Timeout-Event auslöst, wenn sich der Roboter für eine gewisse Zeit nicht bewegt hat und keines der zuvor genannten Events gemeldet wurde. In diesem Fall wird ebenfalls mit der Bestimmung des nächsten Ziels fortgefahren.

5.6 Ablaufsteuerung / Dialog

Nachdem in den vorangegangenen Kapiteln die Suche einer Person in unterschiedlichen Bereichen und die folgende Verifikation einer liegenden Person beschrieben wurde, soll in diesem Kapitel auf die Kommunikation zwischen Roboter und Mensch eingegangen werden.

Um die jeweiligen Inhalte zu transportieren, nutzt der Roboter die gesprochene Sprache, in der Form, dass vorgefertigte text-to-speech-synthetisierte Sätze situationsabhängig ausgegeben werden. Gleichzeitig erfolgt eine Ausgabe auf dem Bildschirm des Roboters, sodass der Nutzer die Information lesen und gegebenenfalls über entsprechende Buttons darauf reagieren kann. Die gesprochene Sprache hat gegenüber der geschriebenen den Vorteil, dass Informationen den Nutzer auch erreichen, wenn sich dieser nicht in direkter Nähe zum Roboter befindet und deshalb die Bildschirmausgaben nicht lesen kann. Zudem ist die auditiv-verbale Form die natürlichste Kommunikationsform. Der Roboter startet diese Art der Kommunikation mit verschiedenen Absichten. Zum einen kann er damit über interne Zustände informieren, ohne dass eine Reaktion des Nutzers notwendig ist. So könnte der Roboter bei niedrigem Akkustand durch eine Ausgabe wie „Ich bin müde. Jetzt muss ich erst einmal eine Pause machen.“ den Nutzer darüber informieren, weshalb er gegebenenfalls eine Tätigkeit abbricht und zu seiner Ladestation fährt. Dem Nutzer ist es durch diese Metakommunikation möglich, die Verhaltensweisen des Roboters richtig einzuordnen, die sonst als unplausibel wahrgenommen werden könnten. Auf der anderen Seite kann der Roboter die Kommunikation auch im Sinne von Frage-Antwort-Dialogen nutzen, um explizite Eingaben vom Nutzer zu erhalten, die die weitere Ausführung einer Aktion bestimmen. Die Frage „In welches Zimmer soll ich fahren?“ zusammen mit entsprechenden Auswahlmöglichkeiten, würde zur Ausführung der jeweiligen konkreten Aktion führen. Im EMERGENCYSERVICE werden beide der genannten Kommunikationsformen genutzt.

Die Nutzerdialoge werden als eigene Frames definiert, die durch den ServiceManager gesteuert werden. Dadurch wird sichergestellt, dass es keine Konflikte zwischen verschiedenen Frames oder Services gibt, die gleichzeitig in Interaktion mit dem Nutzer treten wollen. Die Dialogstruktur wird durch einen Entscheidungsbaum geregelt, der

je nach aktuellem Event die entsprechende Aktion auswählt. Die Erkennung eines Notfalls erzeugt ein Event auf einem Slot, das die Aktivierung des EMERGENCYFRAME auslöst. Dieser Frame ist so konzipiert, dass er bei Aktivierung das EMERGENCY-SEARCH-Behavior startet und damit die Suche der Person in der Wohnung beginnt. Gleichzeitig wird der Nutzer durch die Sprachausgabe mit dem Inhalt „Ich suche dich. Wo bist du?“ darauf aufmerksam gemacht, dass der Roboter ihn sucht. Da zu diesem Zeitpunkt nicht sichergestellt ist, ob es sich um einen tatsächlichen Notfall oder um einen Fehlalarm handelt, ist die Sprachausgabe absichtlich neutral gehalten, um den Nutzer nicht durch negativ konnotierte Begriffe wie „Alarm“, „Notfall“ oder „Hilfe“ in Aufregung zu versetzen.

Wurde eine stehende Person gefunden, soll mit der Sprachausgabe „Brauchst du Hilfe? Ich habe etwas bemerkt.“ geklärt werden, ob ein Notfall vorliegt oder ob ein Fehlalarm ausgelöst wurde. Dabei erscheinen die beiden Sätze zusammen mit den Buttons „Ok“ und „Alarm“ auf dem Bildschirm des Roboters. Betätigt der Nutzer den „Ok“-Button, gelangt der Roboter zurück in den Zustand, in dem er sich vor der Alarmierung befand. Erst ein erneuter *NoActivity*- bzw. *Immobility*-Alarm führt zur Aktivierung des EMERGENCYSERVICE.

Da in der Konzeption dieser Arbeit davon ausgegangen wird, dass eine gestürzte Person sowohl bewusstlos wie auch bei Bewusstsein sein kann, jedoch nicht in der Lage ist, aus eigener Kraft aufzustehen oder den Bereich zu verlassen, muss der folgende Dialog so konzipiert sein, dass mit ihm auf beide möglichen Situationen reagiert werden kann. Dazu wird die Person mit „Geht es dir gut? Wenn ich Hilfe holen soll, berühre mich jetzt am Gehäuse.“ angesprochen.

Zusätzlich zu den Eingabemöglichkeiten über den Bildschirm, wird nun auch die Berührung der kapazitiven Sensoren am Gehäuse erfasst und als Nutzereingabe gewertet. Damit wird dem Umstand Rechnung getragen, dass auf dem Boden liegende Personen den Bildschirm für Eingaben nicht erreichen können. In diesem Fall, wäre die Gehäuseberührung gleich einer Bestätigung, einen Alarm auszulösen. Über einen Button auf dem Display kann das Absetzen eines Notrufes unterbunden werden. Ist die Person z.B. durch eine Ohnmacht für eine bestimmte Zeit nicht in der Lage, zu reagieren,

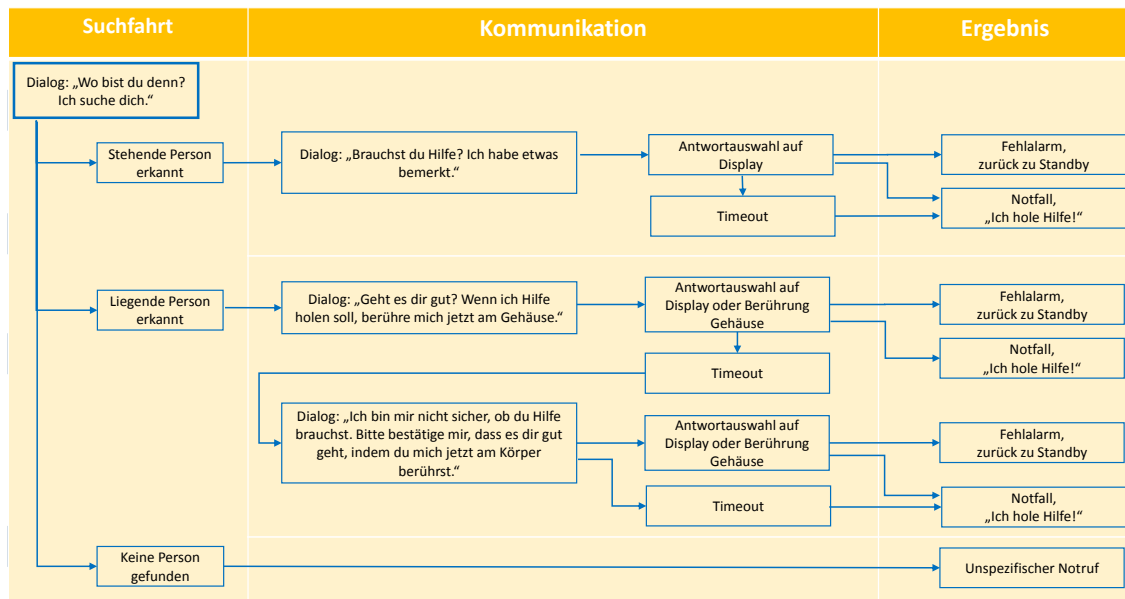


Abbildung 5.6: Der Ablauf des Dialogs hängt von der vorliegenden Situation ab. Dabei wird unterschieden, ob eine liegende, sitzende oder stehende Person durch den Roboter gefunden wurde. Für den Fall, dass die gestürzte Person bewusstlos ist, soll der Dialog auch ohne aktiv Bestätigung zum Absetzen eines Notrufes führen. Wird in allen abgesuchten Bereichen keine Person gefunden, wird ein unspezifischer Notruf abgesetzt.

führt ein Timeout zum nächsten Teil des Dialogs. Für diesen Fall werden die Sätze „Ich bin mir nicht sicher, ob du Hilfe brauchst. Bitte bestätige mir, dass es dir gut geht, indem du mich jetzt am Körper berührst.“ ausgegeben. Nach dieser Sprachausgabe hat der Nutzer die Möglichkeit, das Senden eines Notrufes entweder durch Betätigung eines Buttons auf dem Bildschirm oder durch eine Berührung des Roboterkörpers zu unterbrechen. Unterbleibt eine Reaktion des Nutzer führt ein erneuter Timeout zum Absetzen des Notrufes. Es ist zu erkennen, dass die Dialoge so konzipiert sind, dass im ersten Teildialog die Berührung des Gehäuses einen Notruf absetzt, wohingegen sie im zweiten Teildialog einen Notruf verhindert. Für den bewusstlosen Nutzer bedeutet dies, dass ein Nichtreagieren auf beide Dialoge einen Notruf auslöst. Der Mechanismus der doppelten Aufforderung dient dazu, Fehlalarme zu vermeiden. In Abbildung 5.6 sind die verschiedenen Dialoge dargestellt.

Wie bereits in Kapitel 4 vorgestellt, verfügt der Roboter über animierte Augen, die

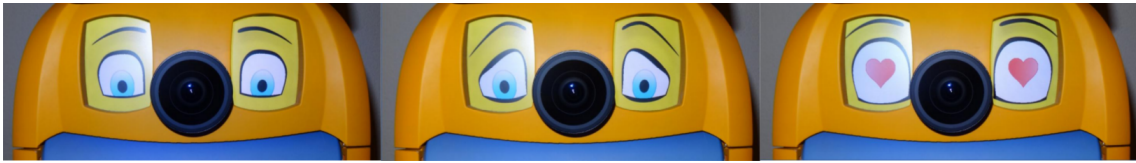


Abbildung 5.7: Der Roboter verfügt über verschiedene Gesichtsausdrücke, die genutzt werden, um den Nutzer über die internen Zustände zu informieren, in denen sich der Roboter aktuell befindet. Der neutrale Gesichtsausdruck (links) ist standardmäßig eingestellt, wenn keine besonderen Ereignisse erfasst wurden oder Aktionen ausgeführt werden. Mit dem besorgten Gesichtsausdruck (mitte) informiert der Roboter der Suchfahrt und des Dialogs über die Vermutung einer Notsituation. Die pulsierenden Herzen (rechts) zeigen die Freude des Roboters, nachdem der Nutzer einen vermeintlichen Notfall als Fehlalarm bestätigen konnte.

genutzt werden können, um über interne Zustände zu informieren oder Emotionen zu transportieren. Von dieser Möglichkeit wird im Verlauf des Dialogs Gebrauch gemacht, um zusätzlich zur Sprachausgabe zu signalisieren, dass der Roboter eine Situation vermutet, die eine Überprüfung erforderlich werden lässt. Damit der Nutzer die Absicht des Roboters erkennen kann, nimmt der Roboter einen besorgten Gesichtsausdruck an, sobald er die Suche nach der Person startet. Auch die Durchführung des Dialogs wird von der besorgten Mine begleitet. Bestätigt der Nutzer, dass kein Notfall vorliegt, signalisiert der Roboter seine Erleichterung durch Herzen, die in seinen Augen pulsieren. Begleitet wird dies durch den Satz „Freut mich, dass ich mir umsonst Sorgen gemacht habe“. Daraufhin gelangt der Roboter aus dem Alarmierungszustand in den Normalmodus. Das wird durch den neutralen Gesichtsausdruck signalisiert.

Diese zusätzliche Kommunikationsform über die Mimik soll neben dem Informationstransport auch die Sympathie des Nutzers erhöhen, auch wenn der Roboter gelegentlich Situationen als vermeintliche Notfälle bewertet, die sich nach der Überprüfung als Fehlalarme herausstellen. In Abbildung 5.7 sind die drei Emotionen (neutral, besorgt, erfreut) abgebildet, die über die Roboteraugen ausgedrückt und im Verlauf des Services zu Notfallbehandlung verwendet werden.

Kapitel 6

Funktionstest

Nachdem in Kapitel 5 die Implementierung einer Strategie zu Notfallbehandlung vorgestellt wurde, sollen in den folgenden Funktionstests die Funktionalität entstandenen Services zur Notfallbehandlung getestet werden. Es soll damit die Frage beantwortet werden, ob die einzelnen Module derart ineinander greifen, dass ein stimmiger Gesamtprozess entsteht.

Ziel ist es, eine qualitative Beschreibung zu erstellen, um das problemlose Interagieren der Module zu bestätigen und gegebenenfalls Störungen des Zusammenspiels zu identifizieren sowie Verbesserungen abzuleiten.

Der vorgestellte Service ist modular aufgebaut. Er enthält sowohl Komponenten, die in dieser Masterarbeit entstanden sind, sowie Module, die im Rahmen anderer wissenschaftlicher Arbeiten am Fachgebiet für Neuroinformatik und kognitive Robotik entwickelt wurden. Aus diesem Grund soll im folgenden Funktionstest nicht die Leistungsfähigkeit der einzelnen Modul beurteilt werden, obwohl natürlich Faktoren wie die Genauigkeit der Personendetektionen oder des Navigationsverhaltens einen großen Einfluss auf den Gesamtprozess haben.

Vielmehr soll der Funktionstest dazu dienen, Stärken und Schwächen des vorgestellten Services zu identifizieren und Erkenntnisse für zukünftige Erweiterungen zu liefern. Zur Beurteilung der Module zur Detektion stehender und liegender Personen, zur Navigation und zum BehaviorManager sei an dieser Stelle auf die Arbeiten von ([WEINRICH, 2016], [LEWANDOWSKI, 2016], [MÜLLER, 2016]) verwiesen. In diesen Arbeiten wurde

die Funktionalität der Module nachgewiesen.

In der vorliegenden Masterarbeit soll die Integration der Module zu einem Gesamt-
ablauf getestet werden.

6.1 Aufbau der Versuchsumgebung

Für die Durchführung von Robotertests unter annähernd realen Bedingungen wurde in den Räumlichkeiten des Fachgebietes für Neuroinformatik und kognitive Robotik eine Testwohnung (Living Lab) eingerichtet. Diese ist von ihrer Größe und dem Interieur ähnlich einer Wohnung, wie sie auch von Senioren bewohnt werden könnten. Durch die Anordnung von Wänden und aufgebauten Schränken lassen sich fünf Bereiche voneinander abgrenzen. Die Wohnung enthält eine Küchenzeile und in den einzelnen Bereichen unterschiedliche Sitzgelegenheiten, wie z.B. Stühle, Sessel und ein Ecksofa. Dabei können in der Testwohnungen die Aktivitäten des täglichen Lebens simuliert werden, die für verschiedene Fragestellungen im Bereich der Assistenzrobotik von Interesse sein können. Zusätzlich führt die Anordnung des Mobiliars dazu, dass der Roboter auf beschränktem Raum manövrieren muss und teilweise gewisse Bereiche der Wohnung nicht erreichen kann. Damit schafft die Testwohnung Bedingungen, mit denen sich Roboter auch in einem realen Szenario konfrontiert wird.

Zur Erfassung der Bewegungen und der daraus abgeleiteten Erkennung von Notfällen, ist das Living Lab mit sieben PIR-Bewegungsmeldern (passive infrared sensors) ausgestattet. Diese sind so in der Wohnung verteilt, dass die Bewegungen in den einzelnen Bereichen von mindestens einem Sensor erfasst werden. Bei den verwendeten Sensoren handelt es sich um Bewegungsmelder des Typs 882 09 (IP 20) der *Legrand GmbH*. Diese batteriebetriebene Sensoren eignen sich gut, um auch nachträglich in bestehenden Wohnungen integriert zu werden, ohne dass eine aufwendige Installation, wie bei kabelgebundenen Systemen, nötig ist. Der Detektionsbereich dieses Modells ist mit 12 m angegeben und der Öffnungswinkel beträgt 90°. Damit lassen sich bei geeigneter Sensoranordnung Bewegungen innerhalb eines Zimmers, in den meisten Fällen, schon mit einem Sensor erfassen. Abbildung 6.1 zeigt den Aufbau der Testumgebung. Hierfür



Abbildung 6.1: Die Testumgebung besteht aus vier, abgetrennten Bereichen. Die Bewegungen in jedem Bereich werden durch Bewegungsmelder (PIR-Sensoren) erfasst. Durch die Anordnung der Sensoren und deren große Erfassungsbereiche können Bewegungen in weiten Teile der Wohnung detektiert werden. Auf seiner Suchfahrt navigiert der Roboter zu festgelegten Beobachungspositionen. Diese sind so gewählt, dass von ihnen die Bereiche bestmöglich eingesehen werden können. Die Pfeile an den Beobachungspositionen geben die Blickrichtung des Roboters an. Der grün schraffierte Bereich wurde für die Funktionstests nicht genutzt und enthielt deshalb keine PIR-Sensoren. Die Basis der Abbildung bildet die Occupancy Map der der Roboter zur Navigation nutzt. Die hellgrauen Bereiche zeigen dabei die festen Hindernisse der Umgebung, die vom Roboter nicht anfahren werden können. Im Gegensatz dazu sind alle dunkelgrauen Bereiche als befahrbar vermerkt.

wurde die Belegtheitskarte (Occupancy Map) verwendet, die vom Roboter kontinuierlich durch Informationen der Laserscanner aktualisiert wird. In ihr sind neben dem Grundriss der Testwohnung auch die festen Hindernisse, wie z.B. Sessel, Tische und Schränke, eingetragen. Durch diese enthalten Informationen kann die Occupancy Map

zur Navigation verwendet werden. Desweiteren zeigt die Abbildung die Positionen der Bewegungsmelder sowie deren Ausrichtung. Es ist zu erkennen, dass durch die Öffnungswinkel der Sensoren von 90° die Erfassung großer Teile der Wohnung gelingt. In einigen Räume führt die Verwendung von zwei Sensoren pro Bereich zur Überlapung der Erfassungsbereiche. Zusätzlich sind in der Abbildung auch die festgelegten Beobachtungspositionen dargestellt, die der Roboter bei seiner Suche durch die Wohnung einnimmt. Dabei geben die Pfeile an, welche Orientierung der Roboter an diesen Positionen hat.

Dem Vorteil der einfachen Installation durch batteriebetriebene Funksensoren steht der Nachteil gegenüber, dass der Sensor nicht beliebig schnell nacheinander auslösen kann. Die Zeit (Latenz) zwischen zwei aufeinander folgenden Events kann in 16 Stufen in einem Bereich zwischen 6 Sekunden und 16 Minuten eingestellt werden. Das verlängert zwar zum einem die Laufzeit der Batterie, führt aber zum anderen dazu, dass selbst bei minimal eingestellter Latenz, manche Bewegungen vom Sensor nicht erfasst werden, wenn sie innerhalb der Latenz erfolgen. Damit bestimmt die Latenz der Sensoren die zeitliche Auflösung des Bewegungserfassungssystems. Praktisch bedeutet dies, dass schnelle Bewegungen zwischen den verschiedenen Bereichen der Wohnung eventuell nicht richtig erfasst werden könnten. Die Sequenz der Sensorevents würde nicht die tatsächliche Bewegungsfolge wiedergeben, da bereits ausgelöste Sensoren eine erneute Bewegung noch nicht wieder erfassen können. Dadurch kann es passieren, dass schnelle Bewegungsfolgen nicht richtig erfasst werden und virtuelle Übergänge zwischen Bereichen entstehen können, die nicht benachbart sind.

Die einzelnen Bewegungsmelder kommunizieren mit einem Empfängermodul. Wird ein Sensor ausgelöst sendet er seine ID. Zusätzlich enthält jedes Event einen Zeitstempel, anhand dessen der zeitliche Aktivierungsverlauf der Sensoren ausgewertet werden kann. Ohne Hindernisse, die das Signal schwächen, gelingt eine Kommunikation zwischen Sender und Empfänger bis zu einem Abstand von 200 m . Auch in der Testwohnung ist das Signal stark genug, um durch Wände und geschlossene Türen empfangen zu werden.

6.2 Durchführung des Funktionstests

Der Funktionstest wird anhand von fünf unterschiedlichen Szenarien durchgeführt. Die ersten beiden Szenarien beziehen sich auf liegende Personen, da der Fokus bei der Entwicklung des Services auf Situationen lag, wie sie durch Schlaganfälle, Stürze oder Herzinfarkte entstehen können.

Aus ersten Probeläufen ist bekannt, dass der Roboter zum Erreichen seiner Ziele möglichst kurze Wege wählt. Hat die Pfadplanung mehrere freie Wege identifiziert, wird stets der direkte Weg gewählt. Daher ist bekannt auf welchen Wegen sich der Roboter auf der Fahrt zum Ziel bewegen wird. Beim ersten Szenario wird dieses Kenntnis genutzt, damit sich der Proband so positioniert, dass er sich direkt auf dem Weg des Roboters befindet. Der Roboter wird durch dieses neue, dynamische Hindernis gezwungen, eine alternative Route zu planen. Gleichzeitig wird er stehen bleiben oder zumindest seine Fahrt verlangsamen. Damit steigt die Chance, dass der Proband als gestürzte Person durch das Detektionsmodul erkannt wird.

Beim zweiten Szenario wird sich der Proband so positionieren, dass er nicht den potentiellen Weg des Roboters versperrt. Dadurch wird der Roboter nicht gezwungen, seine Fahrt zu verlangsamen. Mit diesem Szenario wird überprüft, ob auch unter diesen Bedingungen Personen erkannt werden und der Dialog gestartet wird.

Das dritte und vierte Szenario entspricht dem des ersten und zweiten, nur dass hier der Proband stehen soll. Hintergrund ist, dass im späteren Einsatz gelegentlich Fehlalarme ausgelöst werden können. In diesen Fällen muss der Nutzer gefunden und angesprochen werden, um einen Fehlalarm zu bestätigen und das Absetzen eines Notrufes zu verhindern. Deshalb ist der Roboter in der Lage, auch stehende Personen zu erkennen. Das dritte Szenario dient daher dazu, zu überprüfen, ob der Roboter stehende Personen erkennt, wenn sie sich in seinem Weg bei der Zielfahrt befinden. Auch hier könnte die Unterbrechung der Fahrt zu einer besseren Erkennung einer Person führen. Ob stehende Personen auch im Vorbeifahren erkannt werden, wird im vierten Szenario getestet.

Im fünften Szenario wird überprüft, ob der Roboter sitzende Personen erkennt. Es zeigte sich bereits, dass der Detektor für stehende Personen unter bestimmten Um-

ständen auch für sitzende Personen genutzt werden kann. Es wird getestet, ob der Detektor auch ausreicht, um sitzende Personen richtig zu erkennen. In diesen Fällen würde der Roboter sitzende Personen wie stehende Personen behandeln. Der Roboter geht dabei davon aus, dass kein Notfall vorliegt und der Dialog dient dem Abwenden eines Fehlalarms. Tabelle 6.1 listet die genannten Szenarien zusammen mit einer Kurzbeschreibung auf.

Szenario	Beschreibung
L 1	Liegende Personen auf der Route des Roboters
L 2	Liegende Personen neben der Route des Roboters
ST 1	Stehende Person auf der Route des Roboters
ST 2	Stehende Person neben der Route des Roboters
S	Sitzende Person

Tabelle 6.1: Szenarien des Funktionstests zur Suche von liegenden, stehenden und sitzenden Personen

Für die genannten Szenarien sind jeweils fünf Durchläufe geplant. Im späteren Einsatz kann es vorkommen, dass sich der Roboter an verschiedenen Orten der Wohnung befindet, wenn ein Notfall signalisiert wird. Gleichzeitig kann sich auch die Person an unterschiedlichen Orten befinden. In der Testung wird deshalb die Kombination aus Startposition des Roboters und Position der Person variiert. Die Testperson positioniert sich deshalb so in den Bereichen der Wohnung, dass für jedes Szenario jeder Bereich mindestens einmal ausgewählt wird. Indem die Testdurchläufe ohne Unterbrechung hintereinander durchgeführt werden, variiert auch die Startposition des Roboters. Die Position, an der die Suchfahrt beendet wurde (ob erfolgreich oder nicht) wird zur Startposition der neuen Suchfahrt. Zusätzlich führen die Kombinationen aus wechselnden Start- und Endpunkten dazu, dass der Roboter den Bereich, in dem sich die Testperson befindet, auf unterschiedlichen Routen betritt. Dies könnte ebenfalls die Erkennung der Person beeinflussen.

Im späteren Einsatz, wird die Notfallbehandlung durch das Auftreten eines der beiden Alarmevents gestartet. In beiden Fällen erfolgt ein Auslösen durch das Erreichen einer

zeitlichen Schwelle. Unter Testbedingungen, kann nicht auf das Erreichen der Zeitschwelle gewartet werden. Dafür wurde an den NOACTIVITYDETECTOR eine Funkfernbedienung des Typs *ELV FS 20 S4* angeschlossen und für den Empfang der Daten eingerichtet. Bei Betätigung des Handsenders wird ein Event generiert, wie es auch bei Überschreiten der Schwelle durch eine lange Inaktivität in der Wohnung ausgelöst werden würde. Der weitere Ablauf des Services bleibt dadurch unverändert. Mit dem Handsender kann die Testperson die Suche starten nachdem sie sich positioniert hat.

Eine Suche gilt als erfolgreich, wenn der Roboter eine Person gefunden hat und daraufhin die Fahrt stoppt. Das ist der Fall, wenn der Roboter den situationsabhängigen Dialog startet. Während aller Durchläufe wird die Zeit gemessen werden, die der Roboter zum Auffinden und Ansprechen benötigt. Zusammen mit den Beobachtungen kann eine erste Aussage zu Faktoren, die die Suchzeit beeinflussen, gemacht werden. Der durchgeführte Funktionstest dient als qualitative Beschreibung des in dieser Arbeit entwickelten Services zur Notfallbehandlung. Die Funktionstests werden mittels erstellter Versuchsprotokolle dokumentiert. Diese enthalten neben der Kennung des Szenarios (siehe Tabelle 6.1) und der laufenden Versuchsnummer, den Startbereich des Roboters und den Bereich, in dem sich die Testperson befindet. Des Weiteren wird die Zeit erfasst werden, die der Roboter für die Suche benötigt.

Im Protokoll wird zudem vermerkt, ob die Person gefunden wurde und ob der Roboter dabei erreichbar war. Darunter ist zu verstehen, dass sich der Roboter nach Erkennen der Person in einem Abstand von maximal einer Armlänge befinden soll, um eine Interaktion von Mensch und Roboter zu ermöglichen. Besonders nach einem Sturz muss sich der Roboter in einem erreichbaren Bereich befinden, damit eine Person, die nicht aus eigener Kraft aufstehen kann, einen Notfall bestätigen kann. Im Gegensatz dazu kann eine stehende Person gegebenenfalls einige Schritte auf den Roboter zugehen, um auf die Frage des Roboters zu reagieren. Um eine einheitliche Aussage zur Erreichbarkeit des Roboters liefern zu können, wird der Bereich um die liegende Person in acht Sektoren unterteilt. Abbildung 6.2 zeigt die gewählte Unterteilung.

Verlief die Suche einer liegenden Person erfolgreich, wird im Protokoll vermerkt, in welchem Sektor sich der Roboter zu dieser Zeit befand. Insbesondere bei liegenden

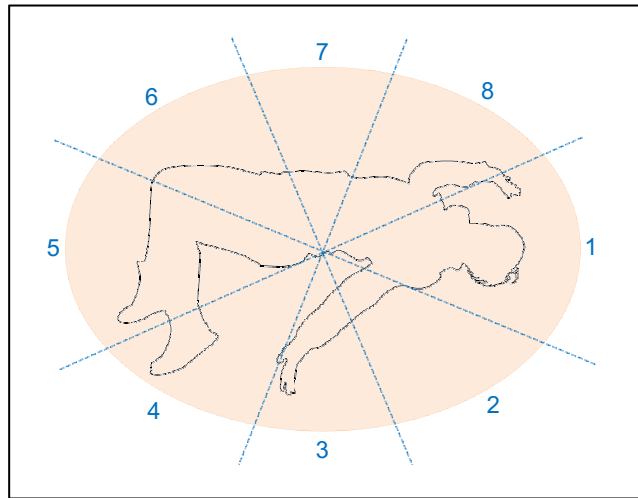


Abbildung 6.2: Sektoren um die liegende Person zur Beurteilung der Erreichbarkeit des Roboters.

Personen ist es wichtig, dass sich der Roboter in den Bereichen rund um den Oberkörper befindet. Nur so kann der Nutzer nach einem Sturz den Roboter erreichen. Für die Versuche mit stehenden und sitzenden Personen wird nicht von der Unterteilung des Bereiches in Sektoren Gebrauch gemacht. In diesen Fällen wird davon ausgegangen, dass es sich in erster Linie nicht um Notfälle handelt und dass sich die Person aktiv auf den Roboter zubewegen könnte.

6.3 Ergebnisse und Ergebnisdiskussion

Im Folgenden werden die Ergebnisse der durchgeführten Funktionstests präsentiert und diskutiert. Die erstellten Versuchsprotokolle finden sich im Abschnitt A des Anhangs.

1. Die Integration der in dieser Arbeit entstandenen Module zu einem funktionierenden Service war erfolgreich.

Es zeigt sich, dass das Zusammenspiel des EMERGENCYSEARCH-Behaviors zur Steuerung der Suchfahrt und des VERIFYFALLENPERSON-Behavior, welches Positionen um die vermeintliche Person festlegt, bis das Vorhandensein einer Person mit hoher Wahr-

scheinlichkeit bestätigt werden konnte, problemlos funktioniert.

Der Roboter erkannte, ob es sich in den vorliegenden Situationen um liegende oder stehende Personen handelte und führte den entsprechenden Dialog aus.

2. In 18 der 25 durchgeführten Szenarien konnte die Person durch den Roboter gefunden werden.

3. Im Durchschnitt dauerten die erfolgreichen Suchfahrten 59,9 s.

Die längste Suchfahrt des Roboters dauerte nicht länger als 175 s (Versuch ST2-3). Die kürzeste Suche wurde nach 14 s beendet (Versuch L2-5).

Dabei ist anzumerken, dass diese Zeiten von vielen Faktoren beeinflusst werden. Liegen die Bereiche, in denen die Person liegt und der Roboter seine Suche startet, direkt nebeneinander, benötigt der Roboter für die Suche weniger Zeit, als wenn er zuvor durch andere Bereiche fahren muss.

Auch die Erreichbarkeit eines Ortes spielt eine Rolle. Wird die Person während der Suchfahrt nur mit einer schwachen Hypothese erkannt, der Roboter ist aber aufgrund von Hindernisse nicht in der Lage, weitere Positionen um die Person herum für eine sichere Hypothese zu erreichen, verlängert sich die Suchzeit bis eine geeignete Position gefunden wurde.

In Bezug auf die Situaionen, in der der Roboter längere Zeit brauchte, um seine festgelegten Beobachtungspositionen einzunehmen, wurde wird vermutet, dass sich der Roboter in diesen Fällen leicht delokalisiert hatte. Dadurch erschienen Hindernisse näher am Roboter als sie in Wirklichkeit waren. Erst durch vorsichtiges Vorwärts- und Rückwärtsfahren bzw. Drehen konnte er die Positionen erreichen. Vor allem, wenn sich die Beobachtungspositionen relativ nah an Hindernissen befinden, kann eine Unsicherheit in der Lokalisation zu verlängerten Zielfahrten führen.

Ein weiterer Faktor, der zur Verlängerung der Suchzeit beitragen kann, ist die Reihenfolge, in der die Ziele angefahren werden sollen. Diese wird aus den Übergangswahrscheinlichkeiten der Bereiche erzeugt (vgl. Kap. 5.4). Da sich die zugrunde liegenden Markov Mobility Chains jedoch über die Laufzeit durch die erfassten Bewegungsevents

ständig verändern, sind deren Ergebnisse nur schwer zu reproduzieren. Treten bestimmte Übergänge zukünftig häufiger auf als zuvor, nehmen diese Übergänge auch in der erzeugten Reihenfolge einen anderen Platz ein.

Insgesamt muss bedacht werden, dass die Suchzeit, selbst wenn sie über alle Beobachtungspositionen der Bereiche führt, nur einen geringen Anteil an der Gesamtzeit der Notfallbehandlung hat. Die Dauer vom Zeitpunkt, zu dem der Notfall eintritt bis dieser erkannt wird, wird dabei deutlich größer als die Dauer für das Suchen und Finden der Person.

4. Die Erfassung der Bewegung mittels PIR-Sensoren funktionierte problemlos.

Der Proband löste während der Positionierung die PIR-Sensoren in den entsprechenden Bereichen aus. Dadurch konnte der Roboter seine Suche im Bereich der letzten Bewegung beginnen. Das führte zu kurzen Suchzeiten, wenn der Roboter die Person in diesem Bereich fand.

5. Die Nutzung der Markov Mobility Chains wird als sinnvoll erachtet.

Durch die genutzten Markov Mobility Chains kann die Suche effektiver gestaltet werden, da zuerst an den Orten nach der Person gesucht wird, an denen die Aufenthaltswahrscheinlichkeit hoch ist. Eine willkürliche Suchfahrt (z.B. durch eine festgelegte Reihenfolge) kann diese Zusammenhänge nicht berücksichtigen. Insbesondere wenn die Personenbewegungen durch die PIR-Sensoren nicht genau erfasst werden, hilft dieses Vorgehen bei einer zielgerichteten Suche.

Teilweise konnte die Person vom Roboter nicht sofort in dem Bereich gefunden werden, in dem sie das Sensorevent auslöste. In diesen Fällen war es hilfreich, dass neben dem Bereich des letzten Sensorevents auch die anderen Bereiche nacheinander überprüft wurden. Die Festlegung, in welcher Reihenfolge die weiteren Positionen angefahren werden, wurde dabei aus den erzeugten Markov Mobility Chains bestimmt. Der Roboter setzte daraufhin seine Suche an den durch die Reihenfolge festgelegten Bereichen fort. Dadurch konnte der Roboter die Person schließlich detektieren, da er den Bereich beim Verlassen oder erneuten Befahren aus einer anderen Perspektive wahrnahm.

Allerdings muss angemerkt werden, dass in diesem Fall auch eine willkürliche Suche

dazu geführt hätte, die Person zu finden, da der Roboter lediglich aus verschiedenen Perspektiven betrachten musste, um sie zu finden.

6. In 9 von 10 Versuchen wurden liegende Personen durch den Roboter gefunden.

In 8 von 10 Durchläufen konnte schon auf der ersten Position, die der Roboter für die Verifikation einnahm, die liegende Person bestätigt werden, so dass das Anfahren weiterer Beobachtungspositionen um die Person herum, nicht notwendig wurde.

Der Übergang vom EMERGENCYSEARCH-Behavior zum VERIFYFALLENPERSON-Behavior erfolgte, sobald der *Score* $> 0,3$ betrug. Wurde im Verifikationsschritt ein *Score* $> 0,9$ erreicht, galt die liegende Person als bestätigt und der Dialog wurde eingeleitet. Diese Schwellen wurden während der Tests nicht verändert.

Insbesondere wenn die Person quer vor dem Roboter liegt, ist die Erkennungsleistung hoch, sodass keine weitere Verifikation benötigt wird. Es gibt jedoch einige Fälle (siehe Funktionstests L2-3 und L2-5) bei denen die Verifikation erst nach Einnehmen der 2. bzw. 3. Position erfolgreich ist. Durch Anfahren mehrerer Positionen bewegt sich der Roboter entlang der Silhouette der Person bis diese quer vor ihm liegt und die Erkennung gelingt.

Dadurch, dass unter Umständen mehrere Positionen um die Person eingenommen werden, kann die Schwelle für eine sichere Personenhypothese sehr hoch angesetzt werden (*Score* $> 0,9$). Auf diese Weise starten alle folgenden Abläufe erst, wenn eine Person absolut sicher bestätigt wurde.

7. Das Setzen weiterer Beobachtungspositionen ist sinnvoll, um einen Bereich komplett zu erfassen.

Die alleinige Fahrt zu einer Beobachtungsposition pro Bereich genügt nicht, um Personen sicher zu finden. Der Vorteil mehrerer Beobachtungspositionen in einem Bereich ist, dass der Roboter den Raum aus verschiedenen Perspektiven betrachten kann. Dadurch erhöht sich die Wahrscheinlichkeit, eine Person zu finden, die von nur einer Position nicht erkannt werden würde. Insbesondere wenn der zu überprüfende Bereich sehr groß und der Erfassungsbereich des Roboters beschränkt ist, kann auf diese Weise sichergestellt werden, dass dennoch alle Bereiche des Raumes betrachtet werden.

Der gegenwärtige Ablauf sorgt dafür, dass der Roboter verschiedene Bereiche der Wohnung überprüft, wobei er sich zwischen den Zielpunkten der unterschiedlichen Bereiche bewegt und diese aus verschiedenen Perspektiven erfasst. Das Setzen mehrerer Ziele in einem Bereich gewährleistet eine großflächige Erfassung der Umgebung und stellt damit eine sinnvolle Ergänzung für die zukünftige Entwicklung dar.

8. Der Erfassungsbereich der Tiefenkamera ist zu gering.

Der eingeschränkte Erfassungsbereich der Tiefenkamera von $1,5\text{ m}$ führte auch zu einem Verhalten, das im Vorfeld nicht abzusehen war. In der Konzeption des Services zur Notfallbehandlung wurde davon ausgegangen, dass der Erfassungsbereich der Kamera deutlich größer ist und eine liegende Person schon von Weitem erkannt wird.

Wäre die Tiefenkamera so eingestellt, dass sie einen größeren Bereich erfassen kann, hätte der Roboter im Szenario L1-1 die liegende Person bereits aus weiterer Entfernung erkannt und nicht erst nachdem die Beine der Person die Fahrt des Roboters unterbrochen. Durch die frühe Erkennung würde sich der Roboter bei der Anfahrt an Mittelpunkt der Personenhypothese orientieren. Die direkte Fahrt zum Schwerpunkt würde die Erkennungsleistung zusätzlich positiv beeinflussen, da die betrachtete Szene nicht nur die Beine der Person, sondern einen Großteil des Körpers enthalten würde. Um die Ergebnisse der Suche und der Verifikation zukünftig zu verbessern, muss die robuste Detektion liegender Personen auf größere Erfassungsbereiche übertragen werden. Gleichzeitig kann die, bereits genannte, Erzeugung weiterer Beobachtungspositionen pro Bereich, zusammen mit einem größeren Erfassungsbereich des Roboters, dazu führen, dass durch verschiedene Perspektiven auch schwierig zu erkennende oder ungünstig liegende Personen gefunden werden.

9. In 8 von 9 erfolgreichen Suchen konnte die liegende Person durch Berührung mit dem Roboter Kontakt aufnehmen.

Während solch einer Suche befand sich der Roboter in einem der Sektoren, die als für die Person erreichbar eingestuft wurden (vgl. Kap. 6.2), obwohl der entwickelte EMERGENCYSERVICE über kein Modul verfügt, welches die explizite Fahrt zu einer gut erreichbaren Position steuert. In diesen Fällen führte die Lage der Person sowie die Befahrbarkeit der Verifikationspositionen dazu, dass die Roboter häufig problemlos

zu erreichen war. Da die Hypothesen meist sicherer werden, wenn der Roboter die liegende Person längs vor sich sieht, kann davon ausgegangen werden, dass der Roboter oft weitere Verifikationspositionen einnehmen wird, wenn er die Verifikation vom Kopf oder den Füßen der Person beginnt, bis er eine sichere Personenhypothese hat. Dadurch bewegt er sich seitlich an der Person entlang, wodurch die Chance steigt, dass er für die Person erreichbar wird.

10. In 8 von 10 Versuchen war die Suche nach stehenden Personen erfolgreich.

Es wurden Personen auf und abseits der Route des Roboters detektiert.

Durch die verwendete 180°-Fischaugenkamera vergrößert sich der Bereich, der visuell erfasst werden kann, gegenüber dem eingeschränkten Erfassungsbereich zur Erkennung liegender Personen. Dadurch konnten auch Personen gefunden werden, die seitlich neben dem Roboter standen, während dieser vorbeifuhr.

Für die zukünftige Anwendung wird es sich zusätzlich positiv auf die Erkennung stehender Personen auswirken, dass sich die Nutzer, sofern sie stehen und kein Notfall vorliegt, in Richtung des Roboters drehen, sobald dieser den Raum befährt. Außerdem kann angenommen werden, dass sich die Personen auf den Roboter zubewegen, wenn sie hören, dass dieser sie sucht. Der dafür gewählte Dialog (vgl. Kap 5.6) wird beim Start jeder Suche ausgegeben und signalisiert dem Nutzer, dass dieser gesucht wird. In beiden Fällen, wird die Aufmerksamkeit auf den Roboter gerichtet und der Nutzer wird animiert, sich auf den Roboter zu zeigen und vor ihn treten, um die Behandlung des vermeintlichen Notfalls zu beenden. Dadurch wird die Erkennung der Person stark vereinfacht. Die Tests zeigten, dass auch ohne aktive Bewegung in die Richtung des Roboters, die Erkennungsleistung stehender Personen hoch ist.

11. In 1 von 5 Versuchen konnte der Roboter eine sitzende Person erkennen.

Ein Grund für die geringe Erkennungsleistung könnte sein, dass sich die Sitzgelegenheiten nicht auf den Routen des Roboters befanden. Stattdessen wurden sie als feste Objekte in die globale Karte integriert, sodass der Roboter seinen Weg an diesen Objekten vorbei plante. Im Gegensatz zu dynamischen Hindernissen muss der Roboter bei festen Hindernissen nicht anhalten und seine Route neu planen. Somit fährt der

Roboter an den sitzenden Personen vorbei, ohne diese zu erkennen, da sie sich nicht im direkten Sichtbereich befinden. Die Erkennung sitzender Personen ist jedoch ein wichtiger Bestandteil des Serviceroboters, da im Alter viele Menschen auf einen Rollstuhl angewiesen sind oder sich generell weniger bewegen.

Daher muss in zukünftigen Erweiterungen des Services ein zuverlässiges Erkennungsmodul für sitzende Personen enthalten sein.

Ein weiterer Grund für die schlechte Erkennungsleistung kann die Verdeckung der sitzenden Personen durch Lehnen oder andere Teile des Mobiliars sein.

Auch bei sitzenden Personen lässt sich der Dialog nutzen, um die Erkennung indirekt zu verbessern. In diesen Fällen kann die Ausgabe des initialen Satzes zu Beginn der Suche dafür sorgen, dass der Bewohner aufsteht und sich dem Roboter zeigt sodass er als stehender Mensch erkannt wird. Wie bereits unter Punkt 10 der Ergebnisse erwähnt, kann die Sprachausgabe und das Erscheinen des Roboters im Raum dafür sorgen, dass sich der Bewohner aufrecht hinsetzt und sich dem Roboter zuwendet, um die Erkennung zu verbessern.

12. Das Auslösen eines unspezifischen Alarms stellt ein sinnvolles Feature dar.

Wie sich zeigte, konnte in einigen Versuchen (vgl. L1-3, ST1-5, ST2-4 sowie S2 bis S5) die Testperson nicht durch den Roboter gefunden werden. Obwohl jeder Fehlalarm vermieden werden soll, ist es von großer Bedeutung, von einem Notfall auszugehen und einen Alarm abzusetzen, weil die medizinische Versorgung des Nutzers an erster Stelle steht. Dieses Event vom Typ *NoPersonFound* dient der Auslösung eines unspezifischen Alarms. Während der Funktionstests wären somit sieben unspezifische Alarme ausgelöst worden.

Damit ein unspezifischer Alarm nicht mit derselben Dringlichkeit wie ein bestätigter Alarm behandelt wird, könnte zukünftig ein unspezifischer Alarm an Angehörige, Nachbarn oder die Pflegeeinrichtung gesendet werden, während ein bestätigter Alarm direkt an eine Rettungsstelle geleitet wird.

13. Die kapazitiven Sensoren am Gehäuse erleichtern die Kommunikation mit dem Roboter.

Für die Szenarien mit liegenden Personen wurde der ausgegebene Dialog so konzipiert, dass der Nutzer über die kapazitiven Sensoren am Gehäuse des Roboters den Dialog steuern kann. Dadurch können Nutzer in Notfällen schnell Hilfe rufen, ohne die entsprechenden Buttons auf dem Display suchen und betätigen zu müssen.

In allen Versuchen erfassten die Sensoren die Berührungen im unteren Bereich des Roboterkörpers richtig. Im oberen Bereich des Roboters registrierten die Sensoren jedoch keine Berührungen als Antwort. Aufgrund der Gestaltung des Robotergehäuses ist derzeit nicht zu erkennen, in welchem Bereich ein Nutzer den Roboter berühren muss, um die Frage nach Hilfe zu beantworten. Hierzu kann das Design entsprechend angepasst werden, sodass die Bereiche, in denen sich die Sensoren befinden, gekennzeichnet sind. Darüber hinaus sollte die Sensitivität der oberen Sensoren geprüft werden. Alternativ könnte auch der im Moment genutzte Dialog geändert werden. Der Roboter kann in diesem Fall dem Nutzer mitteilen, dass eine Berührung im unteren Bereich erfolgen muss, um Hilfe zu rufen.

14. Die Zeit zwischen den Teildialogen bei gestürzten Personen ist zu lang.

Die Versuche zeigten, dass die Zeit zwischen den beiden Dialogen, die an eine gestürzte Person gerichtet werden zu lang ist. Im Moment beträgt sie 60 s. Erfolgt auf die erste Frage nach Hilfe keine Reaktion, dauert es 60 s bis die nächste Frage gestellt wird. Es ist anzunehmen, dass der Nutzer nicht den Zusammenhang zwischen der unbeantworteten ersten und der folgenden zweiten Frage erkennt, wenn diese erst nach einer Minute gestellt wird.

Aus diesem Grund wird empfohlen, den BehaviorManager, der für die Überwachung der Timeouts zuständig ist, dahingehend zu ergänzen, dass Timeouts beliebiger Länge einstellbar sind. Im beschriebenen Fall wird eine Wartezeit von 20 s als ausreichend erachtet.

15. Durch Überwachung der Roboterpositionen während der letzten 3 s konnte die Zielanfahrt verbessert werden.

In Kapitel 5.5 wurde beschrieben, dass der Roboter in einer Endlosschleife lief, als er versuchte, Positionen nahe der Person zu erreichen, die in der Planung als erreichbar angenommen wurden, tatsächlich aber durch die Person belegt wurden. Das Ergebnis war, dass der Roboter versuchte, diese Positionen anzufahren, während die Kollisionsvermeidung die Fahrt jedes Mal unterbrach. Das ständige Fahren und Stoppen ließ den Roboter in der Nähe seines Ziels „zappeln“, ohne dass er diesen Zustand beenden konnte. Durch die Überwachung der Positions- und Winkeländerungen während der letzten 3 s konnte dieses Verhalten abgestellt werden. Der Roboter nähert sich nun so weit wie möglich an die Person an, beendet jedoch den Versuch, die Zielposition zu erreichen, wenn die Analyse der Positions- und Winkeldifferenzen auf ein beginnendes „Zappeln“ hindeuten.

16. Die Reihenfolge der angefahrenen Ziele war nicht immer plausibel.

Wie bereits in Kapitel 5.4 beschrieben, startet der Roboter seine Suche im Bereich des letzten Sensorevents. Es fiel jedoch auf, dass die Reihenfolge, in der die weiteren Bereiche angefahren wurden, nicht immer mit der erwarteten Reihenfolge übereinstimmte. So fuhr der Roboter im Versuch L2-1 die Bereiche in der folgenden Reihenfolge ab: BP 2, BP 5, BP 6, BP 4 und BP 1 (vgl. Abbildung 6.1, S. 59). Ausgehend von der ersten Zielposition BP 2 wäre als nächstes die Fahrt zu Position BP 1 oder BP 4 zu erwarten gewesen. Die Fahrt zur Position BP 3 wäre möglich, aber unwahrscheinlich, aufgrund der wenigen erfassten Übergänge. Ein Grund für die von den Erwartungen abweichende Reihenfolge könnte sein, dass das LivingLab vor und während der Versuchsdurchführung von den Mitarbeitern des Fachgebietes betreten werden konnte. Die der Zielpositionsplanung zugrunde liegende Matrix der Übergangswahrscheinlichkeiten erfasst jede Bewegung in den Bereichen, unabhängig davon ob nur eine oder mehrere Person zu dieser Zeit anwesend sind. So könnte die Bewegung von zwei Personen in verschiedenen Bereichen dazu führen, dass zwischen diesen Bereichen scheinbare (virtuelle) Übergänge erfasst werden, die in der Realität nicht existieren. Die nicht plausible Reihenfolge könnte daher durch die zuvor erzeugten virtuellen Übergänge

entstanden sein.

In einer zukünftigen Anwendung, in der eine Person allein in der Wohnung lebt, würden solche virtuellen Übergänge nicht entstehen, solange sich keine weiteren Personen in der Wohnung befinden. Würden virtuelle Übergänge (z.B. durch Besuch) erzeugt werden, würden die dynamischen Markov Mobility Chains dafür sorgen, dass sich mit der Zeit wieder die realen Übergänge stärker ausprägen, sobald die Person alleine ist (vgl. Kap. 5.3).

17. Die Anordnung der PIR-Sensoren hat einen Einfluss auf die Erzeugung der Markov Mobility Chains.

Es sollte darauf geachtet werden, dass die Sensoren so in den Bereichen angebracht sind, dass sich deren Erfassungsbereiche nicht überlagern. Gleichzeitig muss bei der Installation darauf geachtet werden, dass dennoch der gesamte Bereich erfasst wird. Aufgrund der großen Erfassungsbereiche und Öffnungswinkel genügt in vielen Fällen ein Sensor, der in einer Raumecke installiert ist. Werden mehrere Sensoren pro Bereich genutzt, kann eine eindeutige Zuordnung einer Bewegung zu einem Teilbereich nicht erfolgen. Die Nutzung mehrerer Sensoren pro Bereich führt daher nicht zwangsläufig zur Verbesserung der Suchstrategie. Soll ein großer Bereich durch mehrere Sensoren abgedeckt werden, müssen diese so angeordnet werden, dass eine Bewegung jeweils nur von einem Sensor erfasst wird.

Zusätzlich sollten die Sensoren so platziert werden, dass sie nur die Bewegungen im eigenen Bereich, nicht aber im benachbarten, erfassen. In der Testwohnung (vgl. Abb. 6.1, S. 59) wurde die Bewegung einer Person, die sich im Esszimmer nahe der Tür zur Küche aufhielt, von den Sensoren S1 und S4, möglicherweise auch von S2, erfasst. Daher sollte die Position der Sensoren so gewählt werden, dass eine Erfassung der Bewegungen aus anderen Bereichen (z.B. durch offene Türen) nicht möglich ist.

18. Während der Versuche gab es keine Fehldetektionen.

Während der Tests konnten keine Fehldetektionen durch den Roboter bemerkt werden. Fehldetektionen wären beispielsweise, dass der Roboter fälschlicherweise Gegenstände für Menschen hält und diese anspricht oder das Verifikationsverhalten durchführt. Es

konnte ebenfalls nicht beobachtet werden, dass der Roboter eine liegende Person als stehende erkannte und den falschen Dialog startete.

Kapitel 7

Zusammenfassung und Ausblick

Nachdem im vorangegangenen Kapitel die Durchführung der Funktionstests und die dabei beobachteten Ergebnisse dargelegt wurden, soll in diesem Kapitel eine Zusammenfassung der Masterarbeit gegeben werden (Kapitel 7.1). Im Anschluss daran wird auf mögliche Entwicklungen eingegangen, mit denen das Konzept der Notfallbehandlung zukünftig erweitert werden kann (vgl. Kapitel 7.2).

7.1 Zusammenfassung

In der vorliegenden Arbeit wurde das Ziel verfolgt, eine Notfallstrategie für einen Service-Roboter zu entwickeln, die die gesamte Handlungskette vom Erkennen eines Notfalls über die Suche und Verifikation der Person in der Wohnung bis hin zu einem Dialog mit der Person an dessen Ende gegebenenfalls ein Notruf steht, enthält.

Nachdem in den Kapiteln 2 und 3 die aktuellen Konzepte zur Erkennung von Notfällen und der Bestimmung der Personenposition sowie die Grundlagen des in dieser Arbeit verwendeten MIRA-Frameworks vorgestellt wurden, behandelt das Kapitel 5 die Entwicklung und Integration von Modulen, die zusammen die Notfallstrategie des Service-Roboters bilden.

Die Erkennung von Notfällen im EMERGENCYSERVICE basiert auf der Auswertung von Bewegungen, die von Bewegungsmeldern, die in der Wohnung installiert wurden, erfasst werden. Dabei werden zwei Szenarien unterschieden. Einerseits ein Notfall, wie

er z.B. durch einen Herzinfarkt oder Schlaganfall entstehen kann. Die Person ist dabei ohnmächtig und regungslos. Die dadurch auftretenden Phasen von Inaktivität in der Wohnung werden vom Modul `NOACTIVITYDETECTOR` erkannt und versetzen den Roboter in Alarmbereitschaft. Andererseits gibt es Notfälle, wie sie durch einen Sturz entstehen. In diesem Fall ist die Person nicht zwangsläufig regungslos, kann aber den Bereich, in dem sie sich gerade befindet, nicht verlassen. Das erzeugte Modul `IMMOBILITYDETECTOR` erkennt einen Notfall, sobald sich eine Person für eine untypisch lange Zeit in einem Bereich aufhält. Die als zulässig erachteten Zeiten werden dabei aus bereichsspezifischen Beobachtungen der Vergangenheit erzeugt.

Als Reaktion auf erkannte Phasen untypisch langer Inaktivitäten bzw. Immobilitäten folgt der Ablauf des `EMERGENCYSERVICE`, der im Folgenden noch einmal dargestellt werden soll.

Wird der Roboter durch eines der genannten Module in Alarmbereitschaft versetzt, soll im nächsten Schritt die Person in der Wohnung gesucht werden. Dafür wurde in dieser Arbeit das `EMERGENCYSEARCH`-Behavior entwickelt. Für die Personensuche werden die einzelnen Bereiche der Wohnung überprüft. Zu diesem Zweck wird eine Liste von vordefinierten Zielpositionen in der Wohnung erzeugt. Die Zielpositionen sind dabei so gewählt, dass von ihnen die jeweiligen Bereiche der Wohnung gut überblickt werden können. Um die Personensuche effizient und zielgerichtet zu gestalten, ist die Reihenfolge der zu überprüfenden Zielpositionen nicht fest vorgegeben. Mit jedem ausgelösten Alarm wird sie neu erzeugt. Dabei befindet sich das erste Ziel in dem Bereich, aus dem das letzte Bewegungsevent empfangen wurde und somit die Person zuletzt registriert wurde. Da es jedoch sein kann, dass ein Sensor eine Bewegung nicht erfasst hat, z.B. aufgrund der Sensorlatenzen (vgl. Kap. 6.1) oder eines Defekts des Sensors, muss die Person auch in den restlichen Bereichen der Wohnung gesucht werden. Die Reihenfolge, nach der diese Bereiche überprüft werden, hängt dabei von den erfassten Übergangswahrscheinlichkeiten vom Bereich des letzten Sensorevents in die anderen Bereiche ab. Der nächste Bereich in der Liste der Zielpositionen ist somit der Bereich, für den in der Vergangenheit die häufigsten Übergänge erfasst wurden. Die weiteren Bereiche in der Liste werden absteigend nach ihren Wahrscheinlichkeiten geordnet.

Für die Fahrt zu den Zielpositionen wird auf das bereits vorhandene NAVIGATION-Modul zurückgegriffen. Dieses übernimmt die Pfadplanung sowie die Navigation und die Kollisionsvermeidung. Während der Fahrt wird überprüft, ob sich im Bereich vor dem Roboter stehende oder liegende Personen befinden. Für die Erkennung stehender und liegender Personen werden die Module PERSONTRACKER und FALLENPERSON-DETECTOR genutzt. Wird eine Person erkannt, beendet dieses Event die Suchfahrt des EMERGENCYSEARCH-Behaviors, andernfalls wird die Suche entlang der Zielpositionen fortgesetzt.

Nachdem eine liegende Person durch das EMERGENCYSEARCH-Behavior gefunden wurde, soll in einem zusätzlichen Schritt überprüft werden, ob es sich tatsächlich um eine Person handelt. Zu diesem Zweck entstand innerhalb dieser Arbeit das VERIFYFALLENPERSON-Behavior. In diesem werden ausgehend vom Mittelpunkt der vermeintlichen Person weitere Beobachtungspositionen um diese herum erzeugt, die nacheinander vom Roboter angefahren werden. Die Betrachtung der Szene aus unterschiedlichen Perspektiven soll die Sicherheit einer Personendetektion erhöhen, bevor weitere Maßnahmen eingeleitet werden. Um eine gute Beobachtung zu ermöglichen, wurde das Verifikationsverhalten so konzipiert, dass der Roboter Positionen um die Person einnimmt, die möglichst nah an dieser sind. Auch für den folgenden Dialog ist es wichtig, dass sich der Roboter in einer von der Person erreichbaren Entfernung befindet. Während der Entwicklung wurde das Behavior vom Autor durch zwei zusätzliche Mechanismen erweitert. So wird zum einen überprüft, ob sich zwischen der Person und der Beobachtungsposition des Roboters ein Hindernis befindet, das die Sicht auf die Person versperrt. In diesem Fall wird diese Beobachtungsposition nicht angesteuert, sondern mit der Bestimmung der nächsten Position fortgefahren. Eine weitere Ergänzung betrifft die Erreichbarkeit von Beobachtungspositionen. Versucht der Roboter, eine Beobachtungsposition anzufahren, die durch den Körper der Person versperrt ist, erfolgt die Anfahrt unter Beachtung eines Mindestabstandes zur Person. Ohne den ergänzten Mechanismus würde der Roboter erfolglos versuchen, die belegte Position zu erreichen. Durch die Ergänzung ist die Beendigung der Zielfahrt und die Überprüfung weiterer Ziele gewährleistet. Konnte der Roboter die Beobachtungspo-

sition erreichen, wird der bereits genannte `FALLENPERSONDETECTOR` genutzt, um eine liegende Person zu erkennen. Bei hoher Sicherheit, dass es sich um eine Person handelt, wird das `VERIFYFALLENPERSON`-Behavior beendet und der Dialog gestartet. Die Konzeption des `VERIFYFALLENPERSON`-Behaviors sowie die Notwendigkeit der ergänzten Mechanismen werden in Kapitel 5.5 dargestellt.

Das Design der erstellten Dialoge wird in Kapitel 5.6 vorgestellt. Der am Ende des Service stehende Dialog dient der Kommunikation mit dem Nutzer. Mit ihm soll geklärt werden, ob es sich bei der vorliegenden Situation um einen Notfall handelt und ob Hilfe verständigt werden muss. Der Dialog wird gestartet, sobald eine stehende oder liegende Person während der Suche gefunden bzw. durch die Verifikation bestätigt wurde. Eine stehende Person wird durch den Roboter gefragt, ob ein Notfall vorliegt. Die Frage wird dem Nutzer visuell-akustisch übermittelt, indem eine Audioausgabe über einen Lautsprecher sowie eine Textausgabe auf dem Display des Roboters erscheint. Der Nutzer kann daraufhin einen Notfall über zwei Buttons auf dem Display bestätigen oder verneinen. Im Falle einer liegenden Person kann die Kommunikation zusätzlich zu den Buttons auf dem Display des Roboters auch über eine Berührung am Robotergehäuse erfolgen. Unterbleibt eine Reaktion des Nutzers nach der ersten Frage des Roboters (beispielsweise bei einer Ohnmacht), wird nach kurzer Zeit eine zweite Frage gestellt. Die Person hat wiederum die Möglichkeit, auf diese Frage zu reagieren. Unterbleibt erneut eine Reaktion, wird davon ausgegangen, dass es sich in diesem Fall um eine Notsituation handelt und ein Notruf wird abgesetzt. Da im späteren Einsatz des Systems gelegentliche Fehlalarme ausgelöst werden können, dient der Dialog als Filter, um Alarme von Fehlalarmen zu unterscheiden. Durch seine Struktur ist der entwickelte Dialog geeignet, Fehlalarme des Systems abzufangen, bevor sie aus der Wohnung zu Angehörigen oder Pflegestellten gelangen.

Die Gesamtheit aus `EMERGENCYSEARCH`-Behavior, `VERIFYFALLENPERSON`-Behavior und dem Dialog sowie den Modulen zur Notfallerkennung und der Detektion stehender bzw. liegender Personen bildet den `EMERGENCYSERVICE`, dessen Entwicklung Ziel dieser Arbeit war.

Zur Überprüfung des problemlosen Zusammenspiels aller Service-Module wurden

Funktionstests durchgeführt (vgl. Kap. 6). In verschiedenen Szenarien wurde die Eignung des Services zur Notfallbehandlung unter zukünftigen Einsatzbedingungen getestet. Der Fokus lag hierbei auf der Suche und Erkennung liegender, stehender und sitzender Personen. Des Weiteren wurde untersucht, ob der situationsabhängige Dialog gestartet wurde und ob der Roboter während des Dialogs erreichbar war.

Die Ergebnisse der Funktionstests zeigen, dass durch Integration von einerseits bestehenden Modulen als auch Modulen, deren Entwicklung Teil dieser Arbeit war, ein funktionierender und zuverlässiger Service zur Notfallbehandlung entstanden ist. Die Ergebnisse sind in vieler Hinsicht vielversprechend. In 9 von 10 Versuchen gelang es, liegende Personen auch unter schwierigeren Bedingungen in der Testwohnung zu finden. In 8 von 10 Versuchen war die Suche stehender Personen erfolgreich. Schwächen zeigte das System beim Finden sitzender Personen.

7.2 Ausblick

Die Durchführung der Funktionstests zeigte, in welchen Bereichen des EMERGENCY-SERVICE weitere Ergänzungen sinnvoll sein können, um den Ablauf zu verbessern. Die nachfolgend aufgeführten Überlegungen zu möglichen Erweiterungen sind dabei aus den diskutierten Ergebnissen des Funktionstests (vgl. Kap. 6.3) abgeleitet.

Ausblick bezüglich des eingeschränkten Erfassungsbereichs des Roboters

Der modulare Aufbau des entwickelten EMERGENCYSERVICE ermöglicht es, vorhandene Module durch andere zu ersetzen oder um Funktionalitäten zu ergänzen. Lediglich die verwendeten Schnittstellen zum Datenaustausch müssen erhalten bleiben bzw. entsprechend angepasst werden. Auf diese Weise kann der Service problemlos erweitert werden, ohne das Gesamtsystem von Grund auf neu zu konzipieren. Die nachfolgend aufgeführten Ideen zur Erweiterung lassen sich dadurch mit geringem Aufwand in den Service integrieren.

Die Detektion liegender Personen war aufgrund der verwendeten Tiefenkamera des Typs Asus Xtion Pro nur in einem eingeschränkten Bereich vor dem Roboter möglich.

Durch die Neigung, mit der die Kamera am Gehäuse des Roboters angebracht ist, reduziert sich der Erfassungsbereich auf ungefähr $1,5\text{ m}$ vor dem Roboter. Dadurch ist der Roboter nicht in der Lage, wie in der Konzeption vorgesehen, einen gesamten Bereich von einer Beobachtungsposition aus zu überblicken. Weiter als $1,5\text{ m}$ entfernt liegende Personen konnten daher nicht detektiert werden.

Da mit der Asus-Kamera die Erkennung liegender Personen auf größere Entfernung aufgrund von Rauschen in den Bilddaten ungenau wird, sollte in weiteren Arbeiten untersucht werden, ob der Algorithmus zur Detektion liegender Personen auch auf andere Tiefenkamerasysteme übertragen werden kann, um die Erkennungsleistung in größeren Szenen zu erhöhen. Ziel sollte es dabei sein, ganze Räume von einer Beobachtungsposition aus zu erfassen und darin anwesende Personen zu erkennen.

Eine weitere Möglichkeit, um dem Problem des geringen Erfassungsbereiches zu begegnen, wäre, das EMERGENCYSEARCH-Behavior zu erweitern. Anstatt der derzeitigen Suchstrategie, bei der pro Bereich eine Position angefahren wird, könnten zusätzliche Beobachtungspositionen definiert werden, damit auf diese Weise der zu überprüfende Bereich bestmöglich abgedeckt ist.

Eine andere Möglichkeit zum effektiven Absuchen eines Bereiches ist angelehnt an [VOLKHARDT und GROSS, 2013]. In diesem Konzept, sollen Personen in einer Wohnung gesucht werden, indem Beobachtungspositionen abhängig von der erwarteten Aufenthaltswahrscheinlichkeit einer Person an diesem Ort ermittelt werden. Von einer Position ausgehend, werden die in der Vergangenheit erfassten Aufenthaltswahrscheinlichkeiten über den Erfassungsbereich des Roboters integriert. Auf diese Weise wird die Position bestimmt, von der eine Personendetektion am wahrscheinlichsten ist. Dieses Konzept kann auf das EMERGENCYSEARCH-Behavior übertragen werden. Hierzu vermerkt der Roboter beim Befahren eines Bereiches in einer neuen Karte, welche Teile des Bereiches durch Beobachtung mit der Tiefenkamera schon erfasst wurden. Wurde bis dahin die Person noch nicht gefunden, muss auch der Rest des Bereiches überprüft werden. In der Karte sind die noch nicht überprüften Bereiche entsprechend markiert. Im Folgenden wählt der Roboter die nächste Beobachtungsposition so, dass er einen möglichst großen Teil des noch nicht betrachteten Bereiches überprüfen kann.

Das geschieht wie bei [VOLKHARDT und GROSS, 2013], durch Integration über den Erfassungsbereich auf der Karte der nicht betrachteten Bereiche. Wird die Person von dieser Position aus weiterhin nicht gefunden, wird die nächste Position, von der der noch nicht erfasste Bereich maximal ist, auf die gleiche Weise bestimmt. In weiterführenden Untersuchungen kann überprüft werden, ob ein großflächiger Bereich auf diese Weise schneller abgesucht werden kann, als mit einer Reihe festgelegter Beobachtungspositionen.

Ausblick bezüglich der Alarmierungskonzepte

Wie in Kapitel 6.2 zur Durchführung der Funktionstests beschrieben, war die Überprüfung der beiden Alarmierungskonzepte nicht Gegenstand dieser Untersuchung. Daher sollten weitere Untersuchungen durchgeführt werden, um zu prüfen, ob die in dieser Arbeit vorgestellten Ansätze zur dynamischen Festlegung der Schwellen für Inaktivität bzw. Immobilität dem Wunsch nach schneller Reaktionszeit bei gleichzeitig niedriger Fehlalarmquote gerecht werden. Diese Untersuchungen können im Weiteren als Referenz zur Beurteilung weiterer Schwellenberechnungen herangezogen werden. So wäre zu ermitteln, inwieweit sich die Aktivitätsmuster der Bewohner an verschiedenen Tagen der Wochen unterscheiden.

Das von [CUDDIHY et al., 2007] entwickelte und in dieser Arbeit aufgegriffene Konzept erfasst Aktivitäten lediglich auf 24-Stunden-Basis. Es findet keine Unterscheidung zwischen den Wochentagen statt, obwohl die Vermutung nahe liegt, dass Aktivitätsmuster zwischen den Wochentagen variieren können. Die für die Immobilitätserkennung genutzten Aufenthaltsdauern pro Bereich werden bislang ebenfalls nicht für verschiedene Stunden oder Tage betrachtet. Hier sollten zukünftige Untersuchungen ansetzen, um die nutzerabhängigen Aktivitätsmuster noch besser zu berücksichtigen.

Ausblick bezüglich möglicher Fehlalarme

Um die Anzahl an möglichen Fehlalarmen weiter zu reduzieren, muss das System unterscheiden können, wann eine notfallbedingte Inaktivität vorliegt bzw. ob eine Inaktivität durch Abwesenheit der Person in der Wohnung verursacht wurde. Zu diesem Zweck wird empfohlen, weitere Sensorik in Form von Türschließsensoren in die Wohnungen zu integrieren. Bei erkannter Abwesenheit kann das System pausiert werden,

um Fehlalarme zu vermeiden.

Ausblick zur Ausführung der Alarmierung

Ziel der vorliegenden Masterarbeit war es, einen funktionierenden Ablauf von der Alarmzeugung bis zur Beurteilung einer vorliegenden Notfallsituation zu erzeugen. Am Ende des beschriebenen Ablaufs steht die Erzeugung verschiedener Events, je nachdem ob eine stehende bzw. liegende Person gefunden wurde oder ob die Suche nach Überprüfung aller Bereiche erfolglos endete. Damit ist eine Schnittstelle für weitere Module geschaffen worden.

An dieser Stelle können weitere Arbeiten ansetzen und geeignete Konzepte entwickeln, mit denen Angehörige oder Rettungsstellen über eine Notsituation eines Nutzers informiert werden können. Denkbar wären hier Konzepte, die einen einfachen Anruf bei Angehörigen oder Pflegeeinrichtungen absetzen. Komplexere Anwendungen könnten zusätzliche Informationen über den genauen Ort, an dem sich die Person befindet senden oder ein Bild der Szene übertragen. Die Möglichkeiten weiterer Anwendungen sind vielfältig. In zukünftigen Arbeiten sollte daher ergründet werden, welche Datenübertragungsform zu einer schnellen und verlässlichen Reaktion auf einen Notfall führt. Die Basis dafür ist mit der vorliegenden Arbeit bereitet.

Anhang A

Protokolle des Funktionstests

Szenario L1	Nr. 1
Bereich Person: Küche	Startbereich Roboter: Esszimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 38 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 4 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none">• Person lag quer zum Roboter mit Beinen auf Fahrspur• zuerst nicht erkannt (da Roboter vor Füßen stand)• im Wegfahren Person erkannt (bei Drehung)	

Szenario L1 Nr. 2	
Bereich Person: Küche	Startbereich Roboter: Wohnzimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 31 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 2 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person lag längs zwischen BP1 und BP2 • Roboter nahm BP 1 ein und erkannte Person • Abstand zur Person 0,5 m 	

Szenario L1 Nr. 3	
Bereich Person: Flur	Startbereich Roboter: Esszimmer
Person wurde	<input type="checkbox"/> gefunden, nach – <input checked="" type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: – <input type="checkbox"/> nicht erreichbar, in Sektor: –
Anmerkung: <ul style="list-style-type: none"> • Person lag längs zur Roboterblickrichtung • Roboter stoppte Fahrt in Sektor 8, konnte aber Ziel (BP 3) nicht erreichen 	

Szenario L1 Nr. 4	
Bereich Person: Esszimmer	Startbereich Roboter: Küche
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 17 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 8 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Abstand zur Person 0,4 m 	

Szenario L1 Nr. 5	
Bereich Person: Wohnzimmer	Startbereich Roboter: Esszimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 63 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 1 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person lag zwischen BP 5 und BP 6 • Roboter nahm BP 5 und BP 4 ein, ohne Person zu erkennen • auf Fahrt zu BP 6 wurde Person gefunden • Abstand zu Person 0,3 m 	

Szenario L2 Nr. 1	
Bereich Person: Küche	Startbereich Roboter: Wohnzimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 174 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 3 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Roboter nahm BP 2, BP 5, BP 6, BP 4 und BP 1 ein, ohne Person zu erkennen • auf Fahrt zu BP 3 wurde Person gefunden • Abstand zu Person 0,5 m 	

Szenario L2 Nr. 2	
Bereich Person: Esszimmer	Startbereich Roboter: Küche
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 60 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 2 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Roboter fuhr im Anstand von 1,5 m an Person vorbei zu BP 4 • auf BP 4 wurde Person erkannt • Abstand zu Person 0,5 m 	

Szenario L2 Nr. 3	
Bereich Person: Wohnzimmer	Startbereich Roboter: Esszimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 75 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 7 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person lag zwischen Durchgang Esszimmer/Wohnzimmer und Wohnzimmertisch • Roboter fuhr zu BP 6 • auf Weg zu BP 5 wurde Person erkannt • zur Verifikation fuhr Roboter von Sektor 2 zum Sektor 7 • Person wurde erkannt • Abstand zu Person 0,4 m 	

Szenario L2 Nr. 4	
Bereich Person: Esszimmer	Startbereich Roboter: Wohnzimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 83 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 3 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person im Esszimmer an Wand zu Küche • Roboter fuhr zu BP 4, BP 5 und BP 6 • auf zur Küche wurde Person erkannt • Abstand zu Person 0,6 m 	

Szenario L2 Nr. 5	
Bereich Person: Küche	Startbereich Roboter: Esszimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 175 s <input type="checkbox"/> nicht gefunden
Roboter war	<input checked="" type="checkbox"/> erreichbar, in Sektor: 3 <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person in Küche 1,5 m von BP 1 entfernt • Roboter fuhr zu BP 2, Bp 4, BP 5, BP 6 und BP 1 • auf Fahrt zum nächsten Ziel wurde Person gefunden • Abstand zu Person 0,4 m 	

Szenario ST1 Nr. 1	
Bereich Person: Esszimmer	Startbereich Roboter: Küche
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 34 s <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person bei Sessel • Roboter fährt zur BP 4 • von BP 4 wurde Person erkannt und angesprochen 	

Szenario ST1		Nr. 2	
Bereich Person: Wohnzimmer		Startbereich Roboter: Esszimmer	
Person wurde		<input checked="" type="checkbox"/> gefunden, nach 15 s <input type="checkbox"/> nicht gefunden	
Roboter war		<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:	
Anmerkung: <ul style="list-style-type: none"> • Person zwischen BP 5 und BP 6 • auf Weg zu BP 6 gefunden 			

Szenario ST1		Nr. 3	
Bereich Person: Esszimmer		Startbereich Roboter: Wohnzimmer	
Person wurde		<input checked="" type="checkbox"/> gefunden, nach 61 s <input type="checkbox"/> nicht gefunden	
Roboter war		<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:	
Anmerkung: <ul style="list-style-type: none"> • Person an Wand zur Küche • Roboter fährt auf BP 4, BP 5, BP 6 • auf Weg zur Küche gefunden 			

Szenario ST1 Nr. 4	
Bereich Person: Küche	Startbereich Roboter: Esszimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 19 s <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person auf Höhe des Tisches zwischen BP 1 und BP 2 • auf Weg zu BP 1 gefunden 	

Szenario ST1 Nr. 5	
Bereich Person: Flur	Startbereich Roboter: Küche
Person wurde	<input type="checkbox"/> gefunden, nach – <input checked="" type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person im Flur auf Höhe S3 • Roboter fährt zu BP2, BP5, BP6, BP4, BP1 und BP3 • erkennt Person nicht • BP3 ca. 2 m von Person entfernt 	

Szenario ST2 Nr. 1	
Bereich Person: Küche	Startbereich Roboter: Esszimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 36 s <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person steht in Küche an der Spüle ca. 1 m von Roboterroute entfernt • Roboter fährt zu BP1 • auf Weg zu nächstem Ziel wird Person gefunden 	

Szenario ST2 Nr. 2	
Bereich Person: Esszimmer	Startbereich Roboter: Küche
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 15 s <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person nahe Stehlampe ca 1,4 m von Roboterroute entfernt • auf Weg zu BP4 wird Person gefunden, obwohl Person ca. 90° seitlich von Roboter steht 	

Szenario ST2 Nr. 3	
Bereich Person: Wohnzimmer	Startbereich Roboter: Esszimmer
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 14 s <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person am Sofa zwischen Durchgang zum Wohnzimmer und Tisch • Abstand zur Roboteroute ca. 0,8 m • auf Weg zu BP5 gefunden 	

Szenario ST2 Nr. 4	
Bereich Person: Küche	Startbereich Roboter: Wohnzimmer
Person wurde	<input type="checkbox"/> gefunden, nach – <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person hinter dem Türflügel der Küche ca. 1,5 m von BP1 entfernt • Person steht im Schatten der Tür • Roboter fährt auf BP1 • keine Erkennung 	

Szenario S Nr. 1	
Bereich Person: Esszimmer	Startbereich Roboter: Küche
Person wurde	<input checked="" type="checkbox"/> gefunden, nach 36 s <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person sitzt im Sessel nah am Durchgang zum Wohnzimmer • Roboter fährt auf BP4 • beim Losfahren wird Person erkannt 	

Szenario S Nr. 2	
Bereich Person: Wohnzimmer	Startbereich Roboter: Esszimmer
Person wurde	<input type="checkbox"/> gefunden, nach – <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person sitzt aus Sofa gegenüber vom Fernseher • Roboter erkennt Person nicht, obwohl direkt angesehen von BP5 und BP6 • Abstand ca. 2,2 m (BP5) bzw. 2,3 m (BP6) 	

Szenario S Nr. 3	
Bereich Person: Esszimmer	Startbereich Roboter: Küche
Person wurde	<input type="checkbox"/> gefunden, nach – <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person sitzt auf Sessel zur Küche • Roboter passiert Sessel zweimal (90° zur Person) ohne zu erkennen 	

Szenario S Nr. 4	
Bereich Person: Wohnzimmer	Startbereich Roboter: Esszimmer
Person wurde	<input type="checkbox"/> gefunden, nach – <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none"> • Person sitzt auf Sofa am Fenster • Roboter fährt auf BP6 • keine Erkennung im Abstand von $1,3\text{ m}$ 	

Szenario S	
Nr. 5	
Bereich Person: Küche	Startbereich Roboter: Wohnzimmer
Person wurde	<input type="checkbox"/> gefunden, nach – <input type="checkbox"/> nicht gefunden
Roboter war	<input type="checkbox"/> erreichbar, in Sektor: - <input type="checkbox"/> nicht erreichbar, in Sektor:
Anmerkung: <ul style="list-style-type: none">• Person sitzt auf Stuhl in Küche und blickt Richtung Esszimmer• nicht erkannt, obwohl zweimal direkt angesehen bei Fahrt zu BP2 und BP1	

Abbildungsverzeichnis

2.1	Zustände des Alarmierungskonzeptes	17
3.1	Struktur der Applikationsarchitektur	20
3.2	Datenaustausch zwischen MIRA-Units	22
4.1	Übersicht der Sensoren und Aktuatoren des Roboters Tweety	25
5.1	Programmablaufplan des Services zur Notfallbehandlung	28
5.2	Schichtenmodell der Notfallbehandlung	31
5.3	FallenPersonDetektor	34
5.4	Verlauf der dynamischen Alarmschwellen	40
5.5	Erzeugung der Verifikationspositionen	50
5.6	Aufbau des Dialoges	55
5.7	Gesichtsausdrücke des Roboters	56
6.1	Aufbau der Testumgebung	59
6.2	Sektoren um die liegende Person	64

Literaturverzeichnis

- [ABOWD et al., 2002a] ABOWD, GREGORY D., A. BATTESTINI und T. O'CONNELL (2002a). *The Location Service: A framework for handling multiple location sensing technologies*. ... Institute for Technology, ...
- [ABOWD et al., 2002b] ABOWD, GREGORY D., A. F. BOBICK, I. A. ESSA, E. D. MYNATT und W. A. ROGERS (2002b). *The aware home: A living laboratory for technologies for successful aging*. Proceedings of the AAAI-02 Workshop Automation, S. 1–7.
- [BHARADWAJ et al., 2014] BHARADWAJ, RICHA, S. SWAISAENYAKORN, C. PARINI, J. BATCHELOR und A. ALOMAINY (2014). *Motion Tracking of a Human Subject in Healthcare Applications Using Compact Ultra Wideband Antennas*. Proceedings of the 4th International Conference on Wireless Mobile Communication and Healthcare - Transforming healthcare through innovations in mobile and wireless technologies“, S. 199–202.
- [CRANDALL und COOK, 2011] CRANDALL, AARON S. und D. J. COOK (2011). *Tracking systems for multiple smart home residents*. Ambient Intelligence and Smart Environments, 9:65–82.
- [CUCCHIARA et al., 2007] CUCCHIARA, RITA, A. PRATI und R. VEZZANI (2007). *A multi-camera vision system for fall detection and alarm generation*. Expert Systems, 24(5):334–345.
- [CUDDIHY et al., 2007] CUDDIHY, PAUL, J. WEISENBERG, C. GRAICHEN und M. GANESH (2007). *Algorithm to automatically detect abnormally long periods*
-

- of inactivity in a home*. Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments, S. 89–94.
- [DALAL und TRIGGS, 2005] DALAL, NAVNEET und B. TRIGGS (2005). *Histograms of oriented gradients for human detection*. In: *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, Bd. I, S. 886–893.
- [EINHORN und LANGNER, 2012] EINHORN, ERIK und T. LANGNER (2012). *Mira-middleware for robotic applications*. Intelligent Robots and . . . , (Iros):2591–2598.
- [FLOECK und LITZ, 2008] FLOECK, MARTIN und L. LITZ (2008). *Activity- and inactivity-based approaches to analyze an assisted living environment*. Proceedings - 2nd Int. Conf. Emerging Security Inf., Systems and Technologies, SECURWARE 2008, Includes DEPEND 2008: 1st Int. Workshop on Dependability and Security in Complex and Critical Inf. Sys., S. 311–316.
- [GAMBS et al., 2012] GAMBS, SÉBASTIEN, M.-O. KILLIJIAN und M. N. DEL PRADO CORTEZ (2012). *Next place prediction using mobility Markov chains*. Proceedings of the First Workshop on Measurement Privacy and Mobility MPM 2012, S. 1–6.
- [GROSS et al., 2012] GROSS, H. M., C. SCHROETER, S. MUELLER, M. VOLKHARDT, E. EINHORN, A. BLEY, T. LANGNER, M. MERTEN, C. HUIJNEN, H. VAN DEN HEUVEL und A. VAN BERLO (2012). *Further progress towards a home robot companion for people with mild cognitive impairment*. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, S. 637–644.
- [GROSS et al., 2011] GROSS, H. M., C. SCHROETER, S. MUELLER, M. VOLKHARDT, E. EINHORN, A. BLEY, C. MARTIN, T. LANGNER und M. MERTEN (2011). *Progress in developing a socially assistive mobile home robot companion for the elderly with mild cognitive impairment*. IEEE International Conference on Intelligent Robots and Systems, S. 2430–2437.
-

-
- [GROSS et al., 2015] GROSS, HORST-MICHAEL, S. MUELLER, C. SCHROETER, M. VOLKHARDT, A. SCHEIDIG, K. DEBES, K. RICHTER und N. DOERING (2015). *Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments*. In: *IEEE International Conference on Intelligent Robots and Systems*, Bd. 2015-Decem, S. 5992–5999.
- [KALTIOKALLIO et al., 2012] KALTIOKALLIO, OSSI, M. BOCCA und N. PATWARI (2012). *Follow @grandma: Long-term device-free localization for residential monitoring*. Proceedings - Conference on Local Computer Networks, LCN, S. 991–998.
- [KLEINBERGER et al., 2011] KLEINBERGER, T., A. JEDLITSCHKA, H. STORF, W. PUTZ, A. GROSS, F.-I. EXPERIMENTELLES und S. ENGINEERING (2011). *Notfallerkennung und -prävention: Ergebnisse und Verwertung*. In: *4. Deutscher AAL-Kongress*.
- [KOCH et al., 2007] KOCH, JAN, J. WETTACH, E. BLOCH und K. BERNS (2007). *Indoor localisation of humans, objects, and mobile robots with RFID infrastructure*. Proceedings - 7th International Conference on Hybrid Intelligent Systems, HIS 2007, S. 271–276.
- [LEWANDOWSKI, 2016] LEWANDOWSKI, BENJAMIN (2016). *Robust depth data based fallen person detection in domestic environments*. Doktorarbeit, TU Ilmenau.
- [LI et al., 2015] LI, YANFEI, M. LIU und W. SHENG (2015). *Indoor human tracking and state estimation by fusing environmental sensors and wearable sensors*. 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015, S. 1468–1473.
- [LUO und CHEN, 2013] LUO, REN C. und O. CHEN (2013). *Wireless and pyroelectric sensory fusion system for indoor human/robot localization and monitoring*. IEEE/ASME Transactions on Mechatronics, 18(3):845–853.
- [MÜLLER, 2016] MÜLLER, STEFFEN (2016). *Realisierung nutzeradaptiven Interaktionsverhaltens für mobile Assistenzroboter*. Doktorarbeit, Technische Universität Ilmenau, Ilmenau.
-

-
- [PFLÜGER et al., 2012] PFLÜGER, MARIUS, J. KROLL und B. STEINER (2012). *Automatische Notfallerkennung durch optische und akustische Sensoren*. 5. Deutscher AAL-Kongress Berlin.
- [ROBINSON et al., 2014] ROBINSON, HAYLEY, B. MACDONALD und E. BROADBENT (2014). *The Role of Healthcare Robots for Older People at Home: A Review*. International Journal of Social Robotics, 6(4):575–591.
- [RODNER et al., 2011] RODNER, T., M. FLOECK und L. LITZ (2011). *Inaktivitätsüberwachung und Alarmhandlung zur Verringerung von Fehlalarmen*. In: *4. Deutscher AAL-Kongress*, S. 1–5.
- [SCHINDLER et al., 2006] SCHINDLER, GRANT, C. METZGER und T. STARNER (2006). *A wearable interface for topological mapping and localization in indoor environments*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3987 LNCS:64–73.
- [TABAR et al., 2006] TABAR, ALI MALEKI, A. KESHAVARZ und H. AGHAJAN (2006). *Smart Home Care Network using Sensor Fusion and Distributed Vision-based Reasoning*. Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, S. 145–154.
- [TUZINSKI und DRESCHER, 2008] TUZINSKI, RÜDIGER und R. DRESCHER (2008). *Fußböden als Bestandteil der Gebäudeautomation*. 62:722–723.
- [VOLKHARDT und GROSS, 2013] VOLKHARDT, MICHAEL und H. M. GROSS (2013). *Finding people in apartments with a mobile robot*. Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013, S. 4348–4353.
- [VOLKHARDT et al., 2013] VOLKHARDT, MICHAEL, C. WEINRICH und H.-M. GROSS (2013). *Multi-Modal People Tracking on a Mobile Companion Robot*. In: *6th European Conference on Mobile Robots (ECMR 2013)*, S. 288–293, Barcelona.
-

-
- [WEINRICH, 2016] WEINRICH, CHRISTOPH (2016). *Personenwahrnehmung für eine sozialverträgliche und nutzerzentrierte Roboternavigation in öffentlichen Einsatzumgebungen*. Doktorarbeit, Technische Universität Ilmenau.
- [WENGEFELD et al., 2016] WENGEFELD, TIM, M. EISENBACH, T. Q. TRINH und H.-M. GROSS (2016). *May I be your Personal Coach? Bringing Together Person Tracking and Visual Re-identification on a Mobile Robot*. In: *Int. Symposium on Robotics (ISR)*, S. 141–148, München.
- [WIESE et al., 2011] WIESE, C. H. R., K.-P. ITTNER, B. M. GRAF und C. L. LASSEN (2011). *Palliative Notfälle - Definition, Besonderheiten und therapeutische Entscheidungen*.
- [YANG et al., 2015] YANG, DAN, W. SHENG und R. ZENG (2015). *Indoor human localization using PIR sensors and accessibility map*. 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), S. 577–581.
- [YU et al., 2007] YU, CHEN RONG, C. L. WU, C. H. LU und L. C. FU (2007). *Human localization via multi-cameras and floor sensors in smart home*. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, 5:3822–3827.
-